RENATA TURKEŠ

TOPOLOGICAL DATA ANALYSIS: WHAT, WHY, HOW & WHEN?

TOPOLOGICAL DATA ANALYSIS: WHAT, WHY, HOW & WHEN?

RENATA TURKEŠ

Thesis submitted to obtain the degree of Doctor in Computer Science
Supervisor: Steven Latré

September 2024

Supervisor
Prof. dr. Steven Latré (University of Antwerp)

Members of the Examination Committee
Prof. dr. Jan Aerts (KU Leuven)
Prof. dr. Erik Mannens (University of Antwerp)
Prof. dr. Johann Marquez-Barja (University of Antwerp)
Prof. dr. Nina Otter (Paris-Saclay University)
Prof. dr. Tim Verdonck (University of Antwerp)
Prof. dr. Tijl De Bie (University of Ghent)

Cover design: University of Antwerp
Printing and binding: Universitas, Antwerp

For Mom.

English abstract

Topological data analysis is a recent and fast growing field that approaches the analysis of (the shape of) data using techniques from algebraic topology. Its main tool, persistent homology, captures information about cycles in the data: its connected components, loops, voids, and so on.

The first part of the thesis investigates the properties of persistent homology. Homology captures cycles in the data, but what additional information is stored in the persistence (“the what”)? We show that it can capture the number of holes, but also important geometric notions of curvature and convexity. Stability theorems provide mathematically provable guarantees of desirable properties of persistent homology (“the why”), but how do these theoretical results translate to practice? We show that persistent homology does not always yield noise robust features in a classification task.

The second part explores two applications of persistent homology. Firstly, we demonstrate how it can be used to study the preservation of topology and geometry (“the how”); here we focus on hyperdimensional computing that encodes the input data into a very high dimensional space. Secondly, we extract persistent homology features from EEG data during an audiovisual task in order to detect attention, providing some understanding when it can be useful in neuroscience applications (“the when”).

In both parts, we depart from the dominant stream in the literature by highlighting the versatility of filtrations and signatures, the input and output of persistent homology, beyond the canonical choices. In this way, we showcase how persistent homology can be seen as a diverse family of rich descriptors of different aspects of shape.

Dutch abstract

Topologische gegevensanalyse is een recent en snelgroeiend vakgebied dat (de vorm van) gegevens analyseert met behulp van technieken uit de algebraïsche topologie. Het belangrijkste onderdeel van topologische gegevensanalyse, persistente homologie, vergaart informatie over de cycli in de gegevens: de samenhangscomponenten, lussen, leegtes, enzovoort.

Het eerste deel van het proefschrift onderzoekt de eigenschappen van persistente homologie. Homologie legt de cycli in de gegevens vast, maar welke bijkomende informatie is verscholen in de persistentie ("de wat")? We tonen aan dat het niet alleen het aantal gaten vastlegt, maar ook belangrijke geometrische concepten van kromming en convexiteit. Stabiliteitstheorema's bieden wiskundig aantoonbare garantie voor gewenste eigenschappen van persistente homologie ("de waarom"), maar hoe vertalen deze theoretische resultaten zich in de praktijk? We tonen aan dat persistente homologie niet altijd kenmerken die robust zijn tegen ruis oplevert in een classificatietaak.

Het tweede deel onderzoekt twee toepassingen van persistente homologie. Ten eerste, we tonen aan hoe het kan gebruikt worden om het behoud van topologie en geometrie te bestuderen ("de hoe"); hierbij concentreren we ons op hyperdimensionaal rekenen dat invoergegevens integreert in een zeer hoge dimensionale ruimte. Ten tweede, halen we tijdens een audiovisuele taak kenmerken van persistente homologie uit EEG gegevens om aandacht op te sporen, wat ons inzicht verschaft wanneer het nuttig kan zijn in neurowetenschappelijke toepassingen ("de wanneer").

In beide delen wijken we af van de dominante richting in de literatuur door de veelzijdigheid van filtraties en kentekenen, de invoer en uitvoer van persistente homologie, te benadrukken bovenop de canonieke keuzes. Op die manier laten we zien hoe persistente homologie kan worden beschouwd als een diverse familie van uitgebreide beschrijvende elementen van verschillende aspecten van de vorm.

Publications

This thesis is based on the following first-author publications:

- Renata Turkeš, Guido Montúfar, and Nina Otter, *On the effectiveness of persistent homology*, Advances in Neural Information Processing Systems 35 (NeurIPS 2022): 35432-35448.

- Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré, *Noise robustness of persistent homology on greyscale images, across filtrations and signatures*, PLOS ONE 16, No. 9 (2021): e0257215.

- Renata Turkeš, Werner van Leekwijck, and Steven Latré, *Topological data analysis of structure preservation via hyperdimensional computing embedding*, under review.

- Renata Turkeš, Steven Mortier, Jorg De Winne, Dick Botteldooren, Paul Devos, Steven Latré and Tim Verdonck, *Who is WithMe? EEG features for attention in a visual task, with auditory and rhythmic support*, under review.

In addition, the appendices include two second-author publications:

- Shuang Liang, Renata Turkeš, Jiayi Li, Nina Otter, and Guido Montúfar, *Pullback geometry of persistent homology encodings*, Transactions on Machine Learning Research (2024).

- Steven Mortier, Renata Turkeš, Jorg De Winne, Wannes Van Ransbeeck, Dick Botteldooren, Paul Devos, Steven Latré, Marc Leman, and Tim Verdonck, *Classification of targets and distractors in an audiovisual attention task based on EEG*, Sensors (2023).

Acknowledgments

Thank you so much to everyone who has helped to make this thesis happen, either by being directly involved in the research, providing a keen eye, or making me laugh when things got difficult:

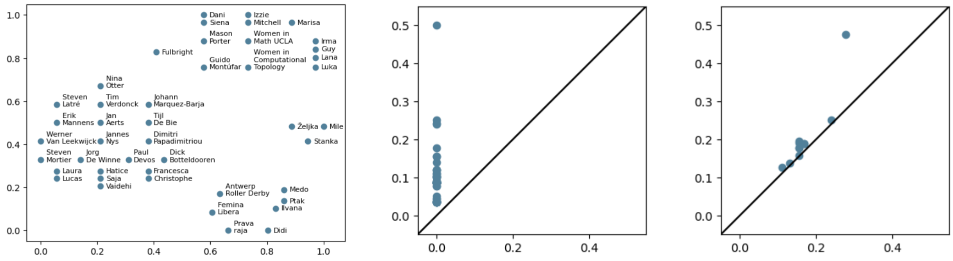


Figure 1.: Persistent homology of thesis patrons.

Persistent homology of the graph of patrons (left plot) reveals that there are four persistent connected components (points in the middle plot far from diagonal): thesis jury and IDLab colleagues and collaborators, UCLA community and other international contacts, friends, and family; and one persistent loop (point in the right plot far from diagonal) indicating a cyclic relationship between these groups — in its center, this thesis.

Ptak, your light shines through, and for that I am forever grateful.

THANK YOU / BEDANKT / HVALA !

Renata Turkeš
Antwerp, September 2024

Contents

List of Figures

List of Tables

Acronyms

TDA	topological data analysis
PH	persistent homology
PD	persistence diagram
PI	persistence image
PL	persistence landscape
HDC	hyperdimensional computing
HD	hyperdimensional
EEG	electroencephalography
SVM	support vector machine

Introduction

Topological data analysis (TDA) is a field that stands at the crossroads of diverse disciplines, such as data analysis, algebraic topology, computational geometry, computer science, statistics, and other related areas. Its main tool, persistent homology (PH), captures information about k-dimensional cycles (connected components, loops, voids, etc.), thereby providing representations of the shape of data. **Chapter 1** gives the preliminaries about persistent homology, where we immediately emphasize the versatility of filtrations and persistence signatures, the input and output of PH. We also introduce the stability theorems, the pivotal result about PH that guarantees robust behavior under minor perturbations of the input.

Over the past two decades, PH has found many applications in data science, e.g., in the analysis of local behavior of the space of natural images, analysis of images of hepatic lesions, human and monkey fibrin, fingerprints, or diabetic retinopathy images, analysis of 3D shapes, neuronal morphology, brain artery trees, protein binding, genomic or orthodontic data, coverage in sensor networks, plant morphology, fluid dynamics, cell motion, structure of amorphous and nanoporous materials, or spread of the Zika virus.[1]

Despite the widespread applications of PH, the reasons behind its success remain elusive; in particular, it is not known for which classes of problems is PH most effective, or to what extent can it detect geometric or topological features. In other words, an important research question is the following:

(RQ) What is seen with persistent homology, or in more detail, what is seen with long and short persistence intervals?

Chapter 2 ("the what") aims to unravel some of these mysteries by identifying some types of problems that can be tackled with PH. 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. There is theoretical evidence that the number of holes of the underlying space can be detected from PH (under some conditions about the target space, the sample density and closeness to the space), but detection of the number of holes has only been explored for some example point clouds. It has also recently been shown both theoretically and experimentally that PH can predict curvature, which inspired us to investigate this problem in

[1] An exhaustive collection of applications of topological data analysis (TDA) to real data can be found at `https://donut.topology.rocks/`.

more detail. Regarding the important geometric problem of classification between convex and concave shapes, to the best of our knowledge there are no previous works investigating the applicability of PH to this task. Our contributions can be summarized as follows.

- We prove that PH can detect convexity in $\mathbb{R}^d$. This is a new and first result that connects PH and convexity.

- We define the tubular filtration, a new filtration (medium through which PH is extracted from data), that is crucial for the detection of convexity.

- We demonstrate experimentally that PH can detect the number of holes, curvature, and convexity from synthetic point clouds in $\mathbb{R}^2$ or $\mathbb{R}^3$ in practice, outperforming support vector machine (SVM) and fully-connected networks on distance matrices, and PointNet, a benchmark deep learning architecture for point clouds. For convexity detection, we also show that PH obtains a good performance on real-world image data of plant leaves.

- We demonstrate experimentally that PH features allow to solve the above tasks even in the case of limited training data, noisy and out-of-distribution test data, and limited computational resources.

- We provide insights about the topological and geometric features that are captured with long and short persistence intervals, and formulate guidelines for applications that are suitable for PH.

- We provide data sets that can be directly used as a benchmark for our tasks or other related point-cloud-analysis or classification problems. More or modified data can be easily constructed with the publicly shared code.

It is often said that the key factor contributing to the practical success of PH is its proven stability, due to the common presence of noise or measurement errors in real data. Stability theorems ensure that the perturbation of PH under noise is upper bounded by the perturbation of its input. However, little attention has been directed towards understanding what these theorems mean in practice. In **Chapter 3 ("the why")**, we ask the following questions:

(RQ1) How much does PH change under noise in the data?

(RQ2) How discriminative does PH remain if the data contains noise?

To address these questions, we evaluate the noise robustness of PH on the MNIST data set of greyscale images. We investigate to what extent PH changes under typical forms of image noise, and quantify the loss of performance in classifying the MNIST handwritten digits when noise is added to the data. More precisely, the contributions are as follows:

- We evaluate the Euclidean l_2 distance between the filtration functions, and Wasserstein W_2 or l_2 distance between the persistence signatures, across 1 000 MNIST images, which quantify the noise sensitivity of the filtrations and resulting PH under rotation, translation, stretch-shear-reflect, brightness and contrast change, or Gaussian, salt and pepper or shot noise. These distances provide insights into noise sensitivity of PH for different filtrations and persistence signatures under different types of noise.

- We evaluate the drop in accuracy of the SVM trained on non-noisy data, and tested on non-noisy and then noisy data. These results help explain the importance of the particular data set (classification problem) for the connection between the stability of PH under noise, and the noise robustness of PH in a classification task.

The results show that the sensitivity to noise of PH is influenced by the choice of filtrations and persistence signatures, and in particular, that PH features are often not robust to noise in a classification task.

The second part of the thesis further investigates the effectiveness of PH by considering two applications. In **Chapter 4 ("the how")**, we show how PH can be used to analyze the preservation of structure with a given mapping. We focus on hyperdimensional computing (HDC), an emerging and efficient learning paradigm that centers on encoding the input data space into a very high dimensional (HD) shape (typically, $D = 10\,000$). Due to the hyperdimensionality and randomness that is underlying the encoding, these spaces are difficult to study and understand. The following research questions are of interest:

(RQ1) What kind of topological and geometric structure in the input space is preserved in the hyperdimensional (HD) space?

(RQ2) What is the power of linear separators in the HD space, compared to the input space?

(RQ3) To what extent is noise tolerated in the HD space?

The literature on HDC can be seen as taking one of two directions. On the one hand, there are some theoretical studies of the behavior of HDC, which provide little intuition on how and why it works. On the other hand are the many applications of HDC in various domains, that focus on the performance of HDC (accuracy of the HDC classifier) on some real-world data that is very complex, providing little understanding about the HD space. To this end, we employ PH to reveal the topological and geometric features that are preserved with this encoding: for instance, it allows to formally study whether a circle in the plane remains a circle in the high-dimensional space. The contributions are focused on an encoding of Euclidean data, and are as follows.

- We provide an intuitive exposition of approximate distance preservation.
- For several synthetic, well-understood Euclidean data sets (in addition to benchmark real-world data) and noise in the input and HD space, we quantify the correlation between the distance matrices between the input and HD data, the perturbation of PH with the HDC encoding, and linear separability of the different representations.

The findings provide insights into the properties of the input data are preserved with the HDC encoding, and to what extent is the approximation of structure influenced by the parameters of the HDC encoding. We see that the number of persistent connected components and loops mainly remains unchanged after encoding; however, the non-negligible distance between PH on input and HD data points to some changes in geometry. Moreover, although the (approximate) distance preservation implies the preservation of linear separability, our experiments also show that the input data with non-linear decision boundary might be linearly separable in the HD space.

The final **Chapter 5 ("the when")** explores the effectiveness of PH in neuroscience applications. We consider a novel WithMe paradigm where experimental participants are shown sequences of 5 so-called `Target` and 5 so-called `Distractor` digits on the computer screen, being instructed to recall the `Target` digits. The main goal of this project is to understand in what ways can the resulting electroencephalography (EEG) data capture attention, which plays a pivotal role in advancing our understanding of cognition. Identifying such biomarkers that monitor attention can enhance the interaction between humans and AI agents, opening a range of applications in health, revalidation or public security, such as improving gait characteristics of Parkinson's patients or detecting driver fatigue. More precisely, the main goal of this chapter is to gain some understanding into the following research question:

(RQ) What type of WithMe electroencephalography (EEG) data representations are closely associated with attention?

We evaluate the performance of SVM on various features extracted from univariate time series (such as time domain features and recurrence plots), along with representations derived directly from multivariate time series (such as global field power and functional brain networks). Both of these types of representations include PH features, that draw interest due to their stability, since neural data is very noisy (as a result of changes in machine calibrations, spurious participant movements and environmental conditions). The results are benchmarked against baseline EEG-specific models, including a deep architecture that learns the task-specific features. The main insights are as follows.

- Attention can be captured with EEG, even for the short visual stimulus, as indicated by an accuracy of more than 70% for some of the features, when classifying between the `Target` and `Distractor` stimuli.

- Raw time series values, concatenated across EEG channels, are among the most discriminative features. This contradicts the common understanding in the literature that the extraction of relevant features is a crucial step in EEG processing.

- PH features perform poorly. Positioning our work with existing applications of TDA in neuroscience, we point to the likely sources of the poor performance, and the type of applications where PH can offer more meaningful insights.

The findings thus serve as a cautionary tale that sheds some light on the specific problems in neuroscience where PH may not have an advantage.

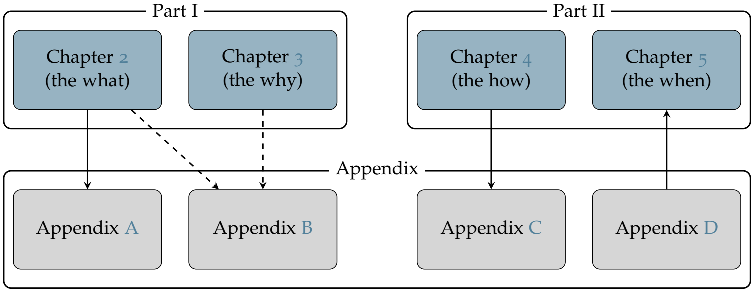


Figure 2.: Overview of the thesis structure. The main chapters (What, Why, When & How, in blue) are based on four first-author publications, that investigate the properties (Part I) and applications (Part II) of persistent homology. The appendices (in grey) include the supplementary material, and two second-author publications that are closely related to some of the main chapters in the thesis.

An overview of the thesis is given in Figure 2. The data and code for the main chapters are available on my personal website `renata-turkes.github.io/`, and it also includes the slides, poster and video presentations whenever available. Appendix A contains the supplementary material for Chapter 2: it includes the proofs of theoretical results, and the additional details about the experiments. Appendix B is closely related to Chapter 2 and Chapter 3, as it proposes pull-back geometry to study the sensitivity of PH to a given feature of interest (which translates to the ability to solve a given classification or regression problem), or given perturbations (such as affine transformations and noise). The supplementary material that details the experimental results for every data set considered in Chapter 4 is given in Appendix C. Finally, Appendix D can be seen as a precursor to Chapter 5, as it validates the WithMe EEG data used in that chapter: here, we show that the benchmark machine and deep learning models from the literature perform well, indicating that the WithMe experimental paradigm is appropriate for eliciting attention.

Finally, in Chapter 6, we address some statements and ideas about PH that are often found in the literature but that are inaccurate, imprecise, or prone to misinterpretation. By pointing to specific results from the thesis, we offer nuanced insights into these "myths" about PH.

1

Background on persistent homology

We start this chapter with intuitive definitions of shape, geometry, topology and homology, leading to a gentle introduction to persistent homology (PH). We highlight how the PH pipeline involves the choice of filtration and persistence signature, the input and output for PH. We then define the most common filtrations: Vietoris-Rips, Distance-to-Measure (DTM) and height, the widely used signatures: persistence diagram (PD), persistence image (PI), persistence landscape (PL), and the standard metrics between them. The final sections discuss (the theoretical complexity of) PH computation, and introduce the main theoretical results that guaranteee the stability of PH under minor perturbations of the input.

1.1 Shape, geometry, topology, (persistent) homology

As humans we make sense of the world around us by studying the "shape" of objects in our real life or digital environment. Geometry offers a common language to model and describe shapes, for instance with descriptors such as distances or angles [98]. Geometric similarity (which can be evaluated with Gromov-Hausdorff distance) measures how far two subsets of a metric space are from each other, up to an isometry, a map that preserves the distances. Invariance under isometries, such as rotation or translation, is obviously useful for object recognition. In some applications, however, the detailed geometric information can be redundant, or even noise, as it can obscure the useful structure. A notable example is the *Seven Bridges of Königsberg* problem, a historic puzzle concerning the possibility of a walk through the city of Königsberg crossing each of its seven bridges exactly once and returning to the starting point (Figure 1.1, left plot).

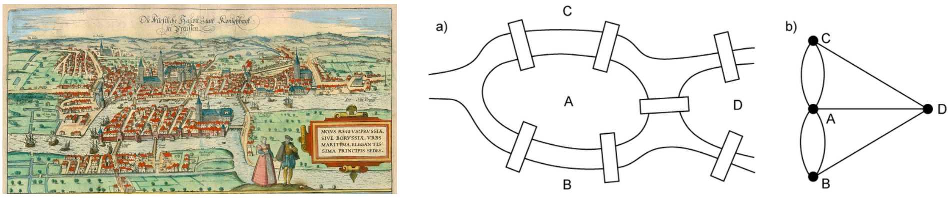


Figure 1.1.: Seven Bridges of Königsberg problem: Can you cross every bridge once and return to the starting point (left plot [264])? Euler abstracted the puzzle into a mathematical graph (right plot [39]): since only the connection information is relevant, the shape of pictorial representations may be distorted in any way, e.g., the distances are irrelevant, it does not matter whether the edges drawn are straight or curved, or whether one node is to the left or right of another.

As the story goes, by focusing solely on the relationships and connections between objects and disregarding the specific measurements such as distances or detailed notions of shape, Leonhard Euler abstracted the physical details into a mathematical graph (marking the birth of graph theory, a branch of mathematics deeply rooted in topology), where the graph nodes represent the landmasses, and the graph edges represent the bridges (Figure 1.1, right plot). This allowed Euler to realize that the problem's solvability was not dependent on the distances or physical layout, and to rather analyze the problem's inherent connectivity structure. In this way, *Seven Bridges of Königsberg* problem reduces to finding an Eulerian cycle in a multi-graph with four nodes and seven edges, which was famously proven impossible by Euler in 1736. What he observed is that if every graph edge (bridge) is to be crossed only once, then every graph node (landmass) must have an even number of edges attached to it, implying that the answer to the problem is negative. In a similar fashion, by focusing on the connectivity and relationships between locations rather than exact geographical accuracy, urban metro maps simplify complex spatial information into easily understandable and clear representations. As another example, we can think of animation in computer graphics, if one aims to develop software that can continuously deform one object into another, in the sense that one can stretch and change the shape, but cannot break or add to the shape [98].

Topology is a discipline in mathematics that studies properties that are preserved under homeomorphisms - isomorphisms or structure-preserving maps between topological spaces: stretching, bending, twisting, but not tearing or gluing, that are continuous, one-to-one and onto, with continuous inverse. Hence, the joke that a topologist cannot distinguish a coffee mug from a donut (Figure 1.2), is mathematically formalized through the fact that there exists a homeomorphism between these shapes, i.e., either shape can be continuously deformed into the other. These topological properties that are preserved under homeomorphisms can be seen as as alternative notion of shape or structure, that is weaker than the geometric properties that rely on a metric (distance): for instance, a small circle, a large circle, and a square, have the same topology. In other words, topology can be seen as a type of geometry devoid of concrete spatial notions such as curvature, convexity, distance, and the like [97] (and is sometimes referred to as "rubber-sheet geometry"). As the most general type of a mathematical spaces that allow for the definition of connectedness, continuity and limits, without specifying a metric, topological spaces are increasingly relevant to applications in data, networks and biology, where natural metrics may be obscured or nonexistent [135]. However, it is known that there cannot exist an algorithm to determine when arbitrary topological spaces are homeomorphic [250], i.e., to determine if two spaces have the same topology. We can however compare some weaker notions, such as dimension and connectivity. This allows us to assess the topological similarity between two spaces with the distance between their topological invariants.

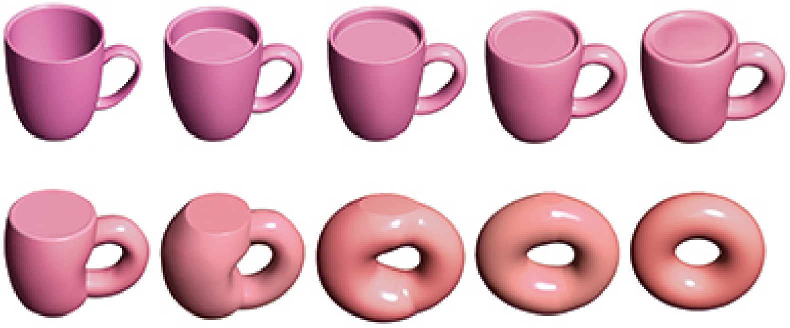


Figure 1.2.: A famous and traditional joke is that a topologist cannot distinguish a coffee mug from a donut. This is because the two objects are topologically equivalent, since it is possible to deform, continuously and without tearing, one into the shape of the other. The illustration is taken from [126].

One of the essential topological invariants is homology, and the related Betti numbers. Homology goes back to the beginnings of topology in Poincaré's influential papers, who represented the notion of connectivity of a space with its k-dimensional cycles: connected components ($k = 0$), loops ($k = 1$), cavities ($k = 2$) and cycles in higher homological dimensions. These cycles are shown to organize themselves into abelian groups, called homology groups, and their ranks are non-negative integers referred to as the Betti numbers of the space. The k-th Betti number β_k is the number of independent k-dimensional cycles that are not boundaries [119]. For example, the circle has Betti numbers $\beta_0 = 1$, $\beta_1 = 1$, $\beta_2 = 0$, and for a torus, we have $\beta_0 = 1$, $\beta_1 = 2$, $\beta_2 = 1$. This homology information can be very useful, as it allows to uncover the underlying structure of a space, and distinguish between topological spaces. Indeed, as a topological invariant, Betti number is preserved under homeomorphisms, so that two spaces cannot have the same topology if some of their Betti numbers are different: for instance, sphere and torus do not have the same topology since their β_1 is respectively 0 and 2 (Figure 1.3).

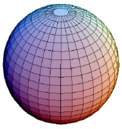
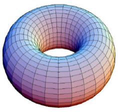
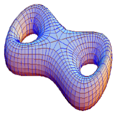
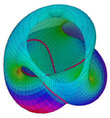
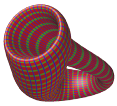
Betti number	Sphere	Torus	Two-holed torus	Projective plane	Klein bottle
					
β_0	1	1	1	1	1
β_1	0	2	4	1	2
β_2	1	1	1	1	1

Figure 1.3.: The Betti numbers β_0, β_1 and β_2 reflect respectively the 0-, 1- and 2-dimensional homology, i.e., the number of connected components, loops and voids (0-, 1- and 2-dimensional cycles). [1]

Note, however, that having the same Betti numbers is a strictly weaker property than having the same topology. Consider, for example, the circle and the cylinder. Both topological spaces consist of one connected component ($\beta_0 = 1$), one loop ($\beta_1 = 1$) and no higher dimensional cycles ($\beta_i = 0$ for $i \geq 2$), but they do not have the same topology as they do not have the same intrinsic dimension. For the same reason, a point and a disk have the same homology (there is a so-called homotopy equivalence between them, i.e., continuous functions that resemble stretching and bending) and therefore the same Betti numbers: one connected component ($\beta_0 = 1$), and no higher dimensional cycles ($\beta_i = 0$ for $i \geq 1$); but they do not have the same topology (there is no homeomorphism between them, i.e., no bijective continuous transformation with a continuous inverse). Homology thus allows us to distinguish between fewer spaces than topology, but unlike topology, homology admits effective computation and comparison algorithms [345]. For a detailed study of homology, we refer to [153].

Persistent homology (PH) can be seen as an extension of the notion of homology to data (e.g., a point cloud, a finite set of data points in space). Real data are a finite set of observations and do not directly reveal any topological information, since topological features are usually associated with continuous spaces. To circumvent this issue, the underlying topological structure of the data can be estimated e.g., with a simplicial complex (Section 1.2), a high-dimensional extension of a graph that consists of vertices, edges, triangles and so on. One can, for instance, connect each two point cloud points with an edge if they are within a given distance $r \in \mathbb{R}$. However, an important question immediately arises: how to choose this scale r? For the example shape in the left plot of Figure 1.4 scale $r = 0.3$ (middle plot) captures only the small circle, whereas scale $r = 0.6$ (right plot) captures only the big circle.

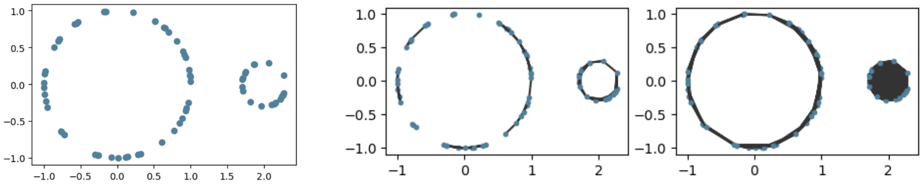


Figure 1.4.: A finite metric space, a point cloud, does not reveal any interesting homology: each point is a connected component, and there are no higher dimensional cycles, so that any two point clouds with the same number of points have the same homology (left plot). A point cloud can be represented with a topological space by connecting each two points with an edge if they are within a given distance $r \in \mathbb{R}$. The choice of scale r is important, since for the given example, $r = 0.3$ reveals only the small circle (middle plot), whereas $r = 0.6$ only captures the big circle (right plot) as the small circle already gets "filled up".

A notable advantage of PH is that it does not require to choose and fix the scale $r \in \mathbb{R}$. Instead, PH relies on a so-called filtration (Section 1.3), a nested family of topological spaces that approximate the data at *any* scale $r \in \mathbb{R}$. Matrix reduction algorithm (Section 1.5) then allows to calculate the information about k-dimensional cycles that *persist* across different scales of data [109, 111, 112, 384]. PH registers the scale (also referred to as resolution, or time) $r \in \mathbb{R}$ at which every k-dimensional cycle appears and disappears in the filtration, that are respectively referred to as the birth and death values b and d. More precisely, PH is a multi-set of persistence pairs, or intervals, (b, d). The k-dimensional cycles (b, d) with large persistence or lifespan $l = d - b$ (i.e., long intervals) represent relevant topological, i.e., homological, properties of the input space (Figure 1.5). As we discuss throughout this thesis, the birth and death values of each cycle often reveal additional information so that, although homology is weaker than topology (which is further weaker than geometry), PH can capture geometric information and therefore goes beyond topology (Figure 1.6).

This PH information can be represented via different signatures, e.g., using sets, vectors, functions, or scalars (Section 1.4). The standard representation of PH is via

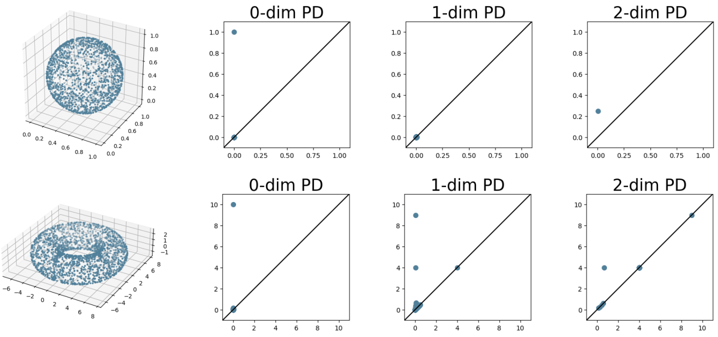


Figure 1.5.: The 0-, 1- and 2-dimensional persistent homology PH_0, PH_1 and PH_2 of a point cloud sampled from the sphere (top row) and torus (bottom row) captures the connected components, loops and voids (respectively 0-, 1- and 2-dimensional cycles). The number of long persistence intervals (b, d) in the multi-set PH_k, here represented with scatter plot points far from the diagonal, reflects the number of k-dimensional cycles (Betti number β_k, see Figure 1.3): $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 1$ for the sphere, and $\beta_0 = 1$, $\beta_1 = 2$, $\beta_2 = 1$ for the torus.

a so-called persistence diagram (PD), which can be seen as a scatter plot of birth and death values for each cycle. Instead of working directly with PDs, these are often represented by vectorized summaries that are better suited for machine learning frameworks. A common choice is a persistence image (PI) (discretized weighted 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). The PH pipeline is visualized in Figure 1.7, and each step is explained in greater detail in the next sections. For good choices of filtration and signature [337], there are theoretical results that guarantee that PH with respect to Wasserstein distance (Section 1.6) is stable under small perturbations of the input (Section 1.7).

To make the exposition as clear and concise as possible, we here provide only an intuitive description of geometry and isometries, topology and homeomorphisms, and homology, and focus in the remainder of this chapter on the main concepts in the thesis, namely (Vietoris-Rips and alpha) simplicial and cubical complex, filtration, PD, PI and PL. For a completely self-contained and precise treatment of PH, we would need to introduce the notions of (abelian) group, ring, field, simplicial chain, module, boundary maps, chain complex, homology group, persistence module, topology, topological and metric spaces. These abstract notions from algebraic topology are not needed for understanding the contributions of this thesis, and we therefore refer interested readers to [110, 153]. For a gentle, but detailed introduction to PH for a broad range of computational scientists, see [258].

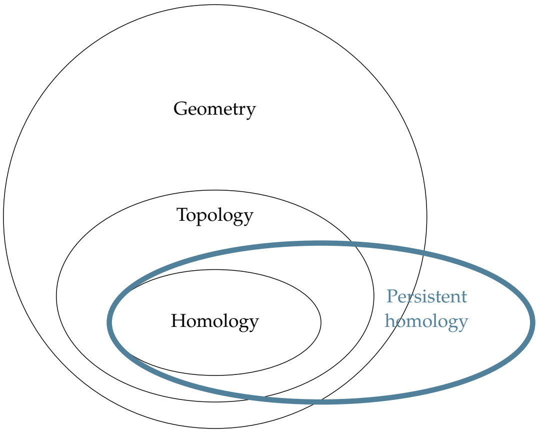


Figure 1.6.: Persistent homology captures information about the cycles in the data, and is shown to capture the homology of the underlying manifold, under some mild assumptions. However, it has been shown that persistent homology can also reveal the underlying dimension or curvature, thus providing additional topological and geometric information.

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

To approximate the shape of data, one can contruct a so-called structured complex, a type of topological space that is particularly important in algebraic topology due to its combinatorial nature that allows for the computation of homology. When the space is a point cloud $X \subset \mathbb{R}^d$, the most common choice for a structured complex is the simplicial complex, a set composed of simplices (points, line segments, triangles, tetrahedrons, and their k-dimensional counterparts), that is closed under taking subsets (so that, for instance, if a triangle is in the simplicial complex, then all its edges and vertices are also elements of the simplicial complex) [112, 258]. Formally, simplicial complexes are defined in the following way:

Definition 1.1 (Abstract simplicial complex)**.** Let X be a set. A collection $K \subset 2^X$ of subsets of X is called a simplicial complex if $\sigma \in K$ and $\tau \subset \sigma$ imply $\tau \in K$. An element $\sigma \in K$ is called a $(|\sigma| - 1)$-simplex, where $|\sigma|$ is the cardinality of σ.

A k-simplex may be realized geometrically as the convex hull of $k+1$ affinely independent points in $\mathbb{R}^d$, $d \geq k$. This gives us the familiar k-simplices: vertices, edges, triangles, etc. [81]. A simplicial complex can be seen as a generalization of a graph, where apart from nodes (0-simplices) and edges (1-simplices), it may also include triangles (2-simplices), tetrahedra (3-simplices), and so on. As we will see in more detail in the next section, the key idea behind PH is to construct a filtration, a sequence of topological spaces by gradually adding simplices to the data, and to study the evolution of topological features (components, holes, cavities,

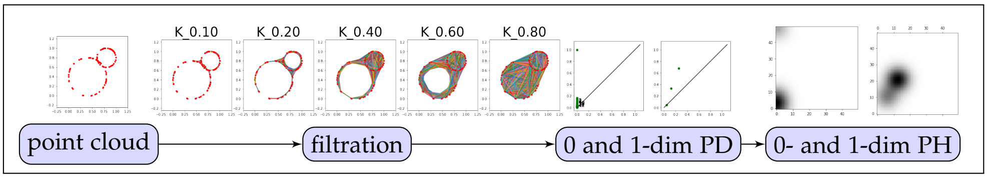


Figure 1.7.: Persistent homology (PH) pipeline. To calculate PH for the given point cloud X, we first construct in some way a filtration $\{K_r\}_{r\in\mathbb{R}}$ which approximates X at different scales $r \in \mathbb{R}$. 0-dimensional PD has one persistent cycle, reflecting the single component, and 1-dimensional PD reveals the two holes. A PD is often represented with some other signature (denoted, in general, with PH) that is more amenable for statistical (learning) frameworks, such as a PI.

and higher-dimensional voids) across the filtration. There are theoretical results that guarantee that there exists $r \in \mathbb{R}$ such that the homology of these complexes, constructed at scale $r \in \mathbb{R}$ on a dense-enough sample X from a manifold M, recover the underlying homology of M (for details, see Appendix A.1.2). In the remainder of this section, we introduce the three common structured complexes that we consider in this thesis.

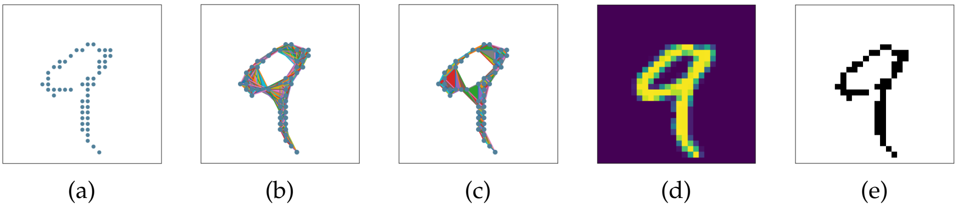

(a) (b) (c) (d) (e)

Figure 1.8.: The topology of a point cloud X (a) can be studied through the Vietoris-Rips (b) or the related alpha (c) simplicial complex that consist of vertices, edges, triangles, ..., and approximate X at some scale $r \in \mathbb{R}$. If X is an image (d), cubical complex (e), a union of vertices, edges, cubes, etc., is a more natural representation.

1.2.1 *Vietoris-Rips simplicial complex*

To approximate a point cloud X with n points (in any metric space), it is common to construct the Vietoris-Rips simplicial complex:

Definition 1.2 (Vietoris-Rips simplicial complex [348])**.** Let X be a metric space, and let $r \in \mathbb{R}$. The Vietoris-Rips simplicial complex of X at scale r is defined in the following way:

$$VR(X, r) = \{\sigma \subset X \mid d(x, y) \leq r \text{ for all } x, y \in \sigma\}.$$

In other words, $VR(X,r)$ is built by constructing (i) a line segment for any pair of points in X within distance r of each other, (ii) a triangle, if the points in a triplet are all within distance r of each other, and so forth (Figure 1.8).

1.2.2 *Alpha simplicial complex*

Since Vietoris-Rips simplicial complex is large [258], one might rather choose the alpha complex [114] (Figure 1.8), which is closely related to the Vietoris-Rips complex [187], but consists of significantly less simplices and is faster to construct when the dimension of the ambient space is 2 or 3 (for details, see Appendix A.1.4). The construction of the alpha complex relies on the Voronoi diagram that partitions a set of points in the Euclidean space into non-overlapping regions.[2] A precise definition is given below.

Definition 1.3 (Alpha simplicial complex [112])**.** Let $X \subset \mathbb{R}^d$, and let $r \in \mathbb{R}$. Let $B(x,r)$ further denote the ball centered at $x \in X$ with radius r, and $V(x)$ the Voronoi cell of $x \in X$, the set of points $y \in \mathbb{R}^d$ that are at least as close to x than to any other point $x' \in X$:

$$V_X(x) = \{y \in \mathbb{R}^d \mid \|y - x\|_2 \leq \|y - x'\|_2 \text{ for any } x' \in X\}.$$

The alpha simplicial complex of X at scale r is defined in the following way:

$$\alpha(X,r) = \{\sigma \subset X \mid \cap_{x \in \sigma} R(x,r) \neq \emptyset\}$$

where $R(x,r) = B(x,r) \cap V_X(x)$.

1.2.3 *Cubical complex*

If data is an image rather than a point cloud, other types of structured complexes might be more suitable. Let Z be a greyscale image, i.e., $Z = [z_{uv}]$, where z_{uv} is the greyscale value of the pixel (u,v), $u \in \{1,2,\ldots,n_x\}$, $v \in \{1,2,\ldots,n_y\}$, n_x and n_y are the numbers of pixels in respectively x and y direction. We can consider the image Z as a 2D point cloud $X(Z,z_0) \subset \mathbb{R}^2$ consisting of all $(u,v) \in \mathbb{R}^2$ corresponding to pixels with a greyscale value above a fixed user-given threshold z_0, and to approximate it with the Vietoris-Rips or alpha simplicial complex (Figure 1.8).

However, a point cloud (and its simplicial complex) is not the most intuitive representation of an image, as it does not exploit its natural grid structure [129]. For

[2] A nice video illustrating the alpha complex, and its difference to the Vietoris-Rips complex can be found at `https://www.youtube.com/watch?v=fnkvPy4ZCNY`.

an image Z, we can rather consider its so-called cubical complex $K(Z)$ (Figure 1.8), the cubical analog to a simplicial complex, in which the role of simplices is played by cubes of different dimension (points, line segments, squares, cubes, and their k-dimensional counterparts) [173]. The points correspond to the pixel corners, the edges to the sides of the pixels, and the squares to the image pixels.[3] More formally, the cubical complex is defined in the following way.

Definition 1.4 (Cubical complex [172])**.** A cubical complex is a union of elementary cubes $I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d$, where I_i is an elementary interval $I_i = [l, l]$ or $I_i = [l, l+1]$ for some $l \in \mathbb{Z}$.

1.3 Filtration, input for persistent homology

In the previous section, we discuss the different ways we can approximate the shape of data X at a scale $r \in \mathbb{R}$. However, different values of the resolution parameter r create different simplicial complexes and reveal different cycles (Figure 1.4). To circumvent this issue, the calculation of PH relies on a so-called filtration that describes X at *any* scale:

Definition 1.5 (Filtration)**.** A filtration is a family of nested topological spaces $\{K_r\}_{r \in \mathbb{R}}$.

Typically, every K_r approximates X at scale $r \in \mathbb{R}$ in some suitable sense. In this way, it provides a lens to look at X. One of the most common ways to build a filtration is via structured complexes, i.e., every K_r is a simplicial or cubical complex. These can be seen as sublevel sets of a filtration function $\phi : 2^X \to \mathbb{R}$, determined by a scale cut-off $r \in \mathbb{R}$:

$$K_r = \{\sigma \subset X \mid \phi(\sigma) \leq r\}.$$

In the next subsections, we discuss some common filtrations: Vietoris-Rips, Distance-to-Measure (DTM) and height, via simplicial or cubical complexes (Figure 1.9). However, depending on the task, there are many other filtrations one could choose, such as rank [273], radial, erosion or dilation [129]. The resulting PH captures completely different information about the cycles [337]. For example, whereas PH with respect to the Vietoris-Rips filtration encodes the size of the hole, PH on the height filtration informs about the position of the hole. For a more detailed discussion about the influence of filtration, see Chapter 2 (and in particular, the associated Appendix A.6) and Chapter 3.

[3] Another way to construct a cubical complex from an image is to consider the dual of the cubical complex defined above: the points reflect the pixels, the line segments the intersections of pairs of non-diagonally neighboring pixels, and squares reflect the intersections of four pixels [320].

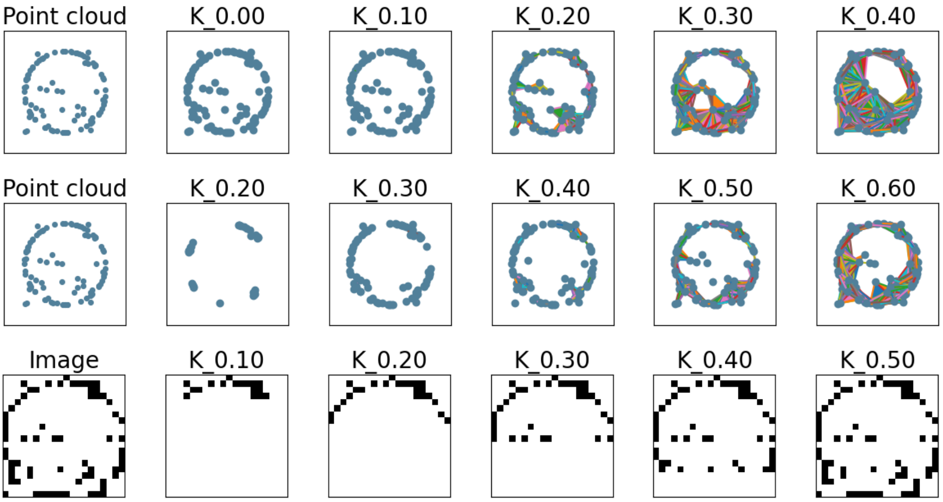


Figure 1.9.: Filtration $\{K_r\}_{r\in\mathbb{R}}$ approximates the data X (e.g., a point cloud or image) across different scales r. The common choice for K_r is the Vietoris-Rips simplicial complex (top row), where two vertices x_i and x_j are connected with an edge if $d(x_i, x_j) \ \leq r$. One can also construct a weighted Vietoris-Rips simplicial complex, where K_r does not need to include all vertices: for instance, Distance-to-Measure (DTM, middle row) filtration function assigns to each vertex the average distance from a given number of nearest neighbors; in this way, the outliers are smoothed out as they appear later in the filtration. Other choices of filtration functions (on simplicial or cubical complexes) are possible, such as the height filtration (bottom row) which allows to" scan" the point cloud or image e.g., from top to bottom.

1.3.1 *Vietoris-Rips filtration*

The Vietoris-Rips filtration of a point cloud $X \subset \mathbb{R}^d$ is the nested family of simplicial complexes $\{VR(X,r)\}_{r\in\mathbb{R}}$ (Definition 1.2) (Figure 1.9, top row). The underlying filtration function $\phi : 2^X \to \mathbb{R}$ assigns to each simplex its diameter:

$$\phi(\sigma) = \max_{x,y\in\sigma} d(x,y).$$

The Vietoris-Rips filtration is related to the sublevel sets of the distance function $\delta_X : \mathbb{R}^d \to \mathbb{R}$, where $\delta_X(y) = \min\{d(y,x) \mid x \in X\}$ is the distance to point cloud X. Indeed, the Vietoris-Rips simplicial complex $VR(X,r)$ approximates the sublevel set

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

where $B(x, r)$ is a ball with radius r centered around $x \in X$ [64].

1.3.2 *Distance-to-Measure (DTM) filtration*

The distance function δ_X underlying the common Vietoris-Rips filtration discussed in the previous section is extremely sensitive to outliers ("even one outlier is deadly", or, in the language of robust statistics, the distance function has breakdown point zero [67]). To circumvent this issue, [10, 64] propose to rather consider distance-to-a-measure (DTM) function $\delta_{X,m} : \mathbb{R}^d \rightarrow \mathbb{R}$, where $\delta_{X,m}(y)$ is defined as the average distance between y and a given number of nearest neighbors in X (and is thus a smooth version of the distance function). In this way, the outliers have a large DTM function value and appear late in the filtration (Figure 1.9, middle row).

The filtration function $\phi : 2^X \rightarrow \mathbb{R}$ (for any so-called weighted Rips filtrat) is defined as $\phi(\{x_i\}) = \delta_{X,m}(x_i)$ for vertices, $\phi(\{x_i, x_j\}) = \phi(\{x_i\}) + \phi(\{x_j\}) + d(x_i, x_j)/2$ for edges, and for simplices with degree greater than one as $\phi(\sigma) = \max_{x_i, x_j \in \sigma}\{\phi(\{x_i, x_j\})\}$. The number of neighbors that are considered is determined by the parameter m, which represents the percentage of the total number of point cloud X points.

1.3.3 *Height filtration*

Similarly to the DTM filtration discussed in the previous subsection, we can consider the so-called weighted Rips filtration that defines filtration function values on the point cloud vertices via an arbitrary function. A useful example is the height filtration function that assigns to each point the distance from some given hyperplane, as it allows to "scan" the object of interest for some direction. More precisely, for a given unit vector $v \in \mathbb{R}^d$, we define $\phi(\{x\}) = \langle x, v \rangle$.

This results in a filtered *simplicial* complex of a point cloud, but a *cubical* filtration with respect to the height function is often more reasonable for images (Figure 1.9, bottom row), see Section 1.3.4 for a precise description. For an illustration of the two approaches on an example point cloud and corresponding image, see Figure A.13.

1.3.4 *Cubical filtration*

To define a filtration function $\phi : K(Z) \rightarrow \mathbb{R}$ for an image $Z = [z_{uv}]$, one needs to define the value of ϕ on each cube in the cubical complex $K(Z)$ (Section 1.2.3). A natural filtration function on a cubical complex assigns to each square the value $\phi(u, v)$ of the image on the corresponding pixel (u, v).. The filtration function on the line segments and points is defined as the minimum value of all bordering pixels.[4]

[4] A natural filtration function on the dual cubical complex assigns the pixel values as the values of the function on the points, and sets the function values for line segments and squares as the maximum value of all bordering simplices. These two methods differ with respect to the diagonally neighboring pixels, as

The cubical filtration is then a nested family $\{K_r\}_{r\in\mathbb{R}}$ of cubical complexes K_r, where K_r is the union of all cubes corresponding to pixels (u, v) with $\phi(u, v) \leq r$ (Figure 1.9, bottom row). Note that this means that (the cubes corresponding to) the pixels with the lowest filtration function value appear first and persist the longest in the filtration.

1.4 Persistence signature, output of persistent homology

Given a filtration $\{K_r\}_{r\in\mathbb{R}}$, persistent homology (PH) detects how the homology of the complexes K_r changes as the parameter value r increases, capturing how the features "persist" across changes in resolution. As already mentioned, k-dimensional PH is a multi-set of intervals (b_i, d_i), with b_i and d_i corresponding to the scale $r \in \mathbb{R}$ when a k-dimensional cycle i appears and disappears in the filtration ($d_i = +\infty$ if the cycle persists forever). In computational settings, there are always only finitely many persistence intervals, and the intervals are truncated to the maximum filtration value. Throughout this thesis, we thus assume PH consists of finitely many intervals (b, d) with $0 \leq b \leq d < +\infty$. 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.

A natural way to represent PH is via a PD, roughly a scatter plot of (b, d) points, with their multiplicity (a precise definition is given in the next subsection). However, PDs lack structure and are not amenable to statistics and machine learnings (e.g., PDs do not have a unique mean [45]). Moreover, Wasserstein distance between PDs (Section 1.6) suffers from combinatorial explosion, growing quickly with the number of persistence intervals [99]. It is therefore of interest to consider different PH signatures, and in the following subsections we discuss the common vectorized summaries, persistence image (PI) and persistence landscape (PL) (Figure 1.10).

Note that a plethora of persistence signatures have been introduced in the literature, e.g., Betti numbers (across scales) [163, 341] or Euler characteristic, the difference between the number of connected components and the number of holes (across scales) [210]. Some PH signatures summarize the same information, but lie in different metric spaces [340]. As already indicated, PH can also be summarized with a scalar, for instance with: amplitude, distance from the empty PD [129]; entropy, a real number calculated using the lifespans of all features [129], which thus only depends on the persistence but not on the particular birth or death times; or an algebraic function of b_i and $d_i - b_i$, e.g., $\sum b_i^p (d_i - b_i)^q$, so that p and q determine the importance of some of the qualities about cycles (e.g., size of holes) [4]. To avoid choosing among the "zoo of persistence signatures" [340], it is possible to learn the best vector summary (with e.g., PersLay, a simple neural network layer [56], or ATOL, an unsupervised vectorization method [304]).

they are considered connected with the first approach, but not the second, what can result in substantially different persistent homology.

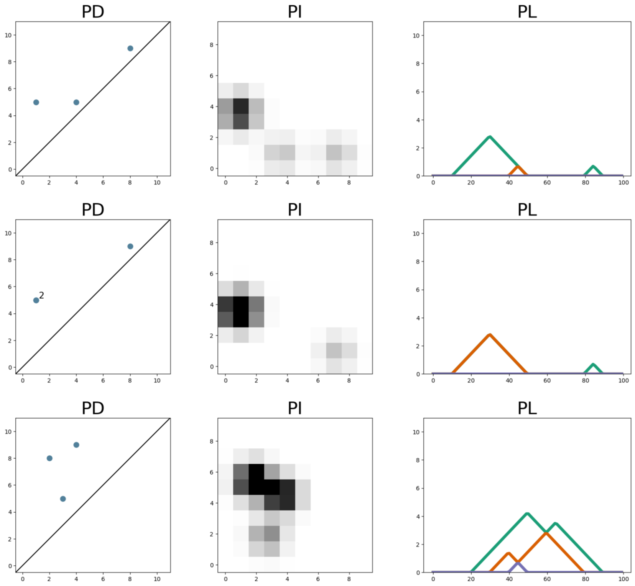


Figure 1.10.: Three common signatures for persistent homology (PH) are persistence diagram (PD), persistence image (PI) and persistence landscape (PL). PD is a scatter plot of birth and death values (b_i, d_i), with their multiplicity, including the diagonal (left column). PI reflects the volume below the weighted sum of Gaussian probability density functions evaluated on a grid superimposed over a PD, where the weight of each PD point can e.g., reflect the the persistence $l_i = d_i - b_i$. PL is a sequence of functions $\mathbb{R} \to \mathbb{R}$ obtained by stacking isoscles triangles over the midpoints of (b_i, d_i) intervals, with the height of the triangles reflecting the persistence $l_i = d_i - b_i$.

1.4.1 *Persistence diagram*

Persistence diagram (PD) is the most straightforward representation of persistent homology (PH), which can be seen as a scatter plot of points (b_i, d_i), counted with their multiplicity, and union all points on the diagonal, counted with infinite multiplicity (Figure 1.10):

Definition 1.6 (Persistence diagram (PD) [77, 114])**.** Let $\{K_r\}_{r \in \mathbb{R}}$ be a filtration, and let k-dimensional persistent homology be the multi-set of intervals (b_i, d_i), with b_i and d_i being the scale r when each k-dimensional cycle i is born, and when it dies, within the filtration. The k-th dimensional persistence diagram PD is the union of a finite multi-set of points in $(b_i, d_i) \in \mathbb{R}^2$, and multi-set of points on the diagonal $\{(x, y) \in \mathbb{R}^2 \mid x = y\}$, where each point on the diagonal has infinite multiplicity.

The x and y axis in the plot, i.e., the birth and death values b and d depict 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 points on the diagonal are included with infinite multiplicity for technical reasons, namely, to ensure the existence of a bijection between any two PDs, that is needed for a definition of the Wasserstein metric (Section 1.6). The distance of a PD point (b, d) from the diagonal reflects the persistence or lifespan $l = d - b$ of the corresponding cycle.

An advantage of PDs compared to other persistence signatures is that they are parameter-free, but they also have an important disadvantage: they are not convenient for statistical inference, because their complicated structure makes common algebraic operations - such as addition, division, and multiplication - challenging [31] (so that, for instance, the mean might not be unique [232]). Furthermore, although PDs can be endowed with a metric structure (Section 1.6) which enables to perform some machine learning techniques such as some clustering algorithms, PDs do not lend themselves to processing with techniques that require a Hilbert space structure, including support vector machine (SVM) and principal component analysis (PCA) [291]. It is therefore common in applications to consider vector representations of PDs, and we discuss some popular choices in the remaining subsections.

1.4.2 *Persistence image*

Persistence image (PI) is constructed by superimposing a grid over a persistence diagram (PD), and depicting the volume below the weighted sum of (Gaussian) probability density functions, on each grid cell (Figure 1.10). This is a more sophisticated variant of counting the number of cycles in each of the grid bins [302]. More precisely, the definition is as follows.

Definition 1.7 (Persistence image (PI) [2]). Let PD be a multi-set of persistence intervals (b_i, d_i). Let $\eta : \mathbb{R}^2 \to \mathbb{R}^2$ be the linear map $\eta(x, y) = (x, y - x)$, $g_{\mu_x, \mu_y} : \mathbb{R}^2 \to \mathbb{R}$ a differentiable probability distribution with mean $(\mu_x, \mu_y) \in \eta(PD)$, and $\alpha : \mathbb{R}^2 \to \mathbb{R}^2$ non-negative weight function that is zero along the horizontal axis (reflecting the PD diagonal), continuous, and piecewise differentiable. The persistence surface is the function $\psi \colon \mathbb{R}^2 \to \mathbb{R}$ defined by

$$\psi(x, y) = \sum_{(b,l) \in \eta(\mathrm{PD})} \alpha(b, l) g_{(b,l)}(x, y).$$

The persistence image (PI) is the volume below the persistence surface, over each region A in the discretization of a relevant subdomain of ψ :

$$PI(A) = \iint_A \psi(x, y) dxdy.$$

In applications, one commonly assumes Gaussian probability density function with some given variance σ:

$$g_{(b,l)}(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-[(x-b)^2+(y-l)^2]}{2\sigma^2}},$$

and for some resolution $P \in \mathbb{N}$, discretizes the sub-domain of ψ into a $P \times P$ grid (an image, with regions A corresponding to image pixels), which can be further transformed into a vector of length $P \times P$.

In other words, the multi-set of birth-death values (b,d) is first transformed into birth-lifespan $(b,l) = (b,d-b)$ coordinates, what makes sense since there are no points below the PD diagonal. A Gaussian probability density function is centered at each point (b,l), and persistence surface ψ is calculated as a weighted sum of these functions. The heights of the peaks in ψ are thus influenced by some given non-negative weight function $\alpha : \mathbb{R}^2 \to \mathbb{R}$. Typically, α reflects some information about the cycles, and it usually depends only on the vertical persistence coordinate l (corresponding to the lifespan of the cycle, $l = d-b$), e.g., $\alpha(b,l) = l$ or $\alpha(b,l) = l^2$. Then, one splits a subdomain of ψ into a $P \times P$ grid of regions. PI is the matrix whose (i,j)-th entry or pixel is the integration value of ψ over the (i,j)-th region. Obviously, one can combine PH information across multiple homological dimensions by concatenating the PI vectors for each dimension, and provide this as input for machine learning algorithms.

For constructing PIs, one thus needs to choose 1) the kernel function $g_{(b,l)}(x,y)$ and its associated parameters, 2) the weight function $\alpha(b,l)$, and 3) the resolution P. An advantage of PIs is their flexibility, e.g., different choices of the weight function allow to highlight different types of cycles, such as persistent cycles (with large persistence or lifespan), or cycles that are born early (with low birth values), or even cycles that are short-lived. The three parameters are also a weakness, as their choice is non-canonical [2]. However, [2, 373] demonstrate that for their machine learning tasks, PIs have low sensitivity to the parameter choices of resolution and distribution variance, which indicates the utility of PIs even with no prior knowledge about the underlying data.

1.4.3 *Persistence landscape*

Persistence landscape (PL) can be seen as a sequence of functions obtained by "stacking isosceles triangles" whose bases are the persistent homology (PH) intervals (b_i, d_i), and whose heights reflect the lifespans or persistence $l_i = d_i - b_i$ (Figure 1.10):

Definition 1.8 (Persistence landscape (PL) [44, 45]). Let PD be a multi-set of persistence intervals (b_i, d_i). For each cycle (b,d), let $g_{(b,d)} : \mathbb{R} \to [0, +\infty)$ be the following piecewise linear function:

$$g_{(b,d)}(x) = \begin{cases} 0 & x \notin (b,d) \\ x-b & x \in (b, \frac{b+d}{2}] \\ d-x & x \in (\frac{b+d}{2}, d] \end{cases}$$

Persistence landscape (PL) is a function $\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$, where $\lambda(j,x)$ is the j-th largest value of $\{g_{(b_i,d_i)}(x)\}_i$. Alternatively, it may be thought of as a sequence of functions $\lambda_j : \mathbb{R} \to \mathbb{R}$, where $\lambda_j(x) = \lambda(j,x)$. In applications, PL is a vector of length $N \times P$, concatenating values of N first λ_i functions, evaluated at P equidistant values within the relevant interval in $\mathbb{R}$.

In contrast to PDs, PLs lie in a Banach space, and are thus easy to combine with tools from statistics: they obey a strong law of large numbers and a central limit theorem, and the space of PLs does have a unique mean [44]. However, for many machine learning tasks, it is necessary to consider finite vectors rather than functions, and a discretization of the function λ into a vector requires two additional parameters: we need to decide on the maximum number of first landscape functions λ_j to consider, and on the number of points where each of these functions is evaluated, referred to as the landscape resolution.

1.5 Computation of persistent homology

Computiation of persistent homology (PH) boils down to linear algebra [258]. For a given filtered complex with N simplices, one first constructs a binary $N \times N$ matrix which encodes the boundary information for each simplex. Matrix reduction via Gaussian elimination allows to simply read off the birth and death times of each cycle (see [258, Section 5.3.2 and Figure 7] for details and a nice toy example). In the worst case, the computational complexity of this algorithm is $\mathcal{O}(N^3)$ [384]. This bound is sharp, as Morozov gave an example of a complex with cubic complexity [243]. A comprehensive benchmarking of the state-of-the-art implementations for the computation of PH is done in [258].

1.6 Distances for persistent homology

Different metrics can be considered on the space of any persistence signatures. The most common distance between PDs is the Wasserstein distance:

Definition 1.9 (Wasserstein distance [320])**.** Let $PD_1, PD2$ be two persistence diagrams, and let $p, q \in [1, +\infty)$. The p-th Wasserstein distance between the persistence diagrams is defined as:

$$W_{p,q}(PD_1, PD_2) = \inf_{\tau} \Big(\sum_i \|(b_i, d_i) - \tau(b_i, d_i)\|_q^p \Big)^{\frac{1}{p}}, \tag{1.1}$$

where the infimum is taken across all bijections[5] $\tau : PD_1 \to PD_2$, and the sum across all persistence intervals $(b_i, d_i) \in PD_1$. The bottleneck distance for $p = +\infty$ is defined as

$$W_{\infty,q}(PD_1, PD_2) = \inf_{\tau} \sup_i \|(b_i, d_i) - \tau(b_i, d_i)\|_q^p. \tag{1.2}$$

One commonly assumes that $q = \infty$, and denotes $W_p = W_{p,\infty}$ for any $p \in [1, +\infty]$.[6]

As suggested by the notation, the bottleneck distance is the limit of the Wasserstein distance for p going to infinity [112]. Wasserstein distance is popular in computer vision [78], and it is the common metric for optimal transportation problem [180] (with a bijection τ from PD_1 to PD_2 corresponding to a transport plan). For the vectorized persistence signatures, such as PIs and PLs, $l_p = \| \cdot \|_p$ distances are common. The parameter p in both W_p and l_p determines the importance of long compared to short distances.

1.7 Stability theorems

Stability theorems are among the most important results in applied and computational topology [320], as they may be viewed as a precise statement about sensitivity to some type of noise: stable representations of PH are robust under small perturbations in the input (Figure 1.11). Stability results can be grouped into theorems for PDs, and for vector summaries of PH: below we provide a *general* statement for such theorems. A precise statement is explicit about the constant on the right-hand side, which can depend on the filtration function, distance parameter p, and the parameters of the PH signature of interest.

Theorem 1.10 (Stability theorem for persistence diagrams (PDs))**.** *Let X be a triangulable[7] topological space with continuous tame[8] functions $\phi, \psi : X \to \mathbb{R}$. There exists a constant $c \in \mathbb{R}$ such that:*

[5] There exists a bijection between any two PDs, since a PD includes the diagonal with infinite multiplicity (see Definition 1.6).

[6] A video lecture by Katharine Turner that counts the ways why it is more reasonable to rather set $q = p$ can be found at `https://youtu.be/d_iqovKui6k?si=cRnQl0dsnRlKlFCe`.

[7] Triangulable space is homeomorphic to a simplicial complex, and it enables to replace topological with piecewise linear spaces.

[8] Tame functions have only finitely many critical values where homology groups change, ensuring a finite number of persistence intervals.

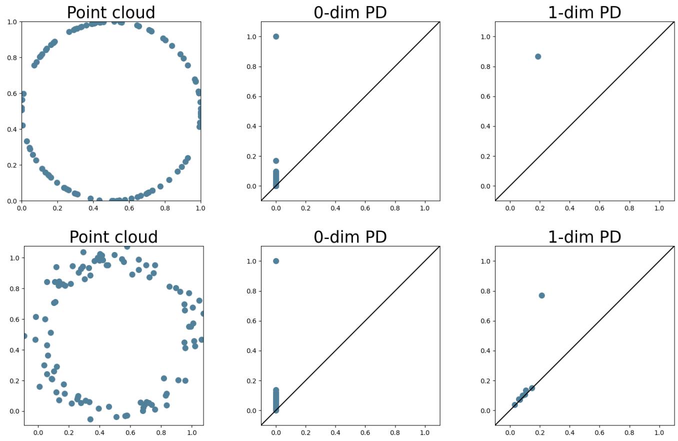


Figure 1.11.: Stability theorems imply that the persistent homology (PH) of a point cloud (top row) and its noisy version (bottom row) remain similar. For example, 1-dimensional PH of the noisy point cloud captures the loop, together with a number of additional loops that might briefly appear in the filtration, resulting in short persistence intervals (close to the PD diagonal).

$$W_p(PD(\phi), PD(\psi)) \leq c\|\phi - \psi\|_p.$$

The stability of PDs was first proved for $p = \infty$ (the easiest case, since W_∞ is the least sensitive to details in the diagrams [112]) [62, 77, 102]. In this case, the right-hand side for the Vietoris-Rips filtration (Section 1.3.1), $\|\delta_X - \delta_Y\|$, corresponds to the Hausdorff distance $d_H(X, Y)$ between the point clouds. [9] A few years later, the stability was shown to hold for large enough p and under additional assumptions [78], and recently, for any p [320].

Theorem 1.11 (Stability theorem for vectorized persistent homology (PH))**.** *Let X be a triangulable topological space with continuous tame functions $\phi, \psi : X \to \mathbb{R}$, and let PH be a vectorized summary of PD. There exists a constant $c \in \mathbb{R}$ such that:*

$$\|PH(PD(\phi)) - PH(PD(\psi))\|_p \leq cW_p(PD(\phi), PD(\psi)).$$

Persistence landscapes are shown to be stable for large enough p [44, Theorem 13, Theorem 16], but this fails to be true for $p = 2$ [320, Theorem 7.7]. Stability

[9] Actually, for the Rips filtration, [63, Theorem 3.1] show that $W_p(PD(\phi), PD(\psi)) \leq cd_{GH}(X, Y)$, where d_{GH} is the Gromov-Hausdorff distance. This is useful since $d_{GH}(X, Y) \leq d_H(X, Y)$; for instance, if Y is obtained from X by rotation (or any other isometric transformation), $d_H(X, Y)$ can be large, whereas $d_{GH}(X, Y) = 0$ and is therefore a more useful upper bound.

of persistence images holds for $p = 1$ [2], but not for $p = 2$ [291, Theorem 3], [2, Remark 6].

Stability of PH connects to manifold reconstruction 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? The theorems in this section ensure that PH can uncover the homology of a manifold M from a finite set of possibly inaccurate point samples [77].

Part I

ANALYSIS OF PERSISTENT HOMOLOGY

2

THE WHAT:
On the effectiveness of persistent homology

EVEN though persistent homology (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 chapter is to identify some types of problems where PH performs on par or better than other 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. For convexity detection, we provide a theoretical guarantee that PH is effective for this task in $\mathbb{R}^d$, and demonstrate the detection of a convexity measure on the FLAVIA data set of plant leaf images. Due to the crucial role of shape classification in understanding mathematical and physical structures and objects, and in many applications, the findings will provide some knowledge about the types of problems that are appropriate for PH, so that it can — to borrow the words from Wigner 1960 — "remain valid in future research, and extend, to our pleasure", but to our lesser bafflement, to a variety of applications.

This chapter is based on the following publication:

> Renata Turkeš, Guido Montúfar, and Nina Otter, *On the effectiveness of persistent homology*, Advances in Neural Information Processing Systems 35 (NeurIPS 2022): 35432-35448.

The research was supported by Fulbright, and respectively the ERC, DFG, NSF-CAREER and the Royal Society grants 757983, 464109215, DMS-2145630, and RGS\R2\212169.

2.1 Introduction

Successful applications of persistent homology (PH) (Chapter 1) range from prediction of biomolecular properties [51, 52, 353], face, gait and activity recognition [168, 195, 205, 383] or digital forensics [14], to discriminating breast-cancer subtypes [319] or quantifying the porosity of nanoporous materials [204]. At the same time, the reasons behind these successes are not yet well understood. Indeed, the data used in real-world applications is complex, so that there are numerous effects at play and one is often left unsure why PH worked, i.e., what type of topological or geometric information it captured that facilitated the good performance.

The title of this chapter is inspired by a famous paper from 1960, "The unreasonable effectiveness of mathematics in the natural sciences" [358], in which Wigner discusses, with wonder, how mathematical concepts have applicability far beyond the context in which they were originally developed. The same, we believe, is true for persistent homology. While this method has been applied successfully to a wide range of application problems, we believe that for PH to remain relevant, there is a need to better understand why it is so successful. Thus, we distinguish between the *usefulness* of PH for applications, which has been attested in hundreds of applications and publications, and its *effectiveness*, namely that PH is capable of producing an intended or desired result. Thus, here we initiate an investigation into the effectiveness of PH, or in other words, 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 [38, 40, 61, 67]? Obtaining a representation of a shape that can be used in statistical models is an important task in data analysis and numerous approaches to modeling surfaces and shapes [339].

To pursue our investigation, we set out to identify some fundamental data-analysis tasks that can be solved with PH. Since PH is inspired by homology, which provides a measure for the number of components, holes, voids, and higher-dimensional cycles of a space (Chapter 1) — 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, and can thereby capture geometric information, such as the size or position of holes. We therefore also investigate how well PH can detect fundamental geometric notions of ***curvature*** and ***convexity***. For each of the three problems, we first discuss theoretical results that provide a guarantee that PH can solve these tasks. Detection of convexity with PH has not been investigated in the literature to date, and we prove a new result.

To investigate how well the PH pipeline works in practice, we compare its performance against several baselines on synthetic point-cloud data sets. As a first machine learning (ML) baseline we take an 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 [145, 282] is a deep learning architecture designed specifically for point cloud data. Similar architectures with convolutional (and fully connected and pooling) layers have been applied for Betti-number and curvature estimation [146, 266]. For convexity detection, we also evaluate the performance of PH on real-world data. The theoretical guarantees above imply that the results for PH would generalize to new data.

Finally, we note that our goal is not to claim the superiority of PH compared to other approaches in the literature, in particular, with state-of-the-art methods for each of the problems. We do not necessarily expect that on well-specified mathematical problems PH will beat state-of-the-art algorithms that have been specifically designed for those tasks. Instead, what we think is interesting and remarkable is that PH can in fact solve tasks it is not specifically or uniquely designed for. Moreover, an advantage of PH is that it can reveal, e.g., both topology and curvature at the same time, avoiding the need to employ and combine state-of-the-art models for each of the tasks.

RELATED WORK In spite of the growing interest in PH, so far there is only limited work in the direction that we pursue here. 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, the sample density and closeness to the space) [70, 187, 255], and there is significant interest in investigating how well this works in practice [66]. However, so far there are only few available results. Some works demonstrate that PH can be used to detect the number of holes, but only on individual toy examples (e.g. [300], [61, Figure 19], [193, Figures 9-20], [67, Figures 2, 3, 6, 11, 12]) without looking into the statistical significance between different classes of data, or the accuracy of some classification algorithms on a comprehensive data set. There are also some works where PH is used to estimate the Betti numbers on a possibly larger data set, but only with the goal of using this information to e.g., study the behavior of deep neural networks [251] or ensure topologically correct dimensionality reduction [265] or image segmentation [158], so that the soundness of this estimation is not investigated, which is the focus of our work.

Some insights about PH and curvature have been obtained in the literature, starting with an illustrative example in [81, 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, [46] show both theoretically and experimentally that PH can predict curvature (with computational experiments replicated in [349]), 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 and Dłotko [45] show 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 of a metric space (an isometric invariant) [257]. There have been several efforts in using PH to estimate fractal dimensions, such as [311] 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 prove that PH can detect convexity in $\mathbb{R}^d$ (Theorem 2.2).
- We define a new tubular filtration function (medium through which PH is extracted from data), that is crucial for the detection of convexity (Definition 2.1).
- We demonstrate experimentally that PH can detect the number of holes (Section 2.2), curvature (Section 2.3), and convexity (Section 2.4) from synthetic 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. For convexity detection, we also show that PH obtains a good performance on a real-world data set of plant leaf images.
- We demonstrate experimentally that PH features allow to solve the above tasks even in the case of limited training data (Section 2.2), noisy (Section 2.2) and out-of-distribution (Section 2.4) test data, and limited computational resources (Section 2.2, Section 2.3, Section 2.4).
- We provide insights about the topological and geometric features that are captured with long and short persistence intervals (Section 2.5), and formulate guidelines for applications that are suitable for PH (Section 2.6).
- 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.

2.2 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 the number of holes, inside versus outside, and connectivity can be effective primitives for recognizing shapes [278]. Extracting such topological information can therefore prove useful for many computer vision tasks. There are theoretical results in the literature that ensure that PH with respect to the alpha simplicial complex is successful for this problem, under some assumptions (Appendix A.1.2), and the computational experiments that follow demonstrate this success in practice.

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 consisting of 1 000 points sampled from a uniform distribution over the shape, resulting in a balanced data set of $1\,000 = 20 \times 50$ point clouds. A few examples of these point clouds are shown in Figure 2.1. The label of a point cloud is the number of holes in the underlying shape.

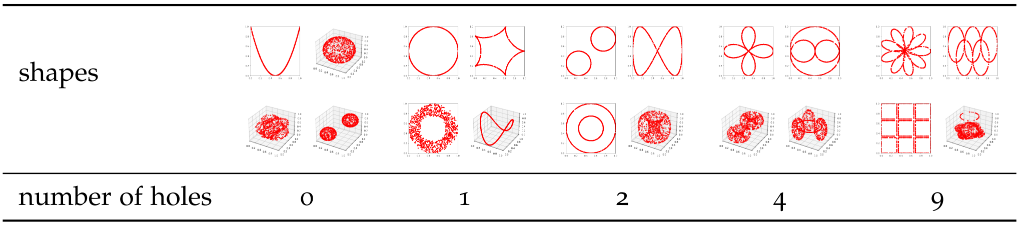


Figure 2.1.: Number of holes data set.

PH PIPELINE For each point cloud X and scale $r \in \mathbb{R}$, we consider its alpha complex (Section 1.2.2). 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 (Section 1.3.2). We extract 1-dimensional PDs, that are then transformed to PIs, PLs (Section 1.4), or a simple signature consisting only of lifespans $l = d - b$ of the 10 most persisting cycles (as there are at maximum 9 holes of interest in the given data set)[1], and classified with an SVM. We consider a "PH simple" pipeline, which relies on the 10 lifespans, and a "PH" pipeline wherein grid search is employed to choose the best out of the three aforementioned persistent signatures and the values of their parameters. For more details on the pipeline, see Appendix A.2.2 and Appendix A.3.2.

[1] 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 transformations (translation, rotation, stretch, shear) and two types of noise (Gaussian noise, outliers). For more details on these transformations, see Appendix A.3.1. We train the classifier on 80% of the original point clouds, and test on the remaining 20% of the data either in its original form or subject to transformations and noise. The results reported in Figure 2.2 (with detailed results across multiple runs in Appendix A.3.3) show that PH obtains very good test accuracy on this classification task, even in the presence of affine transformations or noise, outperforming baseline machine- and deep-learning techniques.[2] We reach a similar conclusion in case of limited training data and computational resources. Firstly, the evolution of the test accuracy across different amounts of training data demonstrates that PH achieves good performance for a small number of training point clouds, which is not the case for other pipelines (Appendix A.3.5). Secondly, although the hyperparameter tuning of the PH pipeline does take time (as we consider a wide range of parameters for the different persistence signatures), it is still less than for PointNet (Appendix A.3.6). Moreover, Figure 2.2 shows that even the simple PH pipeline, where the SVM is used directly on the lifespans of the 10 most persisting cycles (without any tuning of PH-related parameters) performs well.

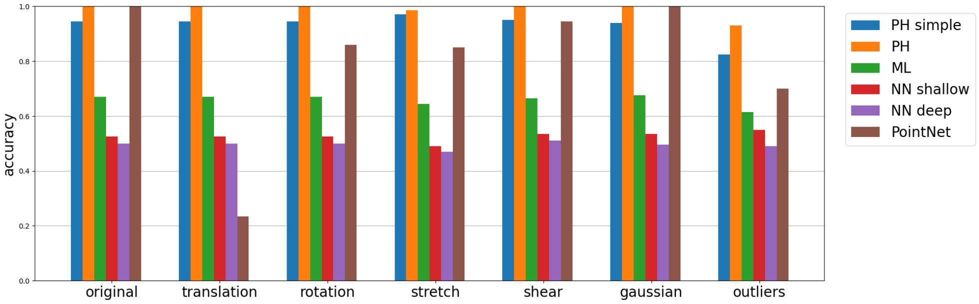


Figure 2.2.: Persistent homology can detect the number of holes.

2.3 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 [59]. For continuous surfaces,

[2] 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 [200, 354, 366, 375–378]), 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 $\beta_0 = 1$) [251]. Of course, the choice of activation function and hyperparameters might have an important influence on performance [251].

normals and curvature are fundamental geometric notions that uniquely characterize local geometry up to rigid transformations [146]. Recently, it has been shown that, using PH, curvature can be both recovered in theory (Appendix A.1.3), and effectively estimated in practice [46]. We run a similar experiment, evaluating the PH pipeline against our baselines, and also taking a closer look into the importance of short intervals.

DATA A balanced data set is generated in the same way as in [46]: We consider unit disks D_κ on surfaces of constant curvature κ: (i) $\kappa = 0$, Euclidean plane, (ii) $\kappa > 0$, sphere with radius $1/\sqrt{\kappa}$, and (iii) $\kappa < 0$, Poincaré disk model of the hyperbolic plane. Curvature κ 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 κ (as it spherical cap). For each $\kappa \in \{-2, -1.96, \ldots, -0.04, 0, 0.04, \ldots, 1.96\}$, we construct 10 point clouds by sampling 500 points from the unit disk D_κ with the probability measure proportional to the surface area measure [44, Section 2.7, Section 4.1]. A few examples with $\kappa \in \{-2, -1, -0.1, 0, 0.1, 1, 2\}$ are illustrated in Figure 2.3[3].These $101 \times 10 = 1\,010$ point clouds are considered as the training data, whereas the test data set is built in a similar way for 100 values of κ chosen uniformly at random from $[-2, 2]$. The label of a point cloud is the curvature κ of the underlying disk D_κ. Note that all these disks are homeomorphic: they are contractible, so that their homology is trivial and homology is thus unable to distinguish between them [46].

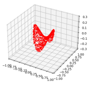
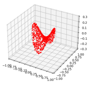
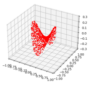
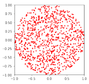
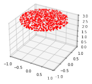
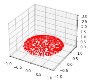
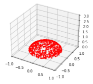
shapes							
curvature	-2.00	-1.00	-0.10	0.00	0.10	1.00	2.00

Figure 2.3.: Curvature data set.

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 [44, Section 2.7]. The input for PH is the filtered Vietoris-Rips simplicial complex (Section 1.3.1).[4] We extract 0- and 1-dimensional PDs, which are then transformed into PIs, PLs (Section 1.4) or lifespans, to be fed to SVM. More details on the pipeline are provided in Appendix A.2.2 and Appendix A.4.1.

[3] The unit disks with negative curvature are here visualized on hyperbolic paraboloids. These saddle surfaces have non-constant curvature, but they locally resemble the hyperbolic plane.

[4] Alpha complex is faster to compute, but involves Delaunay triangulation, whose unique existence is guaranteed only in Euclidean spaces. To calculate PH, we rely on the Ripser software [335], which is at the time the most efficient library to compute PH with Vietoris-Rips complex [258].

RESULTS Figure 2.4 shows the mean squared errors for the PH and other pipelines, together with their regression lines, with detailed results across multiple runs listed in Appendix A.4.2. The results show that PH indeed detects curvature, outperforming other methods.[5] Next to the PH pipeline discussed above, wherein a grid search is used to tune the parameters (Appendix A.2.2), we also consider SVM on the lists of lifespans of all persistence intervals (PH simple), and SVM only on the 10 longest lifespans (PH simple 10), in order to investigate if it is only the longest intervals that contribute to prediction. We see that the performance drops if we only focus on the longest 10 intervals, so that the many short intervals together capture the geometry of interest for this problem. Similarly as in Section 2.2, the grid search across the different parameters for persistence signatures does take time (Appendix A.4.3), but Figure 2.4 shows that SVM on a simple signature of all (0-dimensional) lifespans performs well. We highlight that the data used here, as was the data in Bubenik's work [46], is sampled from surfaces with *constant* curvature. In future work it would be interesting to conduct similar experiments on shapes with non-constant curvature.

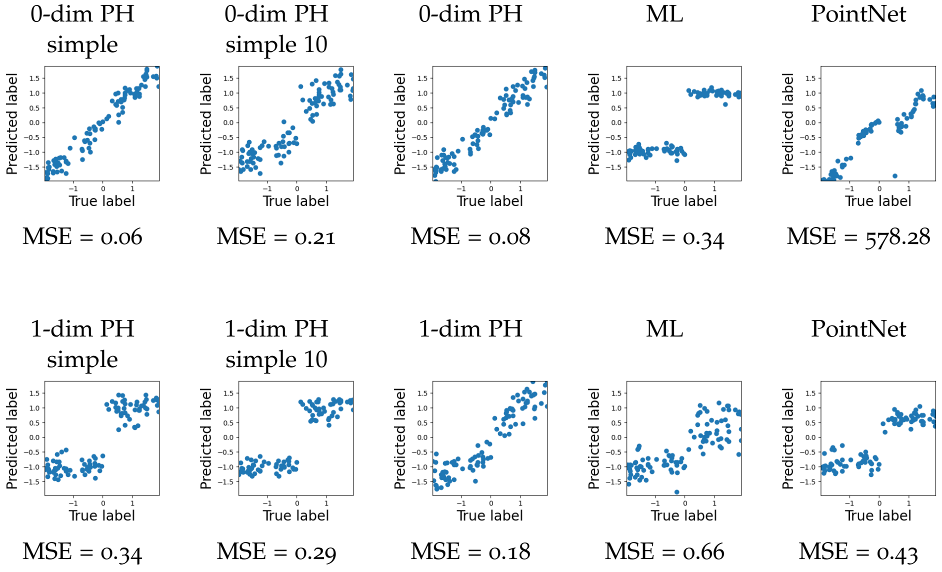


Figure 2.4.: Persistent homology can detect curvature.

2.4 Convexity

In this section, we consider the binary classification task that consists of detecting whether a point cloud is sampled from a convex set. Convexity is a fundamental concept in geometry [87], which plays an important role in learning, optimization

[5] 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.

[30], numerical analysis, statistics, information theory, and economics [301]. Furthermore, points of convexities and concavities have been demonstrated as crucial for human perception of shapes across many experiments [310].

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 PH's effectiveness might seem 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 a 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 2.2), 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 2.3). However, as we show in Theorem 2.2 (proof in Appendix A.1.1) and as our experiments suggest, it is a single, and the second-longest persistence interval that enables us to detect concavity. A crucial ingredient in our result is the introduction of tubular filtrations (Definition 2.1), which, to the best of our knowledge, are a novel contribution to the TDA literature (details in Appendix A.1.1).

Definition 2.1 (Tubular filtration)**.** Given a line $\alpha \subset \mathbb{R}^d$, we define the **tubular function with respect to** α as follows:

$$\begin{aligned} \tau_\alpha \colon \mathbb{R}^d &\to \mathbb{R} \\ x &\mapsto d(x,\alpha)\,, \end{aligned}$$

where $d(x,\alpha)$ is the distance of the point x from the line α. Given $X \subset \mathbb{R}^d$ and a line α, we are interested in studying the sublevel sets of τ_α, i.e., the subsets of X consisting of points within a specific distance from the line. We define

$$X_{\tau_\alpha,r} = \{x \in X \mid \tau_\alpha(x) \leq r\} \ = \{x \in X \mid d(x,\alpha) \leq r\}\,.$$

We call $\{X_{\tau_\alpha,r}\}_{r\in\mathbb{R}_{\geq 0}}$ the **tubular filtration with respect to** α.

Theorem 2.2 (Convexity detection with PH)**.** *Let $X \subset \mathbb{R}^d$ be triangulable[6]. We have that X is convex if and only if for every line α in $\mathbb{R}^d$ the persistence diagram in degree 0 with respect to the tubular filtration $\{X_{\tau_\alpha,r}\}_{r\in\mathbb{R}_{\geq 0}}$ contains exactly one interval.*

DATA We construct a balanced data set 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

[6] Recall, triangulable spaces are homeomorphic to a simplicial complex.

algorithm is actually detecting convexity, or only the different basic shapes. A few examples are shown in Figure 2.5. To construct a random convex shape, we generate 10 points at random, and then build their convex hull using the quickhull algorithm [350]. We construct random concave shapes in a similar way, but instead of the convex hull, we build the alpha shape [113, 115] with the optimized alpha 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.

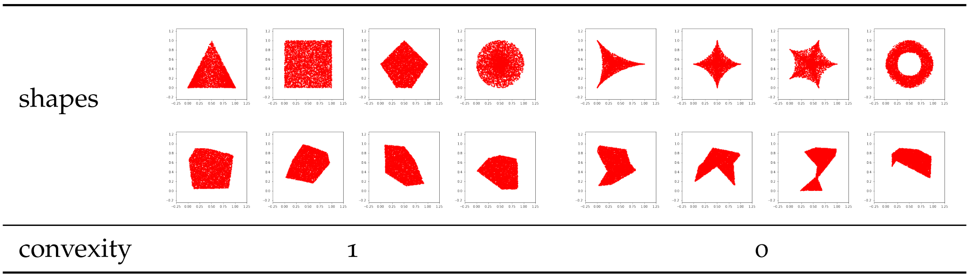


Figure 2.5.: Convexity data set.

ph pipeline To build a filtration (Section 1.3), we consider cubical complexes filtered by tubular functions that measure the distance of points from a certain line (Section 1.3.3, Definition 2.1). For a good choice of line, *multiple* components would be seen in the filtration of a point cloud sampled from a concave shape, at least for some values $r \in \mathbb{R}$ (see also the illustrations in Appendix A.1.1). For this reason we consider the cubical complex, rather than the standard Vietoris-Rips simplicial complex (Section 1.2.1) wherein these separate components could be connected with an edge (for details, see Appendix A.5.1). To build an image from the point cloud, we construct a 20×20 grid and 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 lines for the tubular filtration function (for a visualization of the pipeline, see Appendix A.5.1). For each of the nine lines, we extract 0-dimensional PD (Section 1.4.1), as it captures information about the components. If the point cloud is thus sampled from a convex shape, its PD will only see a single component for any line, whereas there will be multiple components at least for some lines for point clouds sampled from concave shapes. For this reason, for each of the nine lines, 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 components *for at least one line*. This scalar could even be used as some measure of the level of concavity of a shape (as in Appendix A.6.2.3, where PH detects a continuous measure of concavity on a real-world data).

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 2.6, with detailed results across multiple runs in Appendix A.5.2). We start with the easiest case, where both the train and test data consist of simple regular convex and concave shapes (Figure 2.5, first row), and then proceed to the scenario where both train and test data are random shapes (Figure 2.5, 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 data set of regular shapes which performs on par. Results reported in Appendix A.5.3 show PH is also computationally efficient.

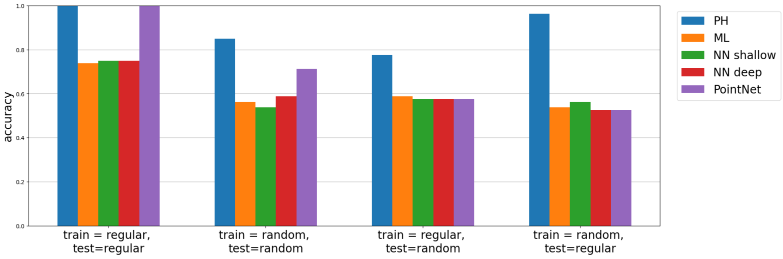


Figure 2.6.: Persistent homology can detect convexity.

The PH pipeline above makes a wrong prediction when concavity is barely pronounced, or if it is missed by the selected tubular filtration lines (for details, see Appendix A.5.4). However, the accuracy of PH can easily be improved simply by considering a finer resolution for the cubical complexes and/or additional tubular filtration lines. The particular PH pipeline summarized in this section would also make a wrong prediction if the data set would include shapes that have small or non-central holes, e.g., a square with a hole in the top left corner. In this case, the accuracy could also be improved by considering a finer cubical complex resolution and by considering additional non-central tubular filtration lines within shapes, or by adding (the maximum lifespan of the) 1-dimensional PH which captures holes. The pipeline is not limited to polygons, or connected shapes, and it can be generalized to surfaces in higher dimensions (Theorem 2.2). In Appendix A.7, we also consider the real-world data set FLAVIA for which we demonstrate that a PH pipeline is effective in detecting a continuous measure of convexity.

2.5 Long and short persistence intervals, topology and 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” [77]. 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 [29, 328]. 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 A.6.

topology and long persistence Stability results (Section 1.7) guarantee that a number of longest persistence intervals reflect the topological signal, i.e., the number of cycles [70]. These theorems give information about the threshold that differentiates between long and short persistence intervals. In Section 2.2, 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 2.2, but the second-longest interval is topological noise in Section 2.4, since every shape in the data set has only a single component (although this second-longest interval captures important *geometric* information, what enabled us to discriminate between convex and concave shapes). 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 2.3 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 2.2 and 2.3) or their position (with respect to the height or tubular filtration, as in Section 2.4). 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 2.4, where we show that a single interval detects convexity.

2.6 Conclusions

MAIN CONTRIBUTION The goal of this chapter is to gain a better understanding of the topological and geometric features that can be captured with persistent homology. We focus on the detection of the number of holes (Section 2.2), curvature (Section 2.3), and convexity (Section 2.4). Theoretical evidence for the first two classes of problems has been established in the literature, and we prove a new result that guarantees that PH can detect convexity (Theorem 2.2). We also experimentally demonstrate that PH can solve all three problems for synthetic point clouds in $\mathbb{R}^2$ and $\mathbb{R}^3$, 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. For convexity detection, we also show the effectiveness of PH in a real-world plant morphology application.

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 A.6. Due to the crucial role of shape classification in understanding and recognizing physical structures and objects, image processing and computer vision [211], our results demonstrate that PH can—to borrow the words from Wigner [358]—"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 2.5). 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 three selected problems and data sets, 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 the supplementary material (Appendix A.6) 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. Alternative approaches for the detection of convexity with PH (relying on higher homological dimensions, or multiparameter persistence) are particularly interesting avenues for further work. 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 [43, 156, 288, 368], an additional network layer for topological signatures, or PH-based priors, regularization or loss functions [43, 72, 75, 76, 158, 360, 379].

3

THE WHY:
Noise robustness of persistent homology, across filtrations and signatures

Often cited as the most favorable property of persistent homology (PH) and the main reason for practical success are the stability theorems that give theoretical results about noise robustness, since real data is typically contaminated with noise or measurement errors. However, little attention has been paid to what these stability theorems mean in practice. To gain some insight into this question, we evaluate the noise robustness of PH on the MNIST dataset of greyscale images. More precisely, we investigate to what extent PH changes under typical forms of image noise, and quantify the loss of performance in classifying the MNIST handwritten digits when noise is added to the data. The results show that the sensitivity to noise of PH is influenced by the choice of filtrations and persistence signatures (respectively the input and output of PH), and in particular, that PH features are often not robust to noise in a classification task.

This chapter is based on the following publication:

> Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré, *Noise robustness of persistent homology on greyscale images, across filtrations and signatures*, PLOS ONE 16, No. 9 (2021): e0257215.

THE WHY: NOISE ROBUSTNESS OF PERSISTENT HOMOLOGY, ACROSS FILTRATIONS AND SIGNATURES

3.1 Introduction

The recent popularity of persistent homology (PH) (Chapter 1) in data analysis is primarily attributed to its proven stability (Section 1.7): PH is robust under small perturbations in the input, which is of crucial importance for practical applications due to the unavoidable presence of noise or measurement error in real data [2]. Moreover, PH is commonly assumed to be a topological invariant and therefore robust under affine transformations.

However, it is often overlooked in the literature how strongly the stability theorems are influenced by the choice of a:

- filtration, the input for PH, or the medium through which the homology information is extracted from data (Section 1.3). Indeed, it is important to remember that PH is not directly calculated on the data (e.g., an image, or a point cloud), but on the filtration that approximates the shape of data at different scales (see Figure 3.3, Figure 3.4). The filtration must satisfy the underlying assumptions in the stability theorem, which then ensures robustness under minor perturbations in the input - filtration, not necessarily under minor perturbations of the data. Moreover, the level of robustness is directly determined by the filtration.

- persistence signature, the output of PH, or the medium used to represent PH (Section 1.4). Indeed, the stability theorems do not provide a guarantee of the noise robustness of PH in general, but rather prove the stability of a selected signature (with the corresponding metric).

In addition, the choice of filtration influences the type of information captured with PH: for some filtrations, PH can reveal geometric information and thus not be invariant e.g., under rotation or translation. Furthermore, even if the stability theorem holds for the given filtration and signature, little attention has been paid to what these stability theorems mean in practice. In particular, it is unclear if the stability results imply the noise robustness of PH features in a classification task.

To investigate these issues, we carry out computational experiments that evaluate the noise robustness of PH on the MNIST dataset of greyscale images, under different types of noise. More precisely, the main objective of this work is to address the following research questions, across different filtrations and persistence signatures:

(RQ1) How much does PH change under noise in the data?

(RQ2) How discriminative does PH remain if the data contains noise?

The findings of this analysis can therefore help to guide the choice of appropriate filtrations and signatures, especially in the presence of noise in the data. To the best of our knowledge, this issue has not been studied in literature so far. In the majority of studies that apply PH to tackle a particular problem (and are thus not concerned with noise robustness in particular), a single filtration and signature are commonly adopted, without a discussion on the motivation, assumptions, and implications behind the specific choice. There are a few noteworthy examples in the literature, such as [129], which do consider multiple filtrations and/or signatures (on the MNIST data set), but they focus on the discriminative power, rather than the noise robustness of PH features. The authors do conclude, however, that PH is reputed for its robustness to noise, and suggest conducting a similar study under different types of image noise [129].

The next sections introduce our choice filtrations and persistence signatures (Section 3.2), and detail how this choice influences the stability theorems (Section 3.3). We then proceed to evaluate the robustness of PH on the MNIST image data set of handwritten digits (Section 3.4), concluding with a summary of findings and limitations of this work, that motivate some suggestions for future research (Section **??**).

3.2 Choice of filtrations and signatures

This section discusses the choice of filtrations and persistence signatures used in this chapter, and specifices the choice of their parameters. We consider the **binary-, greyscale-, density-, and radial-filtered cubical complexes, and the Rips and DTM-filtered simplicial complexes** constructed from an image $Z = [z_{uv}]$ (Figure 3.1).

BINARY FILTRATION The binary filtration function considers binary values of pixels by introducing a greyscale threshold z_0:

$$\phi_{z_0}(u, v) = \begin{cases} 0 & z_{uv} \geq z_0 \\ 1 & \text{otherwise.} \end{cases} \tag{3.1}$$

PH with respect to this filtration function corresponds to the homology of the image [129], meaning that it only determines the *number* of cycles (Betti numbers, see Section 1.1). It is of crucial importance that the greyscale threshold parameter z_0 is sufficiently low, so that all dark pixels are part of the filtration immediately at scale $r = 0$. Indeed, if only a single pixel along some hole has a greyscale value below the given threshold z_0, this pixel will only be a part of the filtration at resolution $r = 1$, as any other pixel in the image, so that the hole is never seen at any scale in the filtration.

GREYSCALE FILTRATION In order to study how cycles persist with respect to the greyscale value, a nonbinary filtration function is a more natural choice. In the greyscale filtration function, one relates each pixel to its greyscale value:

$$\phi_{\mathrm{grsc}}(u,v) = \max(Z) - z_{uv}.$$

An advantage of the greyscale compared to other considered filtrations is that it is parameter-free. In particular, it does not require an a-priori defined greyscale threshold. Next to the number of cycles, PH with respect to the greyscale filtration function thus also captures information about the brightness of the cycles.

DENSITY FILTRATION If the greyscale value of a single pixel changes significantly (e.g., from black to white), an existing hole in an image might get disconnected, or an additional single-pixel hole might appear. To avoid such sensitivity to outlying greyscale values, we can rather consider the density filtration function. Thereby, we relate each pixel to the number of "dark-enough" pixels in its neighborhood. More precisely, let the neighbourhood $N((u,v), d_0, z_0)$ be the set of all pixels (u',v') with $z_{u'v'} \geq z_0$ (for a given threshold z_0), that are within given distance from pixel (u,v) :

$$\|(u',v') - (u,v)\|_2 \leq d_0.$$

Density filtration function is then defined as:

$$\phi_{d_0,z_0}(u,v) = N(d_0) - |N((u,v), d_0, z_0)|,$$

where $N(d_0)$ is the total number of pixels within distance d_0, for any (u,v). The threshold parameter z_0 is not of crucial importance. For instance, if only one pixel along a hole is very bright, the hole will never be seen in the binary filtration, but it will persist from early on in the density filtration, for most of the values of z_0. A good choice for the size of the neighbourhood d_0 obviously depends on the size of the image. For the data set of 28 × 28 MNIST images, we take $d_0 = 1$.

RADIAL FILTRATION While the greyscale and density filtration capture information about the brightness of cycles, it is possible to capture other information. For example, the *position* of cycles is captured with PH if one considers the radial filtration function defined as the distance from a given reference pixel (u_0, v_0) :

$$\phi_{(u_0,v_0),z_0}(u,v) = \begin{cases} \|(u,v) - (u_0,v_0)\|_2 & z_{uv} \geq z_0 \\ \max_{(u',v')} \|(u',v') - (u_0,v_0)\|_2, & \text{otherwise} \end{cases}$$

Similar to the binary filtration function, the greyscale threshold z_0 is crucial for the radial filtration as well, whereas the density and Rips filtration are less sensitive to this parameter (point cloud points corresponding to non-neighboring pixels can still be connected with an edge, for a sufficiently large resolution r). However, to be consistent, we take the same threshold value $z_0 = 0.5\max(Z)$ for the Rips, DTM, binary, density and radial filtrations.

The choice of the reference pixel (u_0, v_0) depends on where the important topological features are expected to be located in an image, and how this location differs across classes of data. For instance, if we consider (u_0, v_0) to be a pixel in the center of the image, the holes in digits 6 and 9 would be seen at the same resolution r in the filtration. Since we aim to differentiate between digits 6 and 9, we will consider $(u_0, v_0) = (0, 0)$.

RIPS AND DTM FILTRATION As discussed in Section 1.2 (see in particular Figure 1.8), it is also possible to construct a point cloud from a given image by replacing every dark enough pixel with a point cloud point, and study the Vietoris-Rips (Section 1.3.1) or Distance-to-Measure (DTM) filtration (Section 1.3.2).

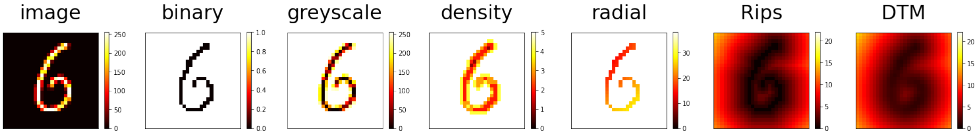


Figure 3.1.: Filtration functions on an image. The first plot shows an example MNIST image Z, with greyscale pixel values in $[0, 250]$. The next four plots respectively show the heatmap for the binary, greyscale, density and radial filtration function $\phi : K(Z) \to \mathbb{R}$, where $K(Z)$ is the cubical complex corresponding to the given example image. The final two plots visualize the heatmap of $\phi : K(Z) \to \mathbb{R}$, where ϕ is the discretized version of the Rips and DTM filtration functions $d_{X(Z,z_0)} : \mathbb{R}^2 \to \mathbb{R}$ and $d_{X(Z,z_0),m} : \mathbb{R}^2 \to \mathbb{R}$, and $X(Z, z_0)$ is the point cloud obtained from image Z and threshold greyscale value z_0.

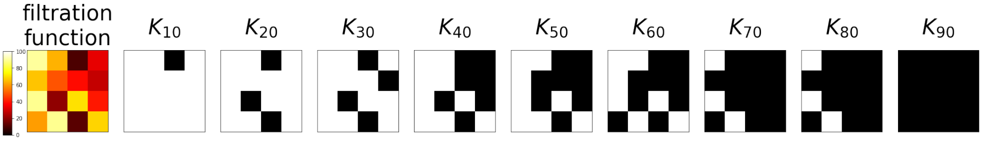


Figure 3.2.: Filtration on a cubical complex. The first image represents the values $[0, 100]$ of the filtration function $\phi : K(Z) \to \mathbb{R}$. The next nine plots show the cubical complexes $K_{10} \subseteq K_{20} \subseteq K_{30} \subseteq \cdots \subseteq K_{90}$, where K_r corresponds to the union of all cubes, i.e., pixels (u, v) with the filtration value $\phi(u, v) \leq r$. There is only one 1-dimensional cycle, i.e., hole, which is first seen in K_{40}, and then disappears or closes in K_{70}.

Persistent homology (PH) information in dimension k captures the values of resolution r when each k-dimensional cycle is born and when it dies in a filtration, denoted with b and d. The cardinality of this multi-set of persistence intervals (b_i, d_i) $(i \in \mathbb{N}^+)$ counts the number of k-dimensional cycles (although many or even a majority might only show up in the filtration for a brief while, i.e., for a small range of resolution values r, yielding very short lifespans or persistence $l_i = d_i - b_i$, and as will be explained later in the chapter, can thus be considered as irrelevant). However, the choice of the filtration defines the interpretation of the birth values and death values, which can reflect additional topological or geometric information, such as their size or position (Table 3.1).

Consider some examples in Table 3.1. The table lists 1-dimensional PH calculated for a few example MNIST images (or an image with an outlying pixel), across selected filtrations. The notation $(b, d)^*$ implies that multiple cycles appear and disappear at the same time (thus, PH is a *multi*-set, where each element has its multiplicity). The notation $(b, d)^{**}$ implies that there are multiple intervals with a similar birth and death value. As already noted, the cardinality of the set of persistence intervals determines the number of cycles. However, the definition of filtration implies the interpretation of birth and death times, so that PH with different filtrations captures different topological (and geometric) information, what further influences its noise robustness and discriminating power. For example, an additional point at an outlying distance from a point cloud can have an important influence on PH with the Rips filtration (e.g., an additional black pixel within a hole will change the persistence of that hole, see persistence intervals in red), but this is less true for the DTM filtration, as the outlier will have a large distance from the nearest point cloud neighbours and will thus appear only very late in the filtration. A reverse example is a pixel with an outlying greyscale value (e.g., a white pixel in a dark region) which has an important influence on PH with the binary, greyscale and radial filtration (in blue), but much less for the density, Rips and DTM filtration. If geometric information is captured, PH becomes sensitive under some affine transformations. Furthermore, 1-dimensional PH with binary, greyscale and density filtration cannot differentiate between digits 0, 6 and 9 (as they all have one hole of similar brightness), but radial filtration allows to discriminate between digits 6 and 9 (as the holes have a different position), and Rips and DTM filtrations enable to distinguish between 0 and 6 (as the holes are of different size).

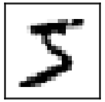
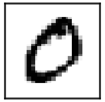
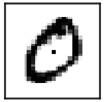
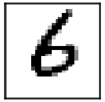
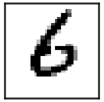
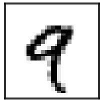
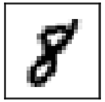
filtration	1-dimensional PH							topological or geometric information about a hole
								
binary	∅	(0, 1)	(0, 1)	(0, 1)	∅	(0, 1)	(0, 1) (0, 1)	-
greyscale	∅	(17, 255) (159, 171)	(17, 255) (159, 171)	(0, 255)	∅	(14, 255) (213, 231)	(9, 255) (19, 255) (65, 86) (223, 242) (234, 255)	greyscale along and within hole
density	∅	(1, 5)	(1, 5)	(2, 5)	(3, 5)	(1, 5) (3, 4) (4, 5)	(1, 5) (2, 5) (3, 4)	density along and within hole
radial	∅	(25.00, 38.18)	(25.00, 38.18)	(24.76, 38.18)	∅	(20.00, 38.18)	(20.12, 38.18) (24.60, 38.18)	position
Rips	(1.00, 1.41)*	(1.00, 1.41)* (1.00, 8.94) (2.00, 2.24)	(1.00, 1.41)* (1.00, 6.40) (2.00, 2.24) (5.00, 6.00)	(1.00, 1.41)* (1.00, 5.00)	(1.00, 1.41)* (2.00, 5.00)	(1.00, 1.41)* (1.00, 4.12) (2.83, 3.00)	(1.00, 1.41)* (1.00, 3.00) (1.00, 3.60) (2.83, 3.00)	sparsity, distance across hole
DTM	(4.13, 4.36)**	(4.40, 4.60)** (4.94, 12.81)	(4.08, 4.38)** (11.51, 11.95) (4.94, 12.32)	(4.08, 4.33)** (5.49, 8.60)	(4.39, 4.54)** (6.10, 8.74)	(4.16, 4.58)** (6.17, 6.19) (4.16, 7.43)	(4.02, 4.33)** (6.43, 6.51)** (4.34, 6.57) (5.00, 7.43)	sparsity, size

Table 3.1.: Persistent homology across filtrations and noise.

Obviously, the choice of filtration has an important influence on the noise robustness and discriminative power of PH. If PH only registers the *number* of holes, it is of topological nature and is invariant under rotations, translations, or stretching (in topology, a coffee mug is equivalent to a donut, see Figure 1.2), which can be useful in some applications, such as recognition of animals, cars, or people in images. If PH also captures the position of holes, it is sensitive to rotation, but able to differentiate, e.g., between digits 6 and 9. If the size of the holes is also captured, the PH information is not robust to rescaling, but it enables us to differentiate between a 6 and a 0.

In order to evaluate the noise robustness of PH, we are interested in computing the distance between PH information of two images. These two images will, for example, be the non-noisy and noisy version of an image. In the computational experiments, the distance between PH is calculated using the **Wasserstein W_2 distance between persistence diagrams (PDs), or l_2 distance between persistence landscapes (PLs) or persistence images (PIs)** (Section 1.4, Section 1.6). The parameter p in both W_p and l_p determines the importance of long compared to short distances.

Furthermore, for a chosen p, the choice of persistence signature also influences the importance of cycle lifespans. Indeed, it is easy to see that the Wasserstein W_p^p and l_p^p distances between PDs and PLs or PIs corresponding to $PH_1 = \{(b,d)\}$ and $PH_2 = \emptyset$ reflect $(d-b)^p$, $(d-b)^{p+1}$ and $\alpha^p(b, d-b)$, where α is the weight function for PIs (see Section 1.4.2). Since we consider $\alpha(b, d-b) = (d-b)^2$ as the weight function, this means that the cycles that persist for a short time matter the least for PIs, and the most for PDs (Table 3.2).

Persistence signature	**Limiting behaviour of $\delta^2(PH, \emptyset)$**
PD	$O(l^2)$
PL	$O(l^3)$
PI, with weight function $\beta(b,l) = l^2$	$O(l^4)$

Table 3.2.: Persistent homology across signatures. The choice of persistence signature, and the corresponding metric, determines how sensitive PH is to cycles (b,d) with short persistence, or lifespan, $l = d - b$. The table lists the limiting behavior, or growth rate, of the function $\delta^2(PH, \emptyset) = \delta^2(\{(b,d)\}, \emptyset) = f(d-b) = f(l)$, where distance δ represents the Wasserstein W_2 distance between persistence diagrams, or l_2 distance between persistence landscapes or persistence images. The growth rate reflects the importance of a cycle with lifespan l, which influences the noise robustness and discriminative power of PH.

The choice of persistence signature, and the corresponding metric, therefore has an important influence on the noise robustness and discriminative power of PH, although, surprisingly, little research has been carried out in this area before [122].

Recently, [122] evaluated the overlap between the l_p distances between persistence landscapes and persistence images, and the Wasserstein W_p distances between persistence diagrams, on three different data sets (including MNIST images). The results clearly show that the distances between vectorized persistence summaries greatly differ from the distances between PDs. Another recent and detailed investigation of the distance correlation between different persistence signatures can be found in [340]: the authors conclude that the considered signatures are "same but different", as they commonly contain the same information, but are shown to yield different results from statistical analyses since they lie in different metric spaces. In addition, the classification accuracy is shown to vary greatly when distances between shapes are given by the distances between their PDs, PLs or PIs in [2, Table 1].

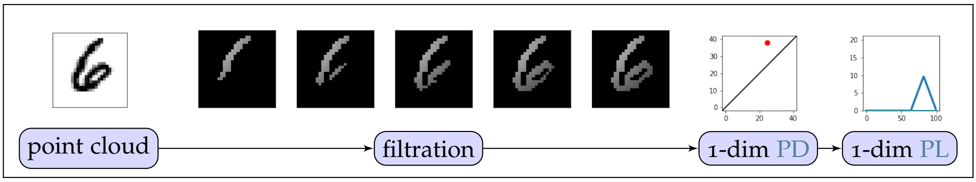


Figure 3.3.: The image structure at scale r can be approximated with a cubical complex, e.g., we can look only at pixels within distance r from the top left pixel. 1-dimensional PH consists of one persistence interval that reflects the one loop in the image, that is represented with the single PD point, or a single triangle in PL.

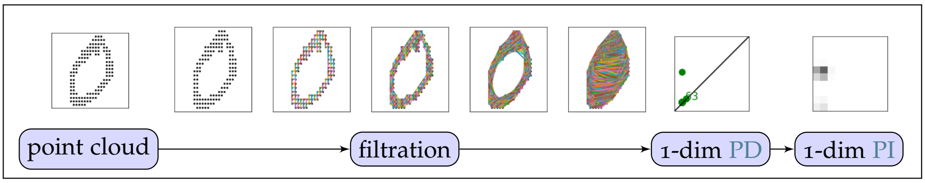


Figure 3.4.: To approximate the structure of an image at scale r, we can consider it as a point cloud, and approximate its structure by constructing an edge between two points whenever they are within distance r. 1-dimensional PH consists of one persistence interval that reflects the one loop in the image, that is represented with the single PD point, or a single dark region in PI.

In our experiments, we consider the PI grid of size 10×10, choose the weight function $\alpha(b,l) = l^2$, and set the Gaussian function variance to 5% of the maximum death value in PDs for the given filtration function and homological dimension. The number of main connected components or holes in the MNIST data set is typically 0, 1 or 2. However, additional cycles might appear in noisy images, and we thus consider the first 10 landscapes λ_j $(j \in \{1,2,\ldots,10\})$ (Section 1.4.3); although we immediately note that this means that PLs and PDs do not necessarily capture the same information (e.g., if there are more than 10 cycles in this case). Obviously, this number should be higher if we expect a large number of important cycles that discriminate between classes of data. We set the landscape resolution $P = 100$.

3.3 Stability theorems across filtrations and signatures

Stability theorems (Section 1.7) may be viewed as a precise statement about robustness to noise: stable representations of PH are not sensitive to noise in the input. More precisely, recall that for persistence diagrams calculated with respect to filtration functions ϕ and ψ, a stability theorem ensures that there exists a constant $c \in \mathbb{R}$ such that:

$$W_p(PD(\phi), PD(\psi)) \leq c\|\phi - \psi\|_p.$$

A stability theorem for other persistence signatures PH states the following:

$$\|PH(\phi) - PH(\psi)\|_p \leq cW_p(PD(\phi), PD(\psi)).$$

In this section, we discuss the importance of the choice of filtration, signature, and data set in the interpretation of stability theorems, that is often overlooked in the literature. This discussion then motivates our computational experiments in the next section.

Stability theorems and the choice of filtration

The choice of filtration plays a crucial role in the existence and practical value of stability theorems. First of all, in order for the stability theorem to hold for a particular filtration, the filtration function must satisfy the underlying assumptions.

Second, note that the stability theorems ensure that PH is robust under minor perturbations of its input - filtration, and not under minor perturbations in the data space itself. Small changes in the space do not always imply small changes in the filtration function, so that stability theorems provide no guarantee of robustness in such a scenario. For instance, if Z' is obtained by changing the image Z only slightly, $\|\delta_{X(Z,z_0)} - \delta_{X(Z',z'_0)}\|_p$ can be large (and it corresponds to the Gromov-Hausdorff distance between point clouds $X(Z,z_0)$ and $X(Z',z'_0)$, for $p = \infty$ [69]). Although PDs are theoretically stable with respect to the Rips filtration (with the distance function $\delta_{X(Z,z_0)} : \mathbb{R}^n \to \mathbb{R}$ as its filtration function), the upper bound for $W_p(PD(\delta_{X(Z,z_0)}), PD(\delta_{X(Z',z'_0)}))$ is so large that it makes little sense in practice: these PDs are sensitive to outliers.

Finally, stability theorems are worst-case results, as they do not necessarily ensure tightness of the upper bound provided for the distance between PH information. This is true even if small perturbations in the data result only in small perturbations of the filtration. Let us consider an image Z, and another image Z' obtained with some transformation $\pi : Z \to Z'$. If we apply the stability theorem to the space Z and filtration functions $\phi_{\text{grsc}} : K(Z) \to \mathbb{R}$ and $\psi_{\text{grsc}} = \phi_{\text{grsc}} \circ \pi : K(Z) \to \mathbb{R}$, the right-hand side in the stability theorem $\|\phi_{\text{grsc}} - \psi_{\text{grsc}}\|_p$ (and this change in the greyscale values precisely corresponds to the change in the image) is an upper bound for the change in PDs.

If Z is the MNIST image of digit 6, and Z' the same image but with one pixel changed from black to white (Table 3.3), then $\|\phi_{\text{grsc}} - \psi_{\text{grsc}}\|_p = 255$ is sufficiently large to allow to change $PD(\phi_{\text{grsc}})$ with one hole to $PD(\psi_{\text{grsc}})$ with no holes. However, if Z is the MNIST image of a digit 0, and Z' the same image but with one pixel changed from white to black (Table 3.1), we again have $\|\phi_{\text{grsc}} - \psi_{\text{grsc}}\|_p = 255$ but PD remains unchanged. As another example, we can consider Z' to be the translated image Z, when $\|\phi - \psi\|_p$ is large for both the greyscale or radial filtration function. However, $W_p(PD(\phi), PD(\psi))$ is zero when ϕ is greyscale (as PDs then only register the number and brightness of cycles), but it is large for the radial filtration function (which also captures the position of cycles).

Stability theorems and the choice of persistence signature

It is clear from the formulation of stability theorems in Section 1.7 that these results only hold for some signatures, and some metrics. As we discuss there, PLs and PIs are shown not to be stable with respect to the l_2 metric in [320], although this is the standard choice in applications, when it is commonly assumed that these are stable representations. This is one of the reasons why [320] recently emphasized that "the stability theorems are one of the most misunderstood and miscited results within the field of topological data analysis". It is, however, interesting to see if the stability holds in practice, and to which degree.

Stability theorems and the type of data set

If the stability theorem holds for a chosen filtration and persistence signature, it does not imply the noise robustness of PH features in a classification task - this depends on the application domain, i.e., the type of data set.

Let us go back to the example of Z being the MNIST image of digit 6, and Z' being the same image but with one pixel changed from black to white (Table 3.1). As already indicated, the upper bound for the greyscale filtration is large enough to allow to change $PD(\phi_{\text{grsc}})$ with one hole to $PD(\psi_{\text{grsc}})$ with no holes. This is problematic for the classification of the MNIST data set using PDs, since any image contains none, one or two holes, but it would pose less of an issue if there is a greater variety in the number of holes across data classes.

3.4 Results and discussion

We start this section by describing the data set of greyscale images, and the different types of noise considered in our experiments. In the next subsection, we investigate how sensitive the persistent homology information is to these types of noise, by evaluating the distance between PH for noisy and non-noisy images. This information, however, only paints a part of the picture, since in practical use cases, the PH information must also vary sufficiently among data points in order to form discriminative features in e.g., classification tasks. In the final subsection, we thus investigate the noise robustness of persistent homology together with its discriminative power, by evaluating the drop in classification accuracy when the test data consists of noisy, compared to non-noisy images.

3.4.1 *(Noisy) data sets*

We consider the MNIST data set [201], as it is a well-defined benchmark of greyscale images, and the shape of each of the digits is well understood. To reduce the computation time, we restrict the study to the first 1000 images in the data set. We investigate three types of affine transformations, changes in image brightness and contrast, and three types of pure noisy transformations, each at two different levels, and in different directions, if applicable (Table 3.3).[1]

For every (non-noisy and noisy) data set, i.e., for each image in each of the data sets, we calculate the values of filtration functions on each pixel, and the 0- and 1-dimensional persistent homology[2] information with respect to all considered filtrations and persistence signatures (with the specified values of the parameters) using the python GUDHI library [331].

3.4.2 *Noise robustness*

The goal of this section is to understand in what way, and to which degree, is the persistent homology information sensitive to noise, across different filtrations and persistence signatures. In order to address this question, we start by visualizing the different filtration functions and persistent homology information for an *example* MNIST image, under different data transformations (Figure 3.5 and Figure 3.6). We can conclude the following.

affine transformations (rotation, translation, stretch-shear-reflect) PH on binary and greyscale filtration remains unchanged under any affine

[1] The greyscale pixel values are clipped to the interval [0, 255].

[2] For 0-dimensional homology, we truncate the death value of infinite intervals to the maximum finite death value for the given filtration function, across all transformations.

Transformation	**Definition of transformation**
rotation	Rotation by 45 degrees clockwise (rotation 45), or 90 degrees counterclockwise (rotation -90).
translation	Translation by 1 pixel right and down (translation 1 1), or 2 pixels left and up (translation -2 -2).
stretch, shear and reflect	Stretch, shear and flip respectively by a factor of 1.5 (i.e., an image is scaled down by factor of 1.5 in the x direction, whereas it remains unchanged in the y direction, so that the image is stretched), 10 degrees and horizontal (stretch-shear-flip 1.5 10 h), or by a factor 0.75, -20 degrees and vertical (stretch-shear-flip 0.75 -20 v).
brightness	-50 or 100 is added to the greyscale value of each pixel (brightness -50 and brightness 100, respectively).
contrast	Greyscale value of each pixel is multiplied with 2 or 0.5 (contrast 2 and contrast 0.5, respectively).
gaussian noise	Random noise drawn from normal distribution $\mathcal{N}(0, 10)$ or $\mathcal{N}(0, 20)$ is added to the greyscale value of each pixel (gaussian noise 10 and gaussian noise 20, respectively).
salt and pepper noise	5% or 10% of random pixels in an image are changed, with equal probability, to either white (i.e., salt) or black (i.e., pepper) (salt and pepper noise 5 and salt and pepper noise 10 respectively).
shot noise	Greyscale value of each pixel is replaced with a random number drawn from the Poisson distribution, with the distribution mean corresponding to the original greyscale pixel value, scaled down with a factor 50 (shot noise 50) or 100 (shot noise 100), as Poisson distribution is spread out more for lower means. Since the Poisson distribution with mean zero is equal to zero, the shot noise only changes non-white pixels.

Table 3.3.: Image noise.

transformations[3], since it only registers the number and brightness of cycles (it is a topological invariant). However, under stretch-shear-reflect the density along and within a hole changes, which results in a change of birth and death values for 1-dimensional cycles with respect to the density filtration. Radial filtration function

[3] In the computational experiments, the affine transformations sometimes slightly disturb the greyscale values, so that, e.g., some cycles can appear or disappear in an image (see, for instance, the additional one-pixel hole for the binary filtration under rotation 45 in Figure 3.6).

captures the position of the cycles, so that the birth and death values of cycles can also change significantly under any affine transformation. PH on Rips and DTM filtration is robust under rotation and translation. However, PH with Rips and DTM filtration captures the size of cycles, and is thus sensitive to affine transformations that rescale the image. For instance, under stretch-shear-reflect that enlarges a digit, the number of point cloud points increases, resulting in many additional short persisting 0-dimensional cycles for these filtrations. The death value of 1-dimensional cycles for Rips and DTM filtration also changes under stretch-shear-flip, as the PH in this case reflects the size of the hole.

brightness PH on binary and radial filtration does not see important changes if the brightness of an image is adjusted. However, a change in image brightness does result in changes of the birth or death values in 1-dimensional PH on greyscale or density filtration, and additional cycles can be captured with density filtration. A change of thickness of a digit also results in additional 0-dimensional cycles for Rips and DTM filtration, that are of short persistence, but there are many. For these filtrations, there is also a minor change in the death value for 1-dimensional cycles, as it captures the size of the hole that can change under a change in brightness.

contrast PH with respect to most of the considered filtrations is invariant under changes in the contrast of an image. The only exception is 1-dimensional PH with greyscale filtration, where the birth or death value of cycles can change.

salt and pepper noise Gaussian, salt and pepper, and shot noise change the greyscale value of some random pixels. For each black pixel on a white background in the salt and pepper noise, a new one-pixel connected component (a long persisting 0-dimensional cycle) appears for PH on binary, greyscale, and radial filtration. If a pixel in a neighborhood of black pixels is changed to white, an additional long persisting 1-dimensional cycle (one-pixel hole) can appear for PH on these filtrations. Also, an existing hole in the non-noisy image may become disconnected in the noisy image, and thus not registered. The additional 0-dimensional cycles are all born at birth value 0 for PH on Rips filtration, but they die earlier (as soon as they are connected to another point cloud point), so that Rips is more robust under this transformation, but still severely impacted by the outliers. One-pixel or disconnected holes are not an issue for PH on Rips filtration, but the death value of 1-dimensional cycles can decrease due to the additional pixels within a hole (see also the image of digit 0, and the same image with a single outlier in Table 3.1). PH on DTM filtration is significantly more robust to salt and pepper noise, as the outliers are “washed out”.

GAUSSIAN NOISE The Gaussian noise produces similar types of perturbations as the salt and pepper noise, but the change in the greyscale value is much less prominent, so that no additional cycles are typically seen with the binary or radial filtration (which take the binary image as input), and have a very low persistence for the greyscale filtration.

SHOT NOISE Shot noise only changes the non-white pixels (to lighter or darker), so that a digit might become disconnected into a few components, a hole might become disconnected, and many one-pixel holes may appear. The additional 0-dimensional cycles have a long lifespan for binary and radial filtration, but they are short for PH on greyscale filtration (or more precisely, they are directly related to the strength of the change of the greyscale pixel values) and density filtration. 1-dimensional PH with these filtrations exhibits similar behavior. As already mentioned, PH with Rips and DTM filtration is more robust under this type of noise, since disconnected components or holes can still be captured, as the Rips and DTM filtration connect non-neighboring pixels with a sufficiently large edge (resolution r in the filtration).

PDs, PLs and PIs reflect the same information about the cycles, and Figure 3.5 and Figure 3.6 show that they change accordingly. However, without considering the metric on these spaces of persistence signatures, we cannot derive any insights regarding the difference in the noise robustness from these figures.

Furthermore, the major part of the discussion above is based only on a single example data point. We therefore calculate the (l_2 or W_2) distance between each image in the data set, and its noisy variant, when images are represented with their filtration functions or persistent homology information (Table 3.4, Table 3.5). The results on the complete data set align well with the findings discussed above for an example image. In addition, the results clearly shows that, for any given filtration and homological dimension, there is a relative difference between PDs, PLs and PIs in robustness under various transformations. For instance, 0-dimensional PH on Rips filtration is more sensitive to salt and pepper than shot noise for any persistence signature, but this difference is much more pronounced for PLs, and in particular for PIs, compared to PDs.

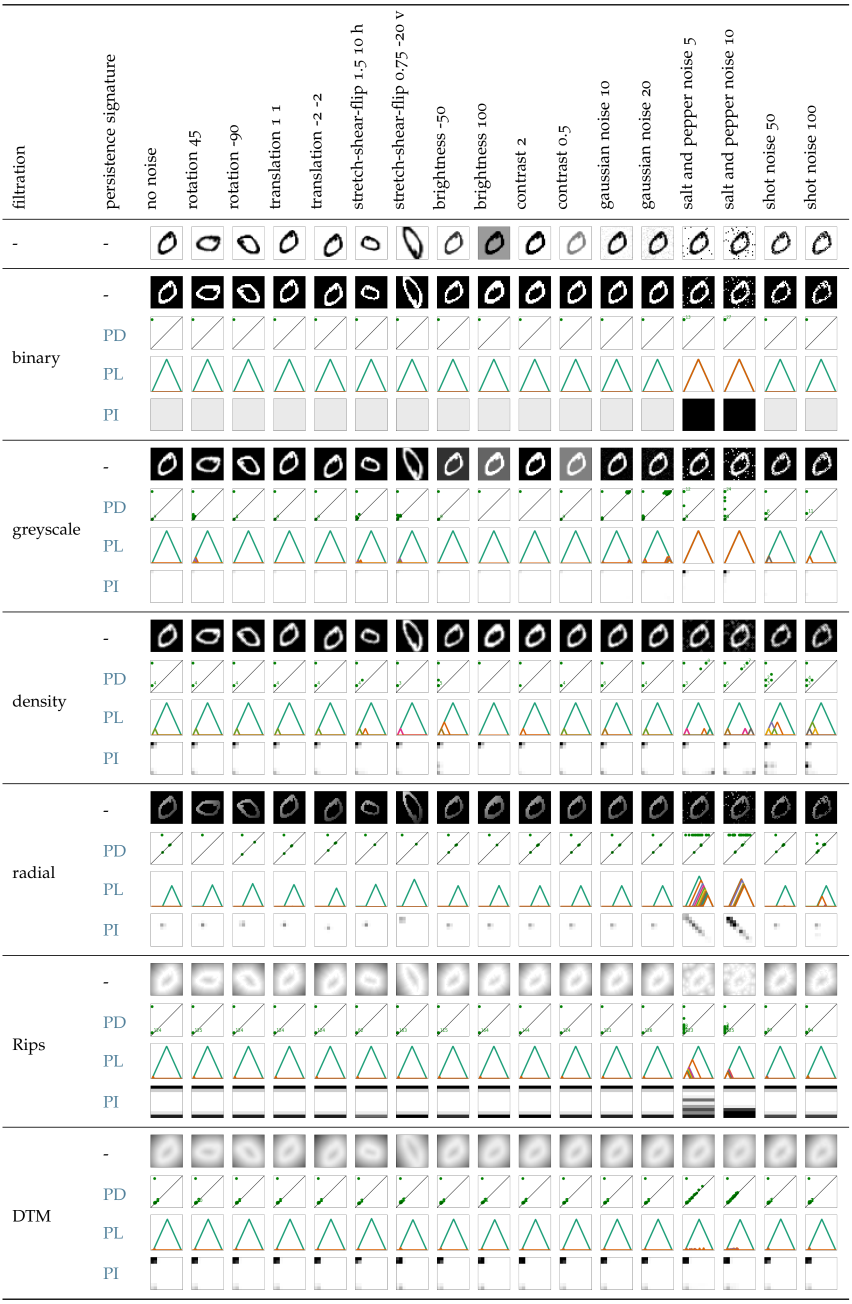


Figure 3.5.: Noise robustness of 0-dimensional PH on an example image. Illustration of the effect of various image transformations when the image is represented with its filtration function values (1st row of each filtration), or 0-dimensional PD (2nd row), PL (3rd row), or PI (4th row).

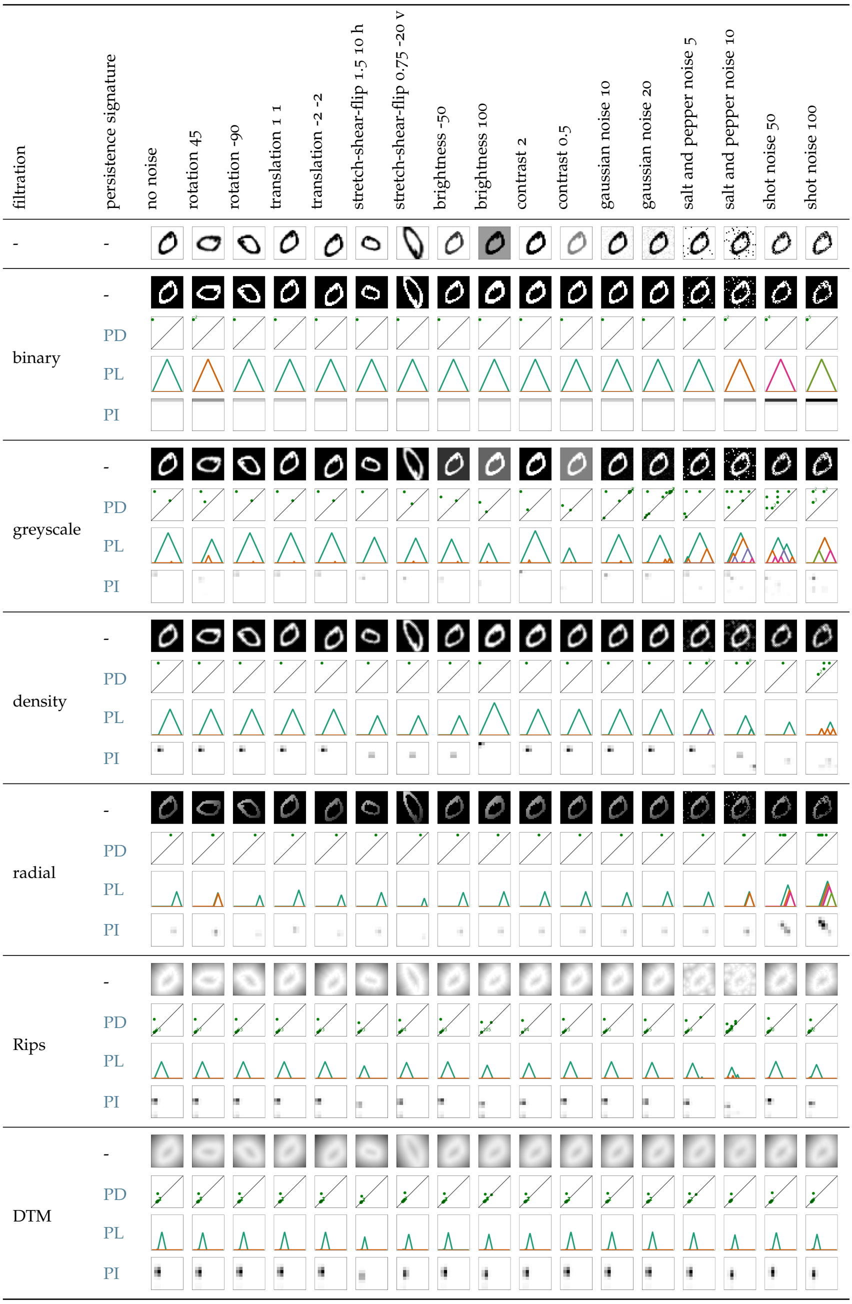


Figure 3.6.: Noise robustness of 1-dimensional PH on an example image. Illustration of the effect of various image transformations when the image is represented with its filtration function values (1st row of each filtration), or 1-dimensional PD (2nd row), PL (3rd row), or PI (4th row).

filtration	homological dimension	persistence signature	rotation 45	rotation -90	translation 1 1	translation -2 -2	stretch-shear-flip 1.5 10 h	stretch-shear-flip 0.75 -20 v	brightness -50	brightness 100	contrast 2	contrast 0.5	gaussian noise 10	gaussian noise 20	salt and pepper noise 5	salt and pepper noise 10	shot noise 50	shot noise 100
binary	-	-	11.4	11.95	9.12	11.9	10.59	12.28	2.73	5.63	4.31	**0.00**	1.45	2.11	4.42	6.24	5.0	6.34
	0	*PD*	0.01	**0.00**	**0.00**	**0.00**	0.02	0.01	0.01	0.02	0.01	**0.00**	**0.00**	**0.00**	1.80	2.43	0.17	0.64
		PL	0.06	**0.00**	**0.00**	0.01	0.16	0.10	0.09	0.18	0.12	**0.00**	0.04	0.03	12.03	12.17	1.42	5.16
		PI	10.82	**0.00**	**0.00**	1.91	27.37	15.92	17.19	30.56	21.01	**0.00**	5.73	5.73	8430.76	15135.00	282.02	1395.47
	1	*PD*	0.04	**0.00**	**0.00**	**0.00**	0.05	0.05	0.03	0.11	0.08	**0.00**	0.01	0.02	0.32	0.52	0.67	0.66
		PL	0.33	**0.00**	**0.00**	**0.00**	0.37	0.40	0.27	0.91	0.64	**0.00**	0.10	0.17	2.61	4.23	5.45	5.33
		PI	18.59	**0.00**	**0.00**	**0.00**	20.20	20.20	14.14	51.72	35.36	**0.00**	5.05	8.89	170.32	333.36	503.68	494.59
grsc	-	-	2454.9	2707.85	1906.79	2679.12	2335.78	2656.99	1268.72	2646.74	653.29	3295.03	207.45	412.55	1104.09	1560.62	814.52	1092.34
	0	*PD*	22.23	0.04	**0.00**	0.34	26.36	24.21	1.64	13.44	13.64	12.78	46.77	92.04	454.15	611.47	90.59	125.29
		PL	65.97	0.18	**0.00**	1.89	94.07	73.28	8.72	50.54	53.11	44.99	88.45	236.47	3059.20	3102.04	449.45	710.91
		PI	2.46	**0.00**	**0.00**	0.10	3.44	2.78	0.66	1.88	2.33	1.69	5.67	19.13	795.05	1427.76	23.79	44.92
	1	*PD*	19.29	**0.00**	**0.00**	**0.00**	22.02	21.96	22.24	46.86	26.27	52.87	17.31	34.23	77.16	118.86	97.97	132.88
		PL	125.50	**0.00**	**0.00**	**0.00**	147.35	133.64	165.39	314.50	162.01	331.13	58.95	121.08	516.37	810.41	532.87	803.11
		PI	19.24	**0.00**	**0.00**	**0.00**	18.80	18.97	20.95	20.49	19.09	19.23	13.05	19.03	29.18	49.70	32.54	57.68
density	-	-	43.71	46.95	28.47	44.68	39.94	47.96	6.65	15.38	11.08	**0.00**	3.29	4.8	10.1	14.79	12.12	17.68
	0	*PD*	0.60	**0.00**	**0.00**	0.01	0.90	0.69	0.61	0.78	0.70	**0.00**	0.29	0.41	1.75	2.35	1.52	2.10
		PL	2.40	0.01	0.01	0.04	3.84	2.72	2.51	3.13	2.78	**0.00**	1.15	1.62	7.35	10.66	7.15	11.04
		PI	6.10	0.01	0.01	0.08	11.77	7.37	6.96	8.60	7.30	**0.00**	2.42	3.58	22.52	33.69	21.81	35.23
	1	PD	0.32	**0.00**	**0.00**	**0.00**	0.45	0.39	0.25	0.60	0.43	**0.00**	0.09	0.17	0.66	1.21	0.69	1.06
		PL	1.78	**0.00**	**0.00**	**0.00**	2.68	2.04	1.44	3.75	2.58	**0.00**	0.51	0.94	2.84	5.04	3.73	5.39
		PI	7.04	**0.00**	**0.00**	**0.00**	10.41	7.81	5.64	16.54	11.27	**0.00**	1.95	3.64	8.32	17.30	12.49	14.21
radial	-	-	205.27	216.86	172.73	202.44	196.24	233.76	49.75	102.54	78.48	**0.00**	25.93	38.34	84.06	119.18	90.77	115.35
	0	*PD*	3.30	4.17	1.67	3.17	3.74	6.46	0.48	1.19	0.86	**0.00**	0.20	0.29	34.17	46.66	3.84	12.07
		PL	19.65	27.58	11.64	21.85	23.97	44.58	2.20	6.59	4.40	**0.00**	0.86	1.12	214.34	277.03	20.39	68.70
		PI	18.11	23.74	16.06	23.37	19.76	33.39	1.57	5.97	3.60	**0.00**	0.51	0.87	92.99	154.38	6.78	21.56
	1	*PD*	1.44	1.37	0.59	1.18	1.51	2.56	0.57	2.00	1.36	**0.00**	0.22	0.35	5.36	8.83	11.45	11.37
		PL	8.07	7.33	3.60	6.51	8.36	13.99	3.00	10.82	7.31	**0.00**	1.19	1.88	29.14	48.57	63.27	62.90
		PI	4.40	4.12	3.23	4.25	4.33	6.26	1.01	4.01	2.59	**0.00**	0.39	0.66	9.44	17.20	24.42	24.86

Table 3.4.: Noise robustness of persistent homology on 1000 MNIST greyscale images. The remaining filtrations, and a description of the results are given in Table 3.5.

filtration	homological dimension	persistence signature	rotation 45	rotation -90	translation 1 1	translation -2 -2	stretch-shear-flip 1.5 10 h	stretch-shear-flip 0.75 -20 v	brightness -50	brightness 100	contrast 2	contrast 0.5	gaussian noise 10	gaussian noise 20	salt and pepper noise 5	salt and pepper noise 10	shot noise 50	shot noise 100
Rips	-	-	75.71	95.39	28.75	55.67	81.14	84.12	6.55	14.88	10.92	**0.00**	3.3	4.79	89.41	106.4	10.27	12.58
	0	*PD*	0.82	0.01	0.01	0.09	2.88	2.85	1.39	2.85	2.18	**0.00**	0.50	0.63	7.72	8.48	1.95	3.29
		PL	0.21	**0.00**	**0.00**	0.07	0.37	0.25	0.24	0.46	0.31	**0.00**	0.11	0.09	35.45	30.59	2.11	4.77
		PI	2.96	0.01	0.02	0.37	29.10	29.33	7.04	28.41	16.83	**0.00**	1.40	1.85	129.83	163.86	10.19	22.72
	1	*PD*	0.51	**0.00**	**0.00**	0.02	1.21	1.05	0.64	1.34	1.01	**0.00**	0.25	0.34	1.15	1.80	1.21	1.56
		PL	0.83	**0.00**	**0.00**	**0.00**	2.11	1.27	0.69	1.83	1.22	**0.00**	0.25	0.42	3.16	5.21	1.85	2.59
		PI	0.68	0.01	**0.00**	**0.00**	2.12	1.68	0.68	2.05	1.28	**0.00**	0.24	0.38	1.71	2.77	1.56	2.46
DTM	-	-	66.74	86.45	25.79	51.24	70.8	73.56	4.35	7.6	5.99	**0.00**	2.14	3.08	22.51	36.64	6.77	9.22
	0	*PD*	1.37	0.01	0.01	0.10	3.73	4.06	1.57	2.97	2.16	**0.00**	0.81	1.05	4.00	5.88	2.21	3.30
		PL	1.22	**0.00**	0.01	0.08	5.28	4.78	1.60	4.19	2.69	**0.00**	0.64	0.87	6.33	9.94	2.82	5.52
		PI	1.66	**0.00**	**0.00**	0.12	14.20	12.01	3.38	12.00	7.33	**0.00**	0.81	1.08	6.89	13.81	4.93	10.58
	1	*PD*	0.60	**0.00**	**0.00**	0.02	1.22	1.23	0.59	1.08	0.83	**0.00**	0.31	0.41	0.85	1.16	0.83	1.08
		PL	0.86	**0.00**	**0.00**	0.02	2.32	2.26	0.79	1.32	1.01	**0.00**	0.38	0.55	1.22	1.92	1.39	2.05
		PI	0.23	**0.00**	**0.00**	**0.00**	0.60	0.56	0.18	0.36	0.25	**0.00**	0.09	0.14	0.30	0.52	0.42	0.65

Table 3.5.: Noise robustness of persistent homology on 1000 MNIST greyscale images (cont). The table shows the distance $\|\phi - \psi\|_2$ between the filtration function values on the non-noisy and noisy image (1st row of each filtration), the Wasserstein distance $W_2(PD(\phi), PD(\psi))$ between 0- or 1-dimensional persistence diagrams (2nd and 5th row), the distance $\|PL(\phi) - PL(\psi)\|_2$ between persistent landscapes (3rd and 6th row), or the distance $\|PI(\phi) - PI(\psi)\|_2$ between persistent images (4th and 7th row), averaged across 1000 images in the MNIST data set.

Finally, Table 3.4 and Table 3.5 imply that stability theorems do not necessarily provide useful information about the stability in practice. For example, under rotation and Gaussian noise, the average value of $\|\phi_{\text{grsc}} - \psi_{\text{grsc}}\|_2$ is respectively equal to 2707.85 and 412.55. However, we see that the distance between PH on noisy and non-noisy images is close to zero for rotation, but it is much larger under Gaussian noise.

3.4.3 *Noise robustness and discriminative power*

In the previous section, we assess the distances between images and their noisy version. In practical applications, however, these distances ought to be compared to the distances between the images in (other classes of) the data set, which reflect the discriminative power in a classification task. Therefore, in this section, we discuss the noise robustness together with the discriminative power of persistent homology, across different filtrations and persistence signatures, for non-noisy and noisy data sets.

In order to do so, we investigate how much the performance of a classifier (more specifically, a support vector machine (SVM)) drops when noise is added to the data set. Since PDs are multi-sets, we use an SVM with a Gaussian kernel:

$$\kappa(Z, Z') = e^{-\frac{\delta^2(Z,Z')}{2\sigma^2}}.$$

For two images Z and Z', $\delta(Z, Z')$ corresponds to the Wasserstein W_2 distance[4] between their PDs, or the l_2 distance between their filtration function values, PLs or PIs. For each representation of the images, the SVM regularization parameter (typically noted as C, which trades off correct classification of training examples against maximization of the decision function's margin) and the kernel parameter σ^2 are first tuned using 5-fold cross-validation on the training set of 70% non-noisy images.[5] As we are focused on noise robustness, we calculate the relative decrease in accuracy for noisy compared to non-noisy test data (the remaining 30% of images in the data set). The results are summarized in Figure 3.7.

We observe that there is at least 35% drop in SVM accuracy, when images are represented with PH, in the following scenarios:

- affine transformations: PH on radial filtration under any affine transformation, and PH on Rips and DTM for stretch-shear-flip.
- brightness: PH on greyscale, Rips and DTM filtration.
- contrast: PH on greyscale filtration.

[4] The space of PDs with Wasserstein metric is not of negative type [340, Theorem 3.2], so that this kernel is not an inner product [57].

[5] We consider $C \in \{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and $\gamma = \frac{1}{2\sigma^2} \in \{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$.

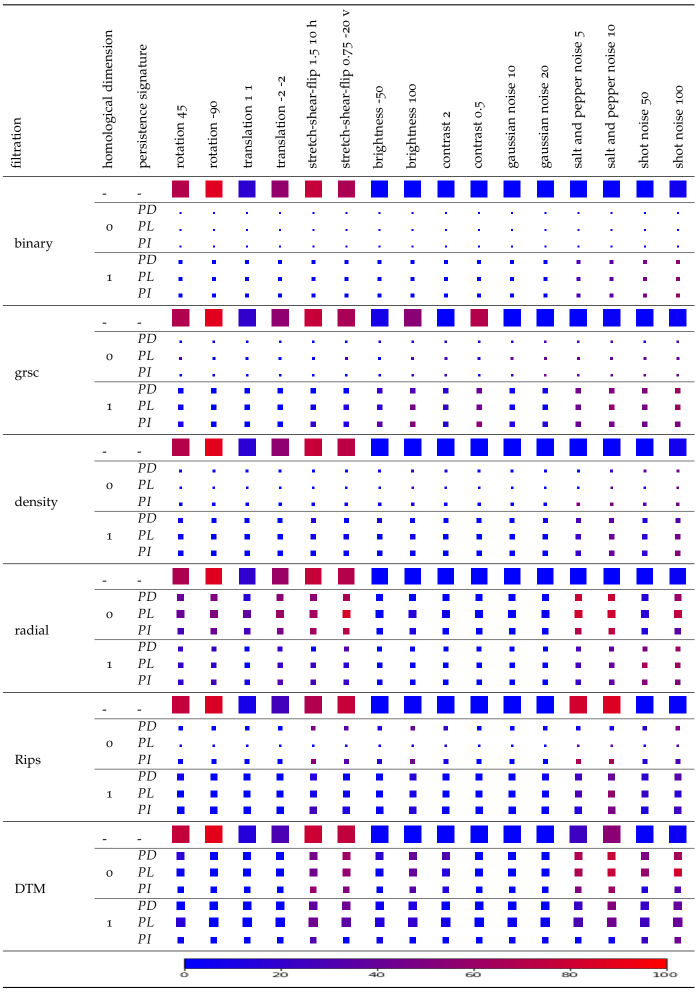


Figure 3.7.: Noise robustness and discriminative power of PH on 1000 MNIST greyscale images. The figure shows the drop in SVM classification accuracy when the test data set is noisy, compared to the non-noisy test set, averaged across 1000 images in the MNIST data set. Each image is represented either with its filtration function values (1st row of each filtration), or with its 0- or 1-dimensional PD (2nd and 5th row), PL (3rd and 6th row) or PI (4th and 7th row). The size of the node reflects the absolute accuracy on the non-noisy test data. The color of the node reflects the accuracy drop, indicated in the color bar. In particular, the presence of red nodes for PH information (2nd to 7th row) implies that PH is not robust under any type of noise, for any filtration and persistence signature.

- Gaussian, salt and pepper, shot noise: There is a drop in SVM accuracy for all filtrations under salt and pepper or shot noise. For Gaussian noise, the drop in accuracy is negligible for PH on all but the greyscale filtration.

The drop in accuracy also varies for different persistence signatures. For example, 1-dimensional PLs on Rips filtration are more sensitive to salt and pepper noise than PDs.

We note, however, that if the classification accuracy on the non-noisy data is low, the loss in performance is limited. For instance, 0-dimensional PH with respect to the binary filtration yields an accuracy of only 10% (not better than a random guess), as it only counts the number of components (Table 3.1), and every digit 0-9 commonly consists of a single component. This is why there is no drop in accuracy when SVM is tested on images under salt and pepper noise (Table 3.7), although this transformation results in an image with many additional connected components (Figure 3.5) and thus significantly changes PH on binary filtration. In these cases, however, the drop in accuracy gives us no reliable information about noise robustness.

When images are represented with their filtration function values on each pixel (including thus the original representation of an image as a vector of greyscale pixel values), the SVM performance is significantly worse for test data consisting of images with affine transformations. However, the performance is relatively stable under changes in image brightness or contrast, or under noisy transformations (with some exceptions). This is an opposite trend compared to PH, which is often robust under affine, but sensitive under noisy transformations (with a significant difference across filtrations and persistence signatures). Even though PH is often reputed for its robustness to noise [129], if data is expected to contain gaussian, salt and pepper or shot noise, the raw representation of images with their greyscale pixel values is robust to noise (there is no drop in SVM accuracy compared to non-noisy data), while this is often not the case for PH features.

Moreover, the absolute SVM accuracy on non-noisy data, when images are summarized with any persistence signature with respect to any filtration cannot compare with the representation of an image with its filtration function values, which contains more detailed geometric information about the image. Indeed, persistent homology only captures information about cycles in an image, and for most of the filtrations, it can only differentiate between two and three classes among the ten MNIST digits 0-9. The classification accuracy can be significantly improved by concatenating PH across different signatures, homological dimensions and/or filtrations (e.g., a combination of PH on radial and Rips filtration captures both information about the position and size of cycles, and can thus discriminate better across classes). For instance, a set of only 28 features obtained from concatenated persistent homology information is shown in [129] as sufficient to attain better classification accuracy than the set of greyscale pixel values. An alternative approach to simultaneously exploit PH from multiple filtrations is multi-dimensional persistence [55]. Since our goal is to gain insight into the noise robustness of individual PH representations, rather than maximizing the performance of classifiers, such an analysis is out of the scope of this work. However, under some types of transformations, the SVM accuracy is better for some PH features (even without concatenation across filtrations or signatures) compared to the raw representation with greyscale pixel values.

3.5 Conclusions

Persistent homology (PH), information about connected components, holes, and cycles in higher dimensions, is commonly characterized in the literature as a *topological* summary *robust to noise*. The main motivation behind this chapter is to illustrate how misleading this description can be, particularly for practical applications. We show that the validity of such a characterization, in theory, depends strongly on the choice of filtration and persistence signature (input and output of PH), and in practice, also on the particular application domain.

First of all, we emphasize that the type of information PH captures about cycles, is determined by the choice of filtration. For some filtrations, this information is only of *topological* nature, but for others, some *geometric* information can be captured as well. Topological invariants are robust under affine transformations, but the same does not necessarily hold for geometric invariants, so that the choice of filtration directly influences the noise robustness of PH.

Moreover, we underline how stability theorems, which provide a theoretical guarantee of the noise robustness of PH, depend on the choice of filtration and persistence signature, as well as the distance metric between them. Firstly, stability theorems make some assumptions about the filtration function, e.g., the function must be tame (the corresponding persistence diagram has finitely many off-diagonal points [62]), monotonic, continuous, Lipschitz or piecewise constant. Secondly, the robustness of PH is only guaranteed under small changes of the input - the filtration, rather than small changes in the space itself. For instance, if one background white pixel in an image is changed to black, the distance between the filtration functions for the common Vietoris-Rips filtration between these two images is large, and indeed, PH with respect to the Rips filtration is sensitive to such outliers. Furthermore, the statement of stability theorems is restricted to the particular choice of persistence signature and metric. This is often overlooked in the literature: e.g., it is common to employ persistence landscapes or persistence images and the Euclidean metric, whereas the stability theorems do not hold in such scenarios.

Finally, even if a stability theorem holds for the particular choice of filtration and persistence signature, it does not imply that PH yields noise-robust features in a classification task - this is domain and application-specific. For instance, changing a single pixel in an image from black to white can result in an additional one-pixel hole, which can be persistent for some filtrations. This change in PH is substantial if the number of holes does not vary greatly across classes of data, when the presence of such noise can be expected to deteriorate the classification accuracy. Reversely, even if there is no theoretical guarantee of the stability of PH for the given filtration and signature, it is interesting to evaluate the noise robustness in practice. To gain a better understanding of the noise robustness of PH, we carry out some computational experiments on the well-known MNIST data set of greyscale images, under some common types of noise to be expected on such data. We conclude that there is a considerable drop in accuracy of SVM trained on PH information of non-noisy and tested on noisy data, for at least 0- or 1-dimensional PH, for at least one of the considered signatures:

- rotation and translation: radial
- stretch-shear-flip: radial, Rips, DTM
- brightness: greyscale, Rips, DTM
- contrast: greyscale
- gaussian noise: greyscale
- salt and pepper, and shot noise: binary, greyscale, density, radial, Rips, DTM.

There is often also an important difference in the drop in accuracy across PDs, PLs and PIs. Taking all the above into consideration, it is clear that one needs to be more careful when referring to persistent homology as a noise-robust topological invariant: this is only true for some filtrations and signatures, and even in such cases, the stability of PH does not necessarily imply that the presence of noise will not weaken the discriminative power of PH features.

The main findings of this chapter provide some guidelines on the choice of suitable filtration(s) and persistence signature(s), and the corresponding metric, for the given data set and expected types of noise. Some important questions that should be addressed when using persistent homology are the following:

- choice of filtration: What information about cycles (number, brightness, position, size) is different across classes of data, but does not change much for the expected type of noise? Does the filtration function satisfy the assumptions in the stability theorem? Do small changes in the data result in small changes in the filtration function?
- choice of persistence signature: Is the signature stable? Are the cycles with the longest persistence or lifespan the most important (i.e., should cycles with short lifespans be considered as noise)? If this is not the case, it is

a good idea to use a flexible signature that allows cycles of e.g., medium persistence to be the most crucial (such as PIs with an appropriate weight function, what is not immediately possible with PDs or PLs). How critical are the important compared to unimportant cycles? Which statistical or machine learning methods do we want to apply to PH? If PH does not need to be summarized as a function or vector, it might be sufficient to use PDs. How important is computational efficiency? If the computation time is limited, it might be useful to avoid PDs and the expensive calculation of Wasserstein distances.

- choice of metrics: Is the signature stable? How critical are the important compared to unimportant cycles? The greater the p, the bigger is the difference across cycles, for both Wasserstein W_p or l_p metric, i.e., PH is less sensitive to unimportant cycles.

In summary, the choice of filtration defines the persistence of different types of cycles (e.g., for PH with Rips filtration, small cycles have short persistence), the choice of signature defines which cycles are least important or noisy (e.g., these are typically the cycles with short persistence), and together with the choice of metric determines the level of sensitivity to noisy cycles.

Our findings are limited to the particular setting in our computational experiments: the choice of filtrations and persistence signatures, and their parameters, the choice of metric, data set, noise, and classifier. For future research, it would be interesting to revisit similar research questions, but in a different context, e.g., for a different data set. The MNIST images of digits 0-9 all typically have one connected component, and none, one or two holes. Both noise robustness and classification accuracy are expected to be better for data sets where the number (but also other properties such as size) of cycles differ greatly across classes, such as images with complex structure which come from materials science, astronomy, neuroscience, plant morphology (e.g., images of cosmic web, protein networks, brain arteries, plant roots). Future experiments thus need to examine different (types of) data, and the relevant types of noise, filtrations and signatures for the given application.

To improve the ease of interpretation of stability theorems, for future theoretical work it would be useful to further bound the upper bound of PH (the right-hand side of stability theorems), $\|\phi - \psi\|_p = \|\phi - \phi \circ \pi\|_p$, with some reasonable distance $d(X, X')$ between the data X and its noisy variant $X' = \pi(X)$, for some common filtration functions ϕ and noise types π. For example, for the canonical Vietoris-Rips filtration, we know that $\|\phi - \psi\|_p = d_H(X, X')$; however the Hausdorff distance can be large if X' is X under rotation. However, [63] show that distance between PDs is actually upper-bounded by the Gromov-Hausdorff distance $d_{GH}(X, X')$ that is a more useful bound, e.g., $d_{GH}(X, X') = 0$ for rotation and indeed, PD with respect to the Rips filtration remains unchanged under rotation. Comparing these bounds across filtrations can help guide the choice of filtration for the given problem and expected type of noise.

Finally, as discussed in [69], while stability is crucial, we note that it might be an overly stringent requirement for many data science problems. Indeed, stability is only achievable if short persistent intervals are considered as less important, whereas there is no inherent reason why this should be the case in practice (for instance, short intervals are crucial for convexity detection in Section 2.4, and medium-length intervals are the most useful in classifying the brain artery tree data in Appendix B.5). For practical applications, it might therefore often be beneficial to consider persistence signatures that have no theoretical guarantee of stability.

Part II

APPLICATIONS OF PERSISTENT HOMOLOGY

4

THE HOW:
Topological data analysis of structure preservation via hyperdimensional computing

In the last decades, a number of computational models have been introduced that are founded on the mathematical properties of very high dimensionality, which are often collectively known as Vector Symbolic Architecture. These include hyperdimensional computing (HDC), a hardware-efficient and noise-robust computational framework that enables on-chip learning, which is becoming more important going toward Internet of Things domain. HDC relies on encoding the input data into a very high dimensional HD space (typically, $D = 10\,000$). In this chapter, we carry out an experimental study on some well understood shapes, in addition to benchmark real-world data, with the main goal of exploring the structure of input data in the HD space. For instance, a simple question of interest is whether a (noisy) circle remains a circle in the HD space? In general, what kind of topological and geometric structure in the input space, described with persistent homology (PH), is preserved in the HD space? What is the discriminative power of the HD representation? How are different types of noise tolerated in the HD space?

This chapter is based on the following publication:

> Renata Turkeš, Werner van Leekwijck, and Steven Latré, *Topological data analysis of structure preservation via hyperdimensional computing*, under review.

4.1 Introduction

In the last decades, a number of computational models have been introduced that are founded on the mathematical properties of very high dimensionality and randomness, such Holographic Reduced Representations, Spatter Code, Sparse-Distributed Memory, Semantic Vectors, Latent Semantic Analysis, Context-Dependent Thinning, which are often collectively known as Vector Symbolic Architecture [178]. These include hyperdimensional computing (HDC) [177], a hardware-efficient and noise-robust computational framework that enables on-chip learning, which is becoming more important going toward Internet of Things domain. HDC relies on encoding the input data into a hyperdimensional (HD) space (typically, $D = 10\,000$), and performs the classification based on the distance to the HD class representatives.

There are a few theoretical studies of the behavior of HDC, but which provide little intuition on how and why it works. For example, recent work [333] proves that the HD encoding approximately preserves l_1 distance and cluster structure, but with no insights on the tightness of the bounds in practice. On the other hand, HDC has been applied in various domains, ranging from analogy-based reasoning, language, speech and activity recognition, brain-computer interfaces, text or image classification [131, 161] (see [285] for an overview). However, the focus of these papers is the performance of HDC (accuracy of the HDC classifier), and the real-world data in these applications is very complex, so that they provide little insight into the workings of HDC, and in particular, of the HD data space.

RESEARCH QUESTIONS To close this gap, in this chapter we aim to provide some intuition on why the HD encoding is reasonable, mainly by exploring the structure of input data in the HD space. For instance, a simple question of interest is whether a (noisy) circle remains a circle in the HD space? More precisely, we focus on the similar questions that were posed in the recent review of theoretical foundations of HDC [333]:

(RQ1) What kind of topological and geometric structure in the input space is preserved in the HD space?

(RQ2) What is the power of linear separators in the HD space, compared to the input space?

(RQ3) To what extent is noise tolerated in the HD space?

METHODOLOGY Our approach to addressing the questions, however, is different. Firstly, we replace rigorous proofs with intuitive exposition of the HDC encoding, where we focus on its distance preservation (Section 4.2). Secondly, we also carry out some simple experiments to analyze the HDC encoding of some well understood

Euclidean data. We consider a number of synthetic data sets that reflect mathematical shapes with distinctive topological and geometric features (such as a circle, concentric circles, lemniscate or a star), and complete the analysis with some real-world data (UCI-HAR, CTG, and ISOLET, respectively benchmark activity recognition, ardiotocography and speech recongition data) (Section 4.3). To address (RQ1), we start by looking into the pairwise distances, distributions of distances and norms of the input and HD space, and assess the correlations between them (Section 4.4). Next, we rely on topological data analysis (TDA) and calculate persistent homology (PH) of the data to compare their topology and geometry (Section 4.5). Data separability in both the input and HD space is evaluated with the Dunn index, and we quantify its linear separability (RQ2) with the classification accuracy of logistic regression (Section 4.6). In each of the experimental sections, we investigate the behavior of the input and HD data in presence of different types of noise (RQ3).

RELATED WORK When it comes to the main goal of this chapter - studying the structure preservation via the HDC encoding - the article that is most closely related is [333]. This is one of the few papers that presents a unified treatment of the theoretical foundations of HDC. In particular, Theorems 17 and 18 show that HDC encodings preserves distances, which is a sufficient condition for preservation of cluster structure, robustness to noise, and sometimes the preservation of linear separability, which are also discussed in the paper. However, due to the heavy theory, the work remains unapproachable to a wider community, and provides little intuition on to what extent and why HD preserves distances, in particular since no examples or experiments are carried out. Distance preservation has also been proved earlier for sparse binary vectors [283].

Moreover, there are a number of other works that provide an empirical comparison of the different HDC encodings, such as [189, 322, 323]. However, the focus of these papers is to adjust the encoding to improve the classification performance, and not to understand why it works (and in particular, what aspects of structure in the input space are preserved in the HD space).

When it comes to employing PH to analyze structure preservation *in general* (not on HDC), there are a few works in this direction. For example, PH has been used to study dimensionality reduction techniques [265, 296], or used to define encodings that will preserve the topology and/or geometry [105, 252]. Similarly (if one thinks of the output layer of a neural network as an encoding of input data), PH has been used to study how the structure in the input data manifold [251, 266, 357, 382] or decision boundary [149, 287] is changed through the layers. However, to the best of our knowledge, this is the first time PH has been used to understand (or, in any combination with, for that matter) HDC.

4.2 Background on HDC encoding

Let $\mathcal{X} \subset \mathbb{R}^N$ be our data. Hyperdimensional computing (HDC) relies on encoding the input data $\mathcal{X}$ into some inner-product space $\mathcal{H} \subset \mathbb{R}^D$ via the HDC encoder, a function $\phi : \mathcal{X} \to \mathcal{H}$ which sends each data sample $X \in \mathcal{X}$ into an HD vector $H = \phi(X) \in \mathcal{H}$. The dimension D is very large, $D \gg N$, and one commonly assumes $D = 10\,000$. Note that we write $\mathcal{X} \subset \mathbb{R}^N$ and $\mathcal{H} \subset \mathbb{R}^D$, but non-Euclidean distances can be considered.[1]

The primary aim of this section is to provide some intuition on why such an encoding approximately preserves distances. We illustrate this with the common Position-ID encoding of Euclidean data[2], and point to the properties and parameters of the HDC encoding (construction of seed vectors, choice of bundle and bind operations, the dimension D and alphabet size K) that influence to what extent are the distances preserved. Theoretical guarantee that provides the bounds for this approximation of distances is given in [333, Theorem 17], but rather than providing a formal proof, here we focus on the clarity of exposition for a broader audience.

Let $X = (x_1, x_2, \ldots, x_N) \in \mathcal{X} \subset \mathbb{R}^N$. Note that

$$X = x_1 e_1 + x_2 e_2 + \ldots x_N e_N,$$

where vectors $e_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ (with i-th coordinate being nonzero) form an orthonormal basis, $\|e_i\|_2 = 1$ and $\langle e_i, e_j \rangle = 0$ whenever $i \neq j$. For example, $X = (3, 7, -2) = 3e_1 + 7e_2 - 2e_3$. In a similar way, one starts by constructing HD seed binary or bipolar vectors, which can be thought of as primitives or atoms from which further representations are made [178]: approximately orthogonal basis vectors $F_1, F_2, \ldots, F_N$ that encode the features (Section 4.2.1), and $V_1, V_2, \ldots, V_K$ that encode their values (Section 4.2.2). To encode the data samples $X \in \mathcal{X}$, these seed vectors are combined using the so-called operations of binding and bundling:

$$H = \phi(X) = (V(x_1) \odot F_1) \oplus (V(x_2) \odot F_2) \oplus \cdots \oplus (V(x_N) \odot F_N).$$

For best clarity, here we assume that bundle $\oplus$ and bind $\odot$ are respectively the coordinate-wise sum and product[3], and we thus write

$$H = \sum_{i=1}^{N} V(x_i) F_i.$$

The similarity of HD vectors is defined as their scalar product.

[1] If $\mathcal{H} \subset \{0, 1\}^D$, it is reasonable to consider the Hamming distance that counts the number of coordinates where two vectors differ.

[2] For other type of data (e.g., vectors with categorical or ordinal values, sets, images, text or time series), different choices are suitable, e.g., such as record-based encoding and N-gram-based encoding [284, 286, 322].

[3] Finally, one could threshold these integer vectors into $H \in \{-1, 1\}^D$, which would lead to another level of approximation that depends on the input dimension or number of features N.

4.2.1 *Seed ID vectors F*

Each feature, or its ID, is represented with a random bipolar HD vector $F \in \{-1, 1\}^D$[4]. An important property of hyperdimensional spaces is that two random vectors are nearly orthogonal:

$$\langle F_i, F_j \rangle \approx 0 \tag{F}$$

so that these ID vectors represent some approximately orthogonal basis.[5] Coordinates of a seed HD vector can be drawn at random according to different probability distributions [178]. In this work, the coordinates are sampled from uniform distribution, yielding dense vectors with a balanced number of -1s and 1s, but other distributions resulting possibly in sparse vectors is also possible.[6]

The quality of this approximation can be formalized with the notion of incoherence, popularized in the sparse coding literature: the HDC encoding of seed ID vectors F_i is μ-incoherent if $|\langle F_i, F_j \rangle| \leq \mu \min_i \|F_i\|$. The larger the encoding dimension D, the more likely it is that $\mu = 0$ (Figure 4.1).

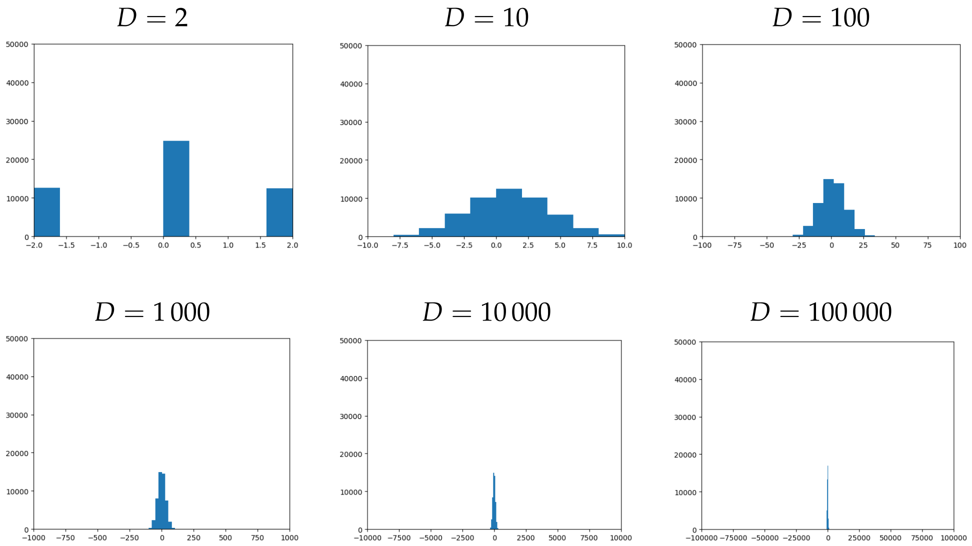


Figure 4.1.: The approximation $\langle F_i, F_j \rangle \approx 0$ for two random vectors $F_i, F_j \in \{-1, 1\}^D$ improves with an increase in dimension D.

[4] Alternatively, it is also common to consider binary vectors $F \in \{0, 1\}^D$.

[5] In real-world data, features are often correlated, so that an orthogonal basis might not make the most sense: better results can at times be achieved by RFF basis whose construction is guided by the input data.

[6] Sub-Gaussian probability distribution are sufficient for the wanted behaviour of the distance between two random vectors[333].

4.2.2 *Seed Position vectors V*

Continuous ranges of feature values are mapped to Position HD vectors, efficiently and in a hardware-friendly way [161]. Let us assume that the input data $\mathcal{X}$ is normalized (for each data sample $X = (x_1, x_2, \dots, x_N) \in \mathcal{X}$, $x_i \in [0,1]$ for every $i \in \{1, 2, \dots, N\}$), and let the unit interval $[0,1]$ be split in K sub-intervals of equal length. We construct HD vectors $V_1, V_2, \dots, V_K$, one for each of the sub-intervals, or their centroids $v_1, v_2, \dots, v_K \in \mathbb{R}$. This time, unlike for the construction of seed ID vectors $F_1, F_2, \dots, F_N$, we do not want the vectors to be orthogonal or unrelated, but rather we aim that the distance between V and V' reflects the distance $|v - v'|$.[7]

The construction is as follows. The first range of values, i.e., the first sub-interval or its centroid $v_1 \in \mathbb{R}$ is represented with a random bipolar vector $V_1 \in \{-1, 1\}^D$. To represent each next centroid v_{i+1} with a vector V_{i+1}, we flip $\lceil \frac{1}{K-1} \frac{D}{2} \rceil$ random coordinates of the previous HD vector V_i, making sure that each coordinate can be flipped at most once. In this way, V_1 and V_K differ in $D/2$ coordinates. An $x \in \mathbb{R}$ is then approximated with its centroid v, and encoded as $V(x) = V(v)$.

For example, for $K = 5$, we split the unit interval in the following way: $[0,1] = [0, 0.2) \cup [0.2, 0.4) \cup [0.4, 0.6) \cup [0.6, 0.8) \cup [0.8, 1]$. We would construct five Position vectors $V_1 = V(v_1), \dots, V_5 = V(v_5)$ that represent the sub-intervals, or e.g., their centroids $v_1 = 0.1, v_2 = 0.3, v_3 = 0.5, v_4 = 0.7, v_5 = 0.9$. V_1 is random, V_2 is constructed from V_1 by flipping $\frac{1}{4} * \frac{10\,000}{2} = 1\,250$ random coordinates, V_3 is constructed from V_2 by flipping some other $1\,250$ random coordinates (so that V_1 and V_3 differ in $2\,500$ coordinates), and so on. If we consider some arbitrary values, $x_1 = 0.1, x_2 = 0.15, x_3 = 0.79, x_4 = 0.81$, we have that $V(x_1) = V(x_2) = V(v_1) = V_1$, $V(x_3) = V(v_4) = V_4$, $V(x_4) = V(v_5) = V_5$.

We now show that the distance between values in $x, x' \in \mathbb{R}$ is approximately reflected by the distance between HD vectors $V = V(x), V' = V(x') \in \{-1, 1\}^D$ that correspond to these values. Firstly, note that the above construction ensures that the probability that coordinates of V and V' differ is the following:

$$p(V \neq V') = |v - v'| \cdot \frac{K}{D} \cdot \left\lceil \frac{1}{K-1} \frac{D}{2} \right\rceil \approx |v - v'| \cdot \frac{K}{K-1} \cdot \frac{1}{2} \approx \frac{1}{2} |v - v'|.$$

These first approximations follow from removing the ceiling (in practice this has no influence), and from $\frac{K}{K-1} \approx 1$, that depends on the how fine-grained is the split of the unit interval $[0,1]$, i.e., how small is the length of the sub-interval or, equivalently, how large is the number K of sub-intervals. Then, since HD vectors Vs are bipolar, and by the definition of the scalar product:

[7] If a feature represents a color, with values $\{\texttt{red}, \texttt{green}, \texttt{blue}\}$, such an encoding would not make sense, since we have no clear understanding of the relationship between these colors, so that it would be best that mutually orthogonal vectors correspond to the three colors.

$$\begin{aligned}\langle V(x), V(x')\rangle &= D[1 * p(V = V') + (-1) * p(V \neq V')] \\ &\approx D[1 * (1 - \frac{1}{2}|v - v'|) + (-1) * \frac{1}{2}|v - v'|] \\ &= D(1 - |v - v'|) \\ &\approx D(1 - |x - x'|) \end{aligned} \tag{V}$$

The second approximation stems from $|x - x'| \approx |v - v'|$ which again decreases with larger K, since values $x \in \mathbb{R}$ are more closely approximated with their centroids v (Figure 4.2).

We illustrate the procedure with the toy example below.

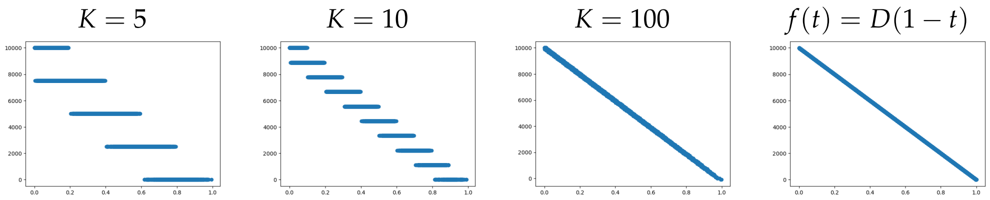

$K = 5$ $K = 10$ $K = 100$ $f(t) = D(1 - t)$

Figure 4.2.: The approximation $\langle V(x), V(x')\rangle \approx D(1 - |x - x'|)$ improves with K, approaching the ground truth. The plots show the relationship between $\langle V(x), V(x')\rangle$ on the y-axis and $|x - x'|$ on the x-axis, where $x, x' \in [0, 1]$ are random.

4.2.3 *Distance preservation*

Let $X, X' \in \mathcal{X}$, and let $H = \phi(X) \in \mathcal{H}$ and $H' = \phi(X') \in \mathcal{H}$. In this section, we aim to show that $\|H - H'\|_2^2 \approx 2D\|X - X'\|_1$, meaning that the HDC encoding ϕ approximately preserves distances. We will rely on the property (F) of the ID vectors, property (V) of Position vectors (V), and property of the bundle and bind operations (P1)-(P3) In addition, it is important to note that the randomness of ID vectors F and the first Position vector V_1 imply their independence, and since (F) ensures that $p(F_i = F_j) \approx p(F_i \neq F_j) \approx \frac{1}{2}$, the following holds:

$$\begin{aligned} p(VF_i \neq V'F_j) &= p(V \neq V')p(F_i = F_j | V \neq V') + p(F_i \neq F_j)p(V = V' | F_i \neq F_j) \\ &= p(V \neq V')p(F_i = F_j) + p(F_i \neq F_j)p(V = V') \\ &\approx \frac{1}{2}p(V \neq V') + \frac{1}{2}[1 - p(V \neq V')] \\ &= \frac{1}{2} \end{aligned}$$

Therefore, since the seed ID and Position vectors are bipolar, the definition of the scalar product implies that:

$$\langle VF_i, V'F_j \rangle \approx 0 \tag{FV}$$

The quality of approximation is the same as in (F), improving with larger dimension D of the HD space.

We now start the proof of distance preservation with the parallelogram law that implies the following:

$$\|H - H'\|_2^2 = \|H\|_2^2 + \|H'\|_2^2 - 2\langle H, H' \rangle.$$

Firstly, note that, property (FV) immediately implies that:

$$\begin{aligned} \|H\|_2^2 &= \langle H, H \rangle \\ &= \Big\langle \sum_{i=1}^{N} V(x_i)F_i, \sum_{j=1}^{N} V(x_j)F_j \Big\rangle \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \langle V(x_i)F_i, V(x_j)F_j \rangle \\ &= \sum_{i=1}^{N} \langle V(x_i)F_i, V(x_i)F_i \rangle + \sum_{i=1}^{N}\sum_{j \neq i} \langle V(x_i)F_i, V(x_j)F_j \rangle \\ &\approx \sum_{i=1}^{N} D + 0 \\ &= ND \end{aligned}$$

Secondly, using the properties (FV) and (V), we have:

$$\begin{aligned}
\langle H, H' \rangle &= \Big\langle \sum_{i=1}^{N} V(x_i)F_i, \sum_{j=1}^{N} V(x'_j)F_j \Big\rangle \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \langle V(x_i)F_i, V(x'_j)F_j \rangle \\
&= \sum_{i=1}^{N} \langle V(x_i)F_i, V(x'_i)F_i \rangle + \sum_{i=1}^{N}\sum_{j\neq i} \langle V(x_i)F_i, V(x'_j)F_j \rangle \\
&\approx \sum_{i=1}^{N} \langle V(x_i), V(x'_i) \rangle + 0 \\
&\approx \sum_{i=1}^{N} D(1 - |x_i - x'_i|) \\
&= ND - D\sum_{i=1}^{N} |x_i - x'_i| \\
&= ND - D\|X - X'\|_1
\end{aligned}$$

Putting the two together, we obtain:

$$\|H - H'\|_2^2 \approx ND + ND - 2ND + 2D\|X - X'\|_1 = 2D\|X - X'\|_1.$$

Note that the strength of the approximation depends on the dimension D that influences the level of orthogonality of the ID vectors F, and on the number K of sub-intervals that determines how fine-grained is the representation of feature values.

4.3 (Noisy) Data sets

We aim to gain some understanding about the type of structures that are preserved with the HDC encoding $\phi : \mathcal{X} \to \mathcal{H}$. To this end, we consider a number of well-defined synthetic shapes in $\mathbb{R}^N$ (Figure 4.4):

- Bumpy circle, Astroid, Apple in $\mathbb{R}^2$: A circle, with some bumps or singularities, in order to asses to what extend HDC preserves loops and the level of detail.
- Lemniscate in $\mathbb{R}^2$: Two loops of the same size, with self-intersection.
- Concentric circles in $\mathbb{R}^2$: Two loops of different size, one within the other.
- Fish, Star in $\mathbb{R}^2$: A number of loops created with a number of self-intersections.

- Olympics in $\mathbb{R}^2$: Five circles of the same size, with some interesections that result in 9 loops.
- Orthogonal circles in $\mathbb{R}^4$: Six orthogonal circles.
- Nine squares in $\mathbb{R}^2$: Nine connected components and loops of the same size.
- Cube vertices in $\mathbb{R}^3$: Nine vertices of a cube.

To generate a data set, we randomly sample 500 points (data samples) on the shape. Each of data sets is normalized, so that each coordinate is in [0, 1]. Later in the paper, we also look into the data separability, and we therefore label the data sets. This can be done in arbitrarily many different ways, and we choose some reasonable labels while ensuring that different data sets exhibit different levels of (linear) separability.

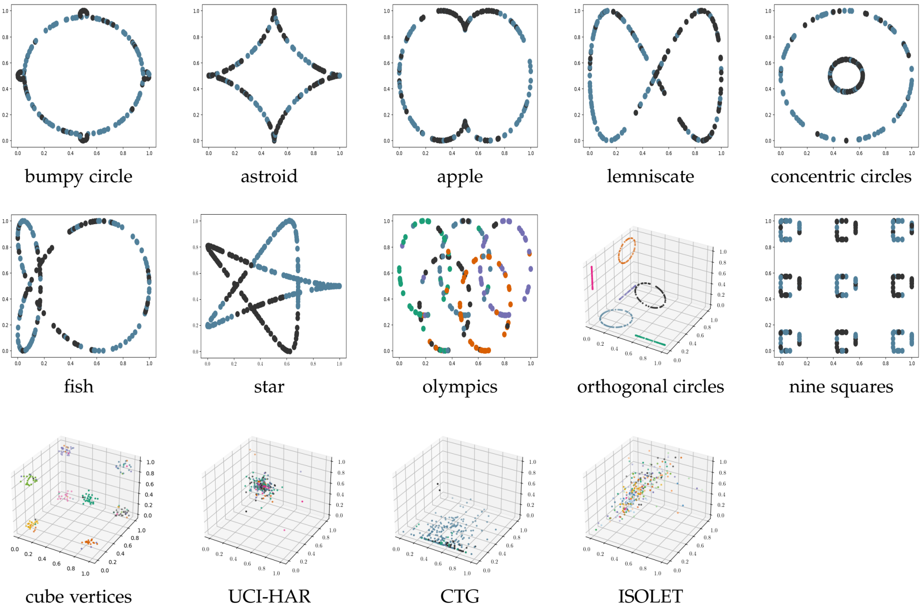
Moreover, we also include the experiments on publicly available Human Activity Recognition (UCI-HAR) [12], Cardiotocography (CTG) and ISOLET data [183] that are commonly used in HDC-related research:

- UCI-HAR: This mobile sensor multivariate time series data was collected from experiments with a group of 30 volunteers who performed 6 activities (walking, walking upstairs, walking downstairs, sitting, standing, laying). UCI-HAR is considered one of the popular data sets used for benchmarking activity recognition [6]. It consists of 10 299 data points with 561 time and frequency domain features, separated into 6 classes that correspond to the performed activity.
- CTG: The data set consists of 21 fetal heart rate and uterine contraction features on 2 126 cardiotocograms, classified into three fetal states by expert obstetricians.
- ISOLET: 150 subjects spoke each of the 26 letters of the alphabet twice, resulting in 7 797 time series data points with 617 features.

Each of the three data sets is also normalized to [0, 1], and we consider 500 randomly chosen data points.

As already indicated, we focus on encoding Euclidean data, and therefore consider $(\mathcal{X}, l_2)$ in the analysis. Since we showed that $\|H - H'\|_2^2 \approx 2D\|X - X'\|_1$ (Section 4.2.3), we assume $(\mathcal{H}, l_2)$, and also consider $(\mathcal{X}, l_1)$ in our experiments.

HDC is often credited for its noise robustness, which motivates us to consider two different types of noise:



bumpy circle, astroid, apple, lemniscate, concentric circles, fish, star, olympics, orthogonal circles, nine squares, cube vertices, UCI-HAR, CTG, ISOLET

Figure 4.3.: Samples from synthethic and benchmark data sets in $\mathbb{R}^N$. For $N \geq 3$, the first three coordinates are plotted.

- Gaussian noise on the input data ($\mathcal{X}', \mathcal{X}''$). HDC is supposed to handle this noise well due to its low precision: an interval of values in $\mathbb{R}$, or its centroid, is encoded with a single Position HD vector V (Section 4.2.2). Indeed, for small Gaussian noise, x and $x + \varepsilon$ often lie in the same sub-interval, and will thus be represented with the same vector, so that the noise will have no effect.

- Component failure on the HD data ($\mathcal{H}', \mathcal{H}''$). Because HD vector is a very high-dimensional distributed representation (Section 4.2), no component is more responsible to store any piece of information than another, so that it is robust against errors in its components [161].

In the remainder of this chapter, we visualize the detailed experimental results only for the bumpy circle; results for the additional data are listed in Appendix C. For better clarity, the main table results in the following sections are color-coded, where the darkest cells highlight the best values for the given data set (e.g., best preservation of persistent homology - the minimum Wasserstein distance, or best linear separability - maximum classification accuracy), and lighter cells indicate poorer performance.

4.4 Distance preservation

MDS EMBEDDING For each of the data sets, we start by calculating its Multi-Dimensional Scaling (MDS) embedding into $\mathbb{R}^2$. Figure 4.5 visualizes the embedding for the bumpy circle, and visualization of the other data sets is given in Appendix C. The different colors in these MDS plots represent the different data classes.

For $(\mathcal{H}, l_2)$, this is a projection from $D = 10\,000$ into 2 dimensions, so that it is not possible to draw any reliable conclusions; these plots only serve as the first visualization and some intuition behind structure preservation. We therefore also study the distance matrices, the distributions of distances and norms in this section, and primarily - the persistent homology (Section 4.5), in order to assess the preservation of topological and geometric structure. To gain some insights in data separability, we also calculate the Dunn index and logistic regression accuracy (Section 4.6).

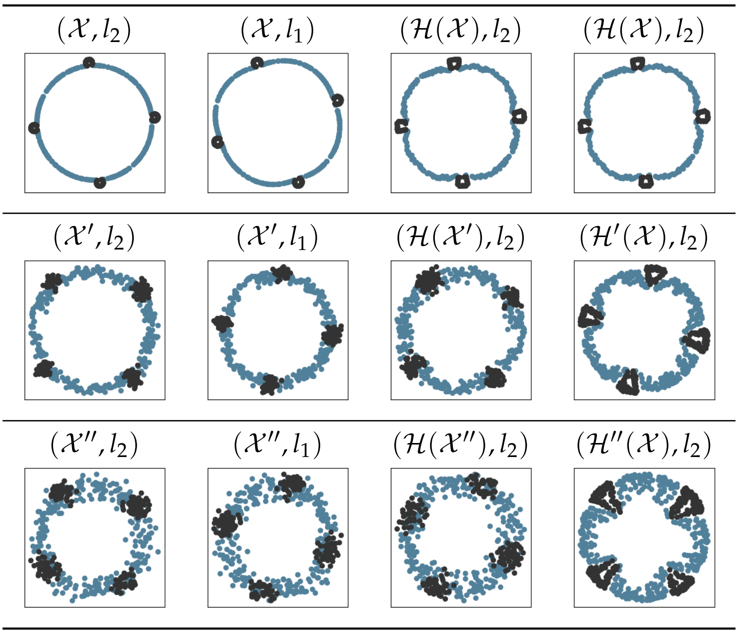


Figure 4.4.: MDS embedding of bumpy circle into $\mathbb{R}^2$.

DISTANCE MATRIX To get an idea of the distance preservation with the HDC encoding, we start by plotting the distance matrices. Figure 4.6 visualizes the distance matrices for the bumpy circle (with results for the other data sets in Appendix C), suggesting a strong preservation of structure. We quantify this with the Spearmann correlation with the l_2 and l_1 distance matrix on the input data $\mathcal{X}$.

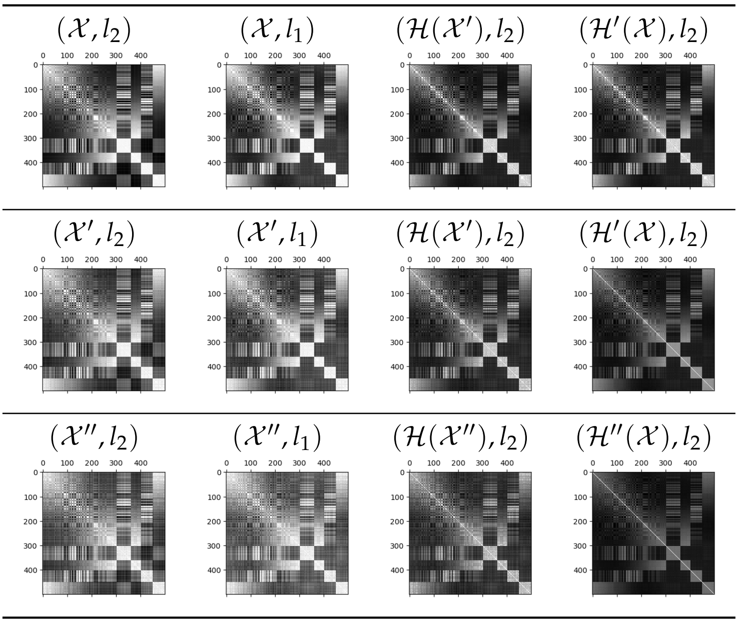


Figure 4.5.: Distance matrices for the bumpy circle.

Table 4.1 shows the Spearmann correlation between the given matrix and the l_2 distance matrix on the input data $\mathcal{X}$. The correlations are very high, and we can see that $(\mathcal{H}, l_2)$ exhibits the same behaviour as $(\mathcal{X}, l_1)$, with the correlation remaining the same even if some random HD coordinates are corrupted. Moreover, the correlation with the HD encoding of the input data under Gaussian noise mimics the correlation of the l_1 distance matrix on the noisy input data.

Table 4.2 shows the Spearmann correlation between the given matrix and the l_1 distance matrix on the input data $\mathcal{X}$. Spearmann correlation assesses the monotonoic relationship, and it therefore comes as no surprise that the correlation between the distance matrices $(\mathcal{H}, l_2)$ and $(\mathcal{X}, l_1)$ is equal to 1 for any data set. This means that if $\|X - X'\|_1 \leq \|X - X''\|_1$, then $\|\phi(X) - \phi(X)'\|_2 \leq \|\phi(X) - \phi(X'')\|_2$. This correlation mostly remains unchanged under random corruption of HD data. For Gaussian noise, the correlation with l_2 distance on the HD representation is greater than the correlation with l_1 on the noisy data, suggesting that the HD representation can handle the Gaussian noise better than the input data.

DISTRIBUTION OF DISTANCES The relationship between pairwise l_2 distances in the HD space and the l_1 distances in the input space is monotonic (Table 4.2), and it is approximately monotonic for l_2 distances (Table 4.1). Moreover, the distance matrices suggest preservation of structure, but the distance matrices in the HD space are darker (Figure 4.6, with results for the other data sets in Appendix C).

Data	$(\mathcal{X},l_2)$	$(\mathcal{X}',l_2)$	$(\mathcal{X}'',l_2)$	$(\mathcal{X},l_1)$	$(\mathcal{X}',l_1)$	$(\mathcal{X}'',l_1)$	$(\mathcal{H}(\mathcal{X}),l_2)$	$(\mathcal{H}(\mathcal{X}'),l_2)$	$(\mathcal{H}(\mathcal{X}''),l_2)$	$(\mathcal{H}'(\mathcal{X}),l_2)$	$(\mathcal{H}''(\mathcal{X}),l_2)$
Bumpy circle	1.00	0.99	0.96	0.91	0.90	0.88	0.91	0.90	0.88	0.91	0.91
Astroid	1.00	0.98	0.95	0.93	0.91	0.88	0.93	0.91	0.87	0.93	0.93
Apple	1.00	0.99	0.97	0.96	0.95	0.93	0.96	0.95	0.94	0.96	0.96
Lemniscate	1.00	0.99	0.97	0.97	0.97	0.95	0.97	0.97	0.95	0.97	0.97
Concentric circles	1.00	0.98	0.95	0.97	0.96	0.93	0.97	0.96	0.93	0.97	0.97
Fish	1.00	0.99	0.98	0.98	0.98	0.96	0.98	0.97	0.96	0.98	0.98
Star	1.00	0.99	0.96	0.98	0.97	0.94	0.98	0.97	0.94	0.98	0.98
Olympics	1.00	0.99	0.96	0.98	0.97	0.94	0.98	0.97	0.94	0.98	0.98
Orthogonal circles	1.00	0.99	0.97	0.97	0.96	0.94	0.96	0.96	0.94	0.96	0.96
Nine squares	1.00	0.99	0.97	0.98	0.97	0.96	0.97	0.97	0.95	0.97	0.97
Cube vertices	1.00	0.99	0.96	0.99	0.97	0.96	0.99	0.97	0.95	0.98	0.98
UCI-HAR	1.00	1.00	1.00	0.98	0.99	0.99	0.98	0.99	0.99	0.98	0.98
CTG	1.00	0.99	0.98	0.97	0.96	0.95	0.96	0.95	0.94	0.96	0.96
ISOLET	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.97

Table 4.1.: Distance preservation: Spearmann correlation with the distance matrix for $(\mathcal{X},l_2)$.

Data	$(\mathcal{X},l_2)$	$(\mathcal{X}',l_2)$	$(\mathcal{X}'',l_2)$	$(\mathcal{X},l_1)$	$(\mathcal{X}',l_1)$	$(\mathcal{X}'',l_1)$	$(\mathcal{H}(\mathcal{X}),l_2)$	$(\mathcal{H}(\mathcal{X}'),l_2)$	$(\mathcal{H}(\mathcal{X}''),l_2)$	$(\mathcal{H}'(\mathcal{X}),l_2)$	$(\mathcal{H}''(\mathcal{X}),l_2)$
Bumpy circle	0.91	0.91	0.89	1.00	0.98	0.93	1.00	0.98	0.98	1.00	1.00
Astroid	0.93	0.91	0.88	1.00	0.97	0.91	1.00	0.96	0.96	1.00	1.00
Apple	0.96	0.95	0.93	1.00	0.99	0.97	1.00	0.99	0.99	1.00	1.00
Lemniscate	0.97	0.97	0.95	1.00	0.99	0.97	1.00	0.99	0.99	1.00	1.00
Concentric circles	0.97	0.96	0.93	1.00	0.99	0.94	1.00	0.99	0.99	1.00	1.00
Fish	0.98	0.97	0.96	1.00	0.99	0.98	1.00	0.99	0.99	1.00	1.00
Star	0.98	0.97	0.94	1.00	0.99	0.96	1.00	0.99	0.99	1.00	1.00
Olympics	0.98	0.97	0.94	1.00	0.99	0.95	1.00	0.99	0.99	1.00	1.00
Orthogonal circles	0.97	0.96	0.93	1.00	0.99	0.96	1.00	0.99	0.99	1.00	1.00
Nine squares	0.98	0.97	0.95	1.00	0.99	0.97	1.00	0.99	0.99	1.00	1.00
Cube vertices	0.99	0.97	0.95	1.00	0.98	0.96	1.00	0.98	0.98	1.00	1.00
UCI-HAR	0.98	0.98	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
CTG	0.97	0.96	0.94	1.00	0.99	0.97	1.00	0.99	0.99	1.00	0.99
ISOLET	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.98

Table 4.2.: Distance preservation: Spearmann correlation with the distance matrix for $(\mathcal{X},l_1)$.

To gain more insights into this behaviour, we plot the distribution of distances. Figure 4.7 visualize these distributions for the bumpy circle (plots for other data sets can be found in Appendix C), clearly showing that the distances in the HD space are relatively larger. This comes as no surprise since we show in Section 4.2.3 that $\|H - H'\|_2 \approx \sqrt{2D\|X - X'\|_1}$, implying that the small distances in the input space become relatively more pronounced in the HD space. This effect is pronounced even more when a number of random HD coordinates is corrupted.

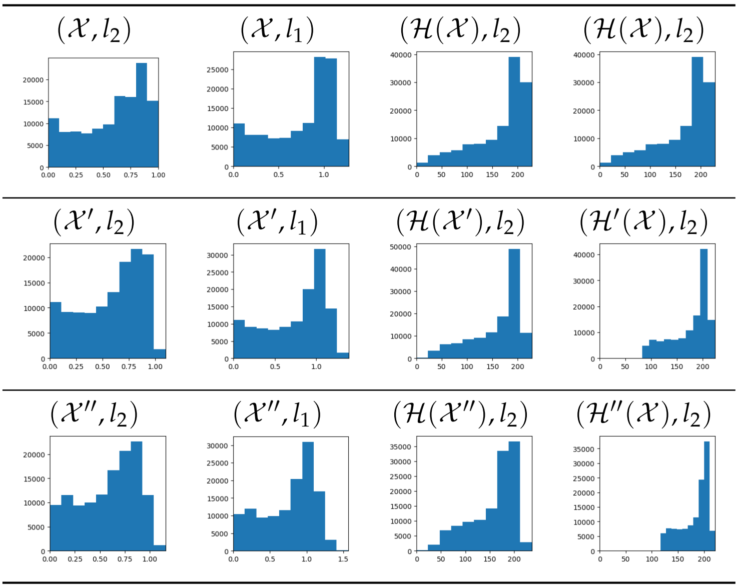


Figure 4.6.: Distribution of distances for the bumpy circle.

NORM DISTRIBUTION We also plot the distribution of norms. Figure 4.8, and the plots for the remaining data in Appendix C, show that, while the norms of the input data samples for the bumpy circle take different values, the norms of the HD vectors concentrate around the same value. Indeed, we show in Section 4.2.3 that $\|H\|_2 \approx \sqrt{ND}$. This means that all points in the HD space lie approximately on a sphere, i.e., in the space between two D-dimensional spheres. The thickness of this space, i.e., of this annulus, is determined by the quality of approximation (F) that ensures the near-otrhogonality of seed ID vectors F, which is influenced by the dimension D (Figure 4.1). Note, however, that this says nothing about the preservation of structure or separability, as the HD data samples may lie on arbitrary regions of the sphere.

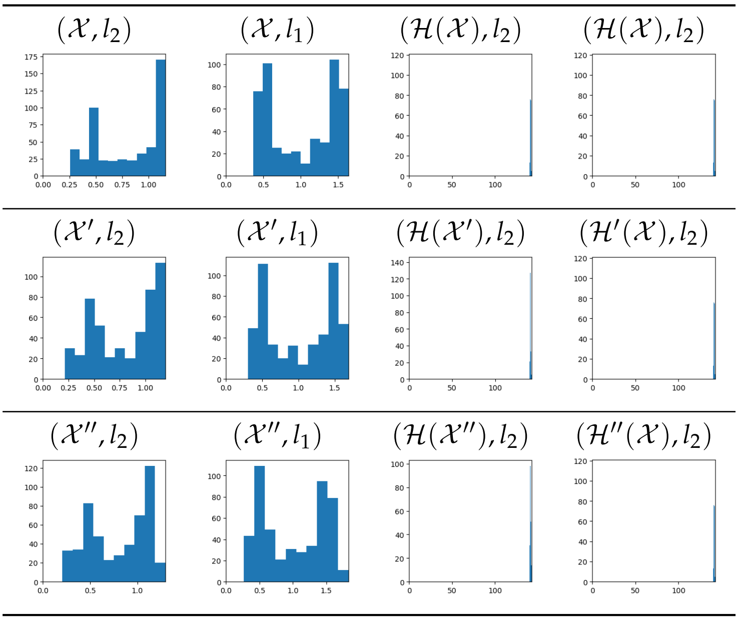


Figure 4.7.: Norm distribution for the bumpy circle.

4.5 Structure preservation

To evaluate the structure (preservation), we consider the 0- and 1-dimensional PH (Chapter 1) that respectively reflects the connected components (or clusters) and loops.

We start by visualizing the 0- and 1-dimensional persistence diagram (PD) and the Betti curves (with filtration parameter and the number of persistence intervals respectively on the x and y axis) for each of the data sets. Figure 4.9 clearly shows that the bumpy circle sees one persistent connected component, and 499 other connected components that live shortly in the filtration, reflecting each of the data samples before they connect to the neighboring points. In the HD space, there is significantly less connected components (which is the case for many other data sets, see detailed results in Appendix C), since some data samples can overlap, as the intervals of values are represented with their centroids (Section 4.2.2). This is, however, not the case for real-world data, as these are higher-dimensional (unlike synthetic data that mostly lie in $\mathbb{R}^2$) and therefore less likely to overlap. The components die relatively much later in the HD space than in the input space (this being more pronounced for corrupted HD data), what is a direct consequence of the relatively larger distances (Section 4.4): two data samples in $H, H' \in \mathcal{H}$ are relatively far away from each other, and it thus takes more time for them to merge into a single connected component within a filtration. 1-dimensional PH shows that each representation of the non-noisy bumpy circle clearly sees the one persistent loop, and four loops that live shortly, reflecting the four bumps. In the presence of noise, additional loops are revelead at times, but they have very short persistence. In general, the loops are born later in the HD space, which is again the consequence of the relatively larger distances. The other data sets exhibit similar behavior, see Appendix C.

To quantify the preservation of structure, in Table 4.3 and Table 4.4, we show the Wasserstein W_2 distance respectively to the 0- and 1-dimensional PD calculated on $(\mathcal{X}, l_2)$. To make these distances meaningful, the birth and death values in each PD is normalized with respect to the maximum distance: $\sqrt{n}$, n and $\sqrt{2nD}$ for $(\mathcal{X}, l_2)$, $(\mathcal{X}, l_1)$, $(\mathcal{H}, l_2)$[8]. For 1-dimensional PH, these distances remain small (in comparison to the distance for the PD of the input data in presence of Gaussian noise), indicating that HDC encoding approximately preserves the loop structure. The distances are much higher for 0-dimensional PH, and our detailed experimental results suggest that this is primarily a consequence of the relatively larger distances in the HD space, that increase the death value of the majority of connected components.

Data	$(\mathcal{X}, l_2)$	$(\mathcal{X}', l_2)$	$(\mathcal{X}'', l_2)$	$(\mathcal{X}, l_1)$	$(\mathcal{X}', l_1)$	$(\mathcal{X}'', l_1)$	$(\mathcal{H}(\mathcal{X}), l_2)$	$(\mathcal{H}(\mathcal{X}'), l_2)$	$(\mathcal{H}(\mathcal{X}''), l_2)$	$(\mathcal{H}'(\mathcal{X}), l_2)$	$(\mathcal{H}''(\mathcal{X}), l_2)$
Bumpy circle	0.00	0.11	0.19	0.01	0.09	0.16	1.00	1.44	1.72	4.91	6.70
Astroid	0.00	0.11	0.18	0.01	0.09	0.15	0.99	1.42	1.65	4.91	6.70
Apple	0.00	0.13	0.19	0.02	0.11	0.16	1.06	1.54	1.72	4.92	6.70
Lemniscate	0.00	0.14	0.23	0.02	0.11	0.19	1.21	1.68	1.92	4.96	6.72
Concentric circles	0.00	0.15	0.26	0.08	0.16	0.23	1.14	1.58	1.78	4.93	6.72
Fish	0.00	0.13	0.20	0.03	0.10	0.16	1.22	1.67	1.83	4.96	6.74
Star	0.00	0.12	0.19	0.02	0.10	0.16	1.24	1.65	1.81	4.96	6.74
Olympics	0.00	0.12	0.19	0.04	0.09	0.14	1.59	1.84	1.96	5.07	6.81
Orthogonal circles	0.00	0.27	0.45	0.24	0.35	0.49	1.33	2.13	2.50	4.95	6.73
Nine squares	0.00	0.24	0.38	0.18	0.31	0.35	1.33	1.76	1.85	4.99	6.77
Cube vertices	0.00	0.15	0.33	0.42	0.46	0.54	2.01	2.07	2.22	5.15	6.84
UCI-HAR	0.00	0.14	0.49	0.95	0.75	0.38	3.90	4.29	4.67	6.08	7.44
CTG	0.00	0.18	0.52	0.69	0.53	0.37	3.35	3.97	4.33	5.83	7.32
ISOLET	0.00	0.07	0.28	1.25	1.13	0.90	6.05	6.01	6.04	7.38	8.35

Table 4.3.: Connectivity preservation: Wasserstein W_2 distance to the 0-dimensional persistence diagram of $(\mathcal{X}, l_2)$.

[8] Recall that the input data $\mathcal{X} \subset \mathbb{R}^n$ is normalized to $[0, 1]$, and that the coordinates of hyperdimensional vectors in $\mathcal{H} \subset \mathbb{R}^D$ take integer values in $[-n, n]$.

Data	$(\mathcal{X},l_2)$	$(\mathcal{X}',l_2)$	$(\mathcal{X}'',l_2)$	$(\mathcal{X},l_1)$	$(\mathcal{X}',l_1)$	$(\mathcal{X}'',l_1)$	$(\mathcal{H}(\mathcal{X}),l_2)$	$(\mathcal{H}(\mathcal{X}'),l_2)$	$(\mathcal{H}(\mathcal{X}''),l_2)$	$(\mathcal{H}'(\mathcal{X}),l_2)$	$(\mathcal{H}''(\mathcal{X}),l_2)$
Bumpy circle	0.00	0.05	0.13	0.09	0.13	0.18	0.43	0.35	0.31	0.35	0.31
Astroid	0.00	0.07	0.17	0.01	0.07	0.18	0.34	0.31	0.28	0.25	0.20
Apple	0.00	0.08	0.13	0.08	0.12	0.18	0.42	0.36	0.30	0.32	0.28
Lemniscate	0.00	0.06	0.14	0.08	0.11	0.19	0.37	0.35	0.33	0.28	0.24
Concentric circles	0.00	0.07	0.11	0.06	0.10	0.12	0.31	0.26	0.26	0.21	0.17
Fish	0.00	0.07	0.13	0.11	0.15	0.18	0.38	0.38	0.32	0.34	0.30
Star	0.00	0.07	0.12	0.03	0.10	0.12	0.28	0.25	0.24	0.19	0.15
Olympics	0.00	0.09	0.17	0.08	0.12	0.18	0.47	0.38	0.33	0.30	0.24
Orthogonal circles	0.00	0.08	0.16	0.18	0.18	0.20	0.37	0.31	0.26	0.24	0.21
Nine squares	0.00	0.16	0.17	0.14	0.21	0.18	0.41	0.30	0.27	0.26	0.21
Cube vertices	0.00	0.09	0.17	0.32	0.31	0.29	0.41	0.38	0.38	0.30	0.27
UCI-HAR	0.00	0.03	0.04	0.04	0.04	0.03	0.09	0.08	0.07	0.06	0.06
CTG	0.00	0.05	0.06	0.10	0.10	0.09	0.24	0.23	0.21	0.16	0.13
ISOLET	0.00	0.03	0.05	0.10	0.09	0.09	0.15	0.14	0.13	0.13	0.11

Table 4.4.: Loop preservation: Wasserstein W_2 distance to the 1-dimensional persistence diagram of $(\mathcal{X}, l_2)$.

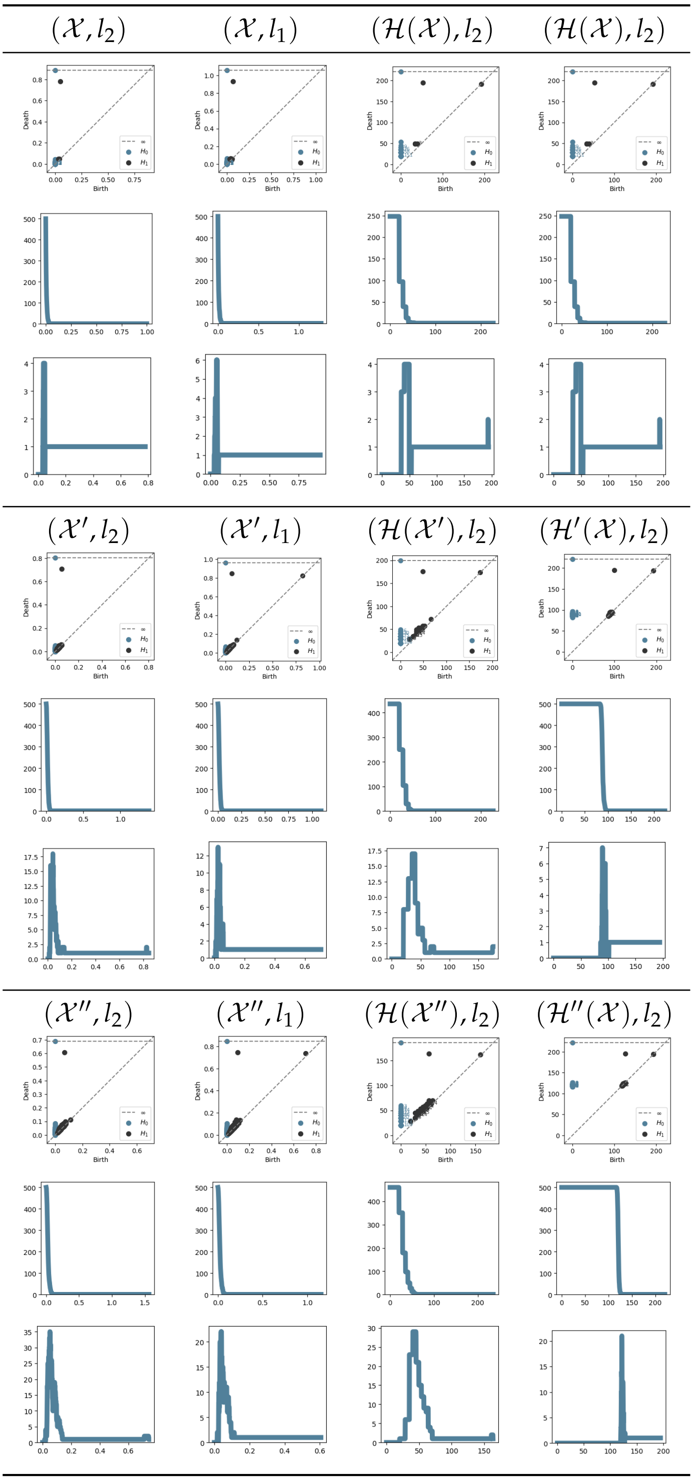


Figure 4.8.: 0- and 1-dimensional persistent homology for the bumpy circle.

4.6 Data separability

Even though an encoding preserves distances, it is of interest to analyze to what extent it preserves, or better, improves the separability of data into classes, simplifying the classification problem. For instance, the kernel trick implicitly maps the input data into a higher-dimensional feature space where the data may be more separable: e.g., with $\phi : \mathbb{R}^2 \to \mathbb{R}^3 : (x_1, x_2) \to (x_1, x_2, x_1^2 + x_2^2)$, the inside and outside of a circle are encoded into the bottom and top of a paraboloid that can be separated with a plane, so that the input data that cannot be separated linearly can be studied with a linearly separable representation. This enables using linear classifiers to solve nonlinear problems.

Here we thus assume that the data is labelled (in the experimental results, the labels are indicated by color in the MDS plots). A common measure of data separability is the Dunn index, ratio between the minimum inter-class and the maximum intra-class distance. Higher Dunn index thus indicates better clustering: data classes are sufficiently far apart, in comparison to the within-class distances. Note that this is different than the 0-dimensional PH that we study in Section 4.5, that reveals the intrinsic connected components or clusters in the data, not considering the labels. For example, 0-dimensional PH for the bumpy circle captures the single connected component, whereas the defined labels split the data into two classes.

Table 4.5 shows that the HDC encoding does not change the Dunn index in a consistent manner across data sets: for some data, the Dunn index in the HD space is comparable to the Dunn index in the input space, but it often significantly increaes or decreases. If the data classes "touch each other" (they are approximately as close as two data samples in the same class), e.g., in the bumpy circle, the Dunn index is close to 1. If the data classes are far from each other, as is the case for the orthogonal circles or cube vertices, it is to be expected that the Dunn index decreases with HDC encoding, since large distances are relatively larger than small distances in the HD space in comparison to the input space (Figure 4.7). For some of the data sets, Gaussian noise somwehat decreases the Dunn index for $(\mathcal{X}, l_2)$, $(\mathcal{X}, l_1)$ and/or $(\mathcal{H}, l_2)$, and the effect is stronger when the noise is added to HD data by corrupting some random coordinates.

Linear separability is of particular interest. Under some conditions, distance preservation is sufficient to preserve linear separability [333, Theorem 22]: if the input data can be separated with a linear boundary, then the same holds for the HD encoding. However, it remains an interesting question whether a linear separator on the HD representation can capture a nonlinear decision boundary on the original data. To address this question, we calculate the classification accuracy of logistic regression on the different data representations (Table 4.6). For every data set, the classification accuracy is at least as good on the HD encoding compared to the input data, and it often improves greatly. It is particulary striking that some of the highly non-linearly separable data become (approximately) linearly separable in the HD space: for

Data	$(\mathcal{X},l_2)$	$(\mathcal{X}',l_2)$	$(\mathcal{X}'',l_2)$	$(\mathcal{X},l_1)$	$(\mathcal{X}',l_1)$	$(\mathcal{X}'',l_1)$	$(\mathcal{H}(\mathcal{X}),l_2)$	$(\mathcal{H}(\mathcal{X}'),l_2)$	$(\mathcal{H}(\mathcal{X}''),l_2)$	$(\mathcal{H}'(\mathcal{X}),l_2)$	$(\mathcal{H}''(\mathcal{X}),l_2)$
Bumpy circle	1.01	1.00	0.99	1.05	1.03	1.01	1.04	1.03	1.03	1.02	1.02
Astroid	0.88	0.88	0.88	0.90	0.90	0.88	0.99	0.98	0.96	0.98	0.98
Apple	0.98	0.98	0.97	1.03	1.03	1.01	1.05	1.05	1.03	1.02	1.02
Lemniscate	1.65	1.62	1.61	1.75	1.72	1.71	1.37	1.35	1.34	1.25	1.18
Concentric circles	0.80	0.80	0.80	0.80	0.80	0.80	0.93	0.93	0.92	0.94	0.95
Fish	0.87	0.86	0.86	0.90	0.89	0.89	0.98	0.97	0.96	0.97	0.98
Star	1.16	1.15	1.13	1.09	1.09	1.07	1.10	1.10	1.08	1.06	1.04
Olympics	1.26	1.24	1.19	1.28	1.25	1.20	1.15	1.13	1.10	1.09	1.06
Orthogonal circles	3.98	3.66	3.46	4.79	3.98	3.52	2.27	2.03	1.92	1.61	1.36
Nine squares	0.94	0.94	0.93	0.93	0.93	0.92	0.99	0.98	0.98	0.98	0.98
Cube vertices	7.17	6.36	4.72	5.32	4.73	3.70	2.36	2.23	1.97	1.57	1.32
UCI-HAR	0.69	0.71	0.74	0.47	0.56	0.65	0.69	0.76	0.82	0.84	0.91
CTG	0.84	0.84	0.85	0.82	0.83	0.84	0.92	0.92	0.92	0.94	0.96
ISOLET	0.78	0.79	0.80	0.77	0.78	0.80	0.88	0.89	0.90	0.92	0.94

Table 4.5.: Data separability: Dunn index.

instance, the average accuracy of logistic regression on concentric circles, fish, and apple increases from only 0.61 in $(\mathcal{X},l_2)$ and $(\mathcal{X},l_1)$ to 0.99 in $(\mathcal{H},l_2)$. However, whereas the Gaussian noise typically does not change the classification accuracy on $\mathcal{X}$, the HD encoding of the noisy data often results in a decreased accuracy. The deterioration of accuracy is commonly less pronounced for the noise on the HD data.

Data	$(\mathcal{X},l_2)$	$(\mathcal{X}',l_2)$	$(\mathcal{X}'',l_2)$	$(\mathcal{X},l_1)$	$(\mathcal{X}',l_1)$	$(\mathcal{X}'',l_1)$	$(\mathcal{H}(\mathcal{X}),l_2)$	$(\mathcal{H}(\mathcal{X}'),l_2)$	$(\mathcal{H}(\mathcal{X}''),l_2)$	$(\mathcal{H}'(\mathcal{X}),l_2)$	$(\mathcal{H}''(\mathcal{X}),l_2)$
Bumpy circle	0.59	0.59	0.59	0.59	0.59	0.59	1.00	0.94	0.85	0.99	0.99
Astroid	0.73	0.73	0.73	0.73	0.73	0.73	0.99	0.92	0.89	0.99	0.99
Apple	0.53	0.55	0.55	0.53	0.55	0.55	0.99	0.99	0.93	1.00	0.99
Lemniscate	0.99	0.97	0.95	1.00	0.96	0.95	1.00	0.96	0.96	0.99	0.99
Concentric circles	0.62	0.59	0.67	0.62	0.59	0.64	0.99	0.98	0.99	0.89	0.86
Fish	0.69	0.69	0.69	0.69	0.69	0.69	0.99	0.97	0.97	0.98	0.97
Star	0.83	0.83	0.81	0.83	0.83	0.81	0.91	0.84	0.82	0.90	0.90
Olympics	0.47	0.45	0.42	0.47	0.43	0.42	0.82	0.67	0.59	0.78	0.74
Orthogonal circles	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Nine squares	0.53	0.53	0.53	0.53	0.53	0.53	0.71	0.69	0.73	0.66	0.67
Cube vertices	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
UCI-HAR	0.93	0.94	0.94	0.93	0.92	0.91	0.97	0.97	0.94	0.95	0.93
CTG	0.87	0.87	0.85	0.88	0.88	0.86	0.91	0.91	0.85	0.89	0.88
ISOLET	0.83	0.84	0.83	0.85	0.85	0.83	0.85	0.81	0.82	0.79	0.77

Table 4.6.: Data separability: Logistic regression accuracy.

4.7 Conclusions

The goal of this chapter was to investigate to what extent are the properties of the input data preserved with the HDC encoding into a 10 000-dimensional space. To this end, we focus on the common ID-Position HDC encoding of Euclidean data. Earlier work provides a theoretical guarantee of an approximate distance preservation for this encoding [333]. We start this chapter with a less rigorous but more intuitive proof of distance preservation, highlighting the influence of the HD dimension D and alphabet size K on the quality of the approximation (Section 4.2). We carry out experiments on a number of synthetic data sets with interesting structure: bumpy circle, astroid, apple, lemniscate, concentric circles, fish, star, olympics, orthogonal circles, nine squares and cube vertices, and HDC benchmark data UCI-HAR, CTG, and ISOLET, under Gaussian noise on the input data, and HD noise on the encoded data, that corrupts a given number of random coordinates.

MAIN FINDINGS The experimental results provide a number of insights.

- Distances: We show empirically that the l_2 metric on the HD data $\mathcal{H}$ correlates well with both l_2 and l_1 metrics on the input data $\mathcal{X}$, with the Spearmann correlation in the latter case always being equal to 1, as the relationship between the metrics is monotonous. The plots of distance matrices provide more intuition about the preservation of distances, suggesting further the preservation of self-intersections and singularities. We also see that the HD data approximately lies on the D-dimensional sphere with radius $\sqrt{ND}$, relatively further away from each other in comparison to the input data samples. Guassian and HD noise do not change these behaviors.

- Connectivity and loops: 0- and 1-dimensional PH of the input and HD data reveal that the HDC encoding approximately preserves the number of connected components and loops, with Gaussian and HD noise making these cycles less prominent. For some data sets, the HD data contains a few additional or less short persistence intervals, in comparison with $(\mathcal{X}, l_2)$ but also $(\mathcal{X}, l_1)$, indicating that some local geometry information is lost.

- Separability: For each of the data sets, the logistic regression accuracy on the HD data is greater than or equal to the accuracy on the input data. Some of the highly non-linearly separable data are linearly separable in the HD space. Gaussian noise on the input data does not significantly change the accuracy, but both the Gaussian noise and the HD noise on the HD data often results in a lower accuracy.

The data and code are publicly available at `https://renata-turkes.github.io/`, including two Jupyter notebooks that can be used to replicate the computational experiments on the synthetic and real-world data, but also to investigate the behavior of HDC as in Section 4.2. The code can also be slightly modified to analyze other HD encodings and data sets.

FUTURE WORK This work can be extended to guide the choice of HDC parameters (definition of encoding, including the bind and bundle operations, dimension D and alphabet size K) that best preserve the structure, by minimizing the Wasserstein distance between PH on the input and HD data. Furthermore, next to the Dunn index and logistic regression accuracy, we can also study the separability in more detail by looking into the PH of the decision boundaries, similarly to [346]. It would be interesting to quantify and compare structure preservation and linear separability across different encodings, and data sets.

SUSTAINABILITY HDC is low power, and offers an energy efficient alternative to conventional realizations of general purpose ML algorithms like support vector machines, multilayer perceptrons, and nearest-neighbor classifiers [333].

5

THE WHEN:
Who is WithMe? EEG features for attention in a visual task, with auditory and rhythmic support

The study of attention has been pivotal in advancing our comprehension of cognition. The goal of this study is to investigate which EEG data representations or features are most closely linked to attention, and to what extent they can handle the cross-subject variability. We explore the features obtained from the univariate time series from a single EEG channel, such as time domain features and recurrence plots, as well as representations obtained directly from the multivariate time series, such as global field power or functional brain networks. To address the cross-subject variability in EEG data, we also investigate persistent homology features that are robust to different types of noise. The performance of the different EEG representations is evaluated with the Support Vector Machine (SVM) accuracy on the WithMe data derived from a modified digit span experiment, and is benchmarked against baseline EEG-specific models, including a deep learning architecture known for effectively learning task-specific features.

This chapter is based on the following article:

> Renata Turkeš, Steven Mortier, Jorg De Winne, Steven Latré and Tim Verdonck, *Who is WithMe? EEG features for attention in a visual task, with auditory and rhythmic support*, under review.

This research was funded by the Research Foundation-Flanders (FWO) under Grant No. G0A0220N.

5.1 Introduction

Understanding the human processing of multi-sensory stimuli in relation to attention has been of great interest in the last decades [92]. Indeed, detecting cognitive states and skills can help improve adaptive learning, in which the learning material and pace are adjusted to match some collected data about learners during a learning task [237]. Moreover, identifying biomarkers that can be used to monitor attention, pleasure and reward, and understanding the relationship between these biomarkers and fine-tuning of stimuli (sound, image, rhythm) can enhance the interaction between humans and artificial intelligence (AI) agents, which is still lacking the degree of engagement and entrainment that characterizes interaction between humans. Such advances in human-centered AI approach open a wealth of applications in public security, health, revalidation, communication and information sharing, entertainment, etc. Some examples include driver fatigue detection [352], rhythmic auditory stimulation to help Parkinson patients improve their gait characteristics and reduce the risk for falling [236], or music systems for synchronization [235] and gait retraining (to prevent running-related injuries) [342] which could be improved by selecting the best rhythmic or music stimulus at the right moment.

A promising methodology for the automated collection of data during a mental task includes the use of bio-sensors that could measure subjects' emotions, attention, and engagement in a non-invasive and non-intrusive way [237]. In this work, we focus on capturing human attention from electroencephalography (EEG) biosignals. EEG data measures oscillatory electrical brain activity at the macroscopic scale with high time resolution [324, 371]. EEG has been shown to have a strong potential to provide biomarkers for diagnoses in many neuropsychiatric disorders [316, 371], including attention deficit hyperactivity disorder (ADHD) [167, 185, 215, 217, 219], but also as indicators of attention during different visual and cognitive tasks [1, 50, 152, 169, 190, 214, 246, 289, 308].

It is commonly understood that a crucial step in EEG processing is to extract relevant features for the considered application [367]. Moreover, the study of EEG data, similarly to other neural data, is further complicated by the high degree of cross-subject variability (due to differences in how the information is represented in the brain, e.g., in terms of the representation of stimulus and activity in the brain), and presence of noise (due to changes in machine calibration, spurious participant movements, and environmental conditions) [297].

The goal of this work is to investigate which type of representations or features of EEG data are the most associated with human attention. We will consider a number of different representations of EEG data (Section 5.2.2, Appendix 5.5.2), including both the features obtained from the univariate time series from a single EEG channel (and then concatenated across channels), such as time domain features and recurrence plots, as well as representations obtained directly from the multivariate time series, such as global field power or functional brain networks.

These two groups of methods are related to the two key principles that help in understanding brain-behavior relationships: segregation, which assumes that the cerebral cortex can be divided into distinct modules, each with its unique structure and functionality; and integration, which assumes that no brain region functions in isolation but rather requires interactions and information exchange between different regions [237]. Since we aim to deal with the issue of person-to-person variability in the EEG data, we found it particularly interesting to consider some topology-based features. Indeed, persistent homology (PH), the main tool of topological data analysis (TDA) (Chapter 1) can be made invariant under different type of transformations (such as translation, rescaling, stretching, or even non-affine deformations), and the stability theorems (Section 1.7) imply that the same is true for robustness to noise. PH has been widely applied in neuroscience: we provide a review of relevant literature in the background on TDA (Appendix 5.5.1), and in the description of the PH-based pipelines on univariate (Section 5.2.2.4) and multivariate (Section 5.2.2.8) time series.

We compare the performance of the different EEG representations on the WithMe data (Section 5.2.1), obtained from a modified digit span experiment. The performance is evaluated as the Support Vector Machine (SVM) accuracy on the features (Section 5.2.4). As a baseline, we also include benchmark EEG-specific models which are shown to work well for the WithMe data [245] (Appendix D), including a deep learning architecture that learns the best features for the task at hand from the EEG multivariate time series. In order to investigate the cross-subject variability, we consider three different scenarios: the accuracy is evaluated on the model trained on the same participant, on seen or on new participants. The results are summarized in Section 5.3, and in Section 5.4 we position them relative to the literature, and discuss the main take-aways and resulting directions for future work.

5.2 Materials & Methods

5.2.1 *WithMe EEG data acquisition*

The experiment includes 42 participants EP01-EP42 (with mean age of 23.71 ± 2.69 years, with no visual or hearing difficulties), who have 64 electrodes positioned on their scalp according to the EEG 10/20 system (Figure 5.1, left panel). Each participant is shown 30 sequences of 10 stimuli on a computer screen: 5 `Targets` (black digit in a circle), and 5 `Distractors` (dark gray digit in a circle, or an empty circle), with each digit presented with equal probability ((for an example sequence, see Table 5.1)). Each of the sequences is shown under 4 different conditions `C1`, `C2`, `C3` or `C4`, in a pseudo-randomized manner (and with no special mention about them made to the participants), that differ with respect to presence of audio and/or rhythm (Figure 5.1, right panel). We note that, for conditions `C2` and `C4` where there is rhythm, each participant is first shown 5 induction stimuli in each sequence to induce the rhythm, but these are ignored in our analysis. Therefore, every participant sees in total $30 \times 10 \times 4 = 1\,200$ `Target` or `Distractor` stimuli. Every stimulus is visible for 200ms, and the interstimulus interval is on average 1.25s. The task is to remember the `Targets`, and the participants need to give their oral responses after the complete sequence is presented. The participants do not know that there are always 5 `Targets`, as they are told they will see "5-7 black numbers".

The WithMe experiment is a novel working memory paradigm that is somewhat inspired by the digit span, oddball and pip-and-pop tasks. The magical number 7 ± 2 is the average digit spanThe average digit span refers to the number of digits that a person can repeat back in correct order immediately after a single auditory presentation. of healthy adults [233], so that a sequence of five digits should be fairly easy to remember for a young adult. The digit span memory task is modified by including `Distractors` between `Target` stimuli. In the oddball attention task, a series of repetitive standard stimuli are infrequently interrupted with a rare oddball stimulus that the participant is instructed to focus on. In the WithMe experiment, however, there is an equal number of `Target` and `Distractor` stimuli, so there are no oddballs. The pip-and-pop attention paradigm makes the `Targets` immediately noticeable among the surrounding items (they "pop out"), but the difference between the WithMe `Target` and `Distractor` stimuli is very subtle. In summary, the WithMe experimental design complicates the three paradigms, in order to allow to observe the effect of added support on attention more easily. Further details can be found in the first paper that introduces the WithMe experiment, and discusses the behavioral analysis of participant performance [92].

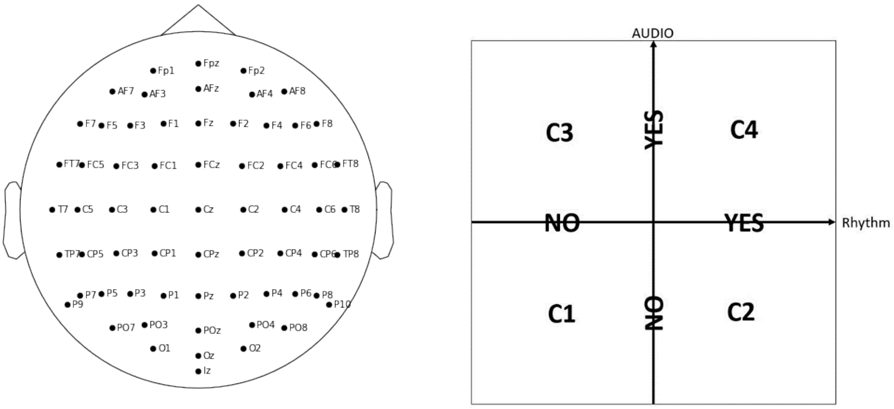


Figure 5.1.: In the WithMe experiment, 64 EEG electrodes are considered, according to the 10/20 system (left plot). Every sequence of numbers is shown to each participant under four different conditions, `C1`, `C2`, `C3` or `C4`, which indicate the presence of auditory and/or rhythmic support (right plot).

The data is pre-processed according to standard techniques: the amplitude values are referenced to the average of the both earlobes, bad channels are detected and then interpolated with 3 neighbouring electrodes, notch filter is applied at 50Hz, and bandpass filter between 0.2 and 100Hz. The data is then epoched from -0.2s to 1s, and a visual inspection of ICA components is performed to remove artefacts. Finally, we downsample the time series with the subsampling period of 50, resulting in 60 time steps, so that each time step corresponds to 1200ms/59=20.34ms. In most situations, downsampling the results to 40Hz or 50Hz (thus, one time point every 25 or 20ms) maintains the advantages of downsampling with minimal loss of information [79]. Finally, the EEG amplitude values are cut off within range $[-50\mu V, 50\mu V]$. The WithMe dataset can then be seen as a $42 \times 1\,200 \times 64 \times 60$ matrix:

- 60 time steps (within $1.2s$),
- 64 EEG channels,
- 1 200 epochs, i.e., EEG multivariate time series reflecting a single `Target` or `Distractor` stimulus, across channels,
- 42 participants.

Some examples of the WithMe EEG multivariate time series across 64 electrodes, for a single participant and a single stimulus, are visualized in Figure 5.5 and Figure 5.8.

5.2.2 *Multivariate time series analysis*

In this section, we describe in detail the different approaches to multivariate time series analysis that we will evaluate in the computational experiments. These methods rely on the different types of features, or representations of multivariate time series, that belong to two different groups. Firstly, one can consider the individual univariate time series (for each EEG electrode), and concatenate the information extracted from each of them separately (Sections 5.2.2.1-5.2.2.4). Alternatively, we can focus on the relationship between the univariate time series (i.e., relationship between different brain regions), which we summarize in a few different ways (Section 5.2.2.5-5.2.2.10). The latter approaches include the baseline `xDAWN-RG` model [84], the IEEE Neural Engineering Conference 2015 Brain Computer Interface (BCI) challenge winner, and `EEGNet` [199], a benchmark deep learning architecture for EEG signal processing and classification which learns the best representation for the given task, that have both been shown to perform well on the WithMe data [245] (Appendix D). A visual summary of all of the different approaches is given in Figure 5.2.[1]

[1] Figure 5.2 focuses on an example WithMe data point, but the pipelines can be applied to any data set of multivariate time series.

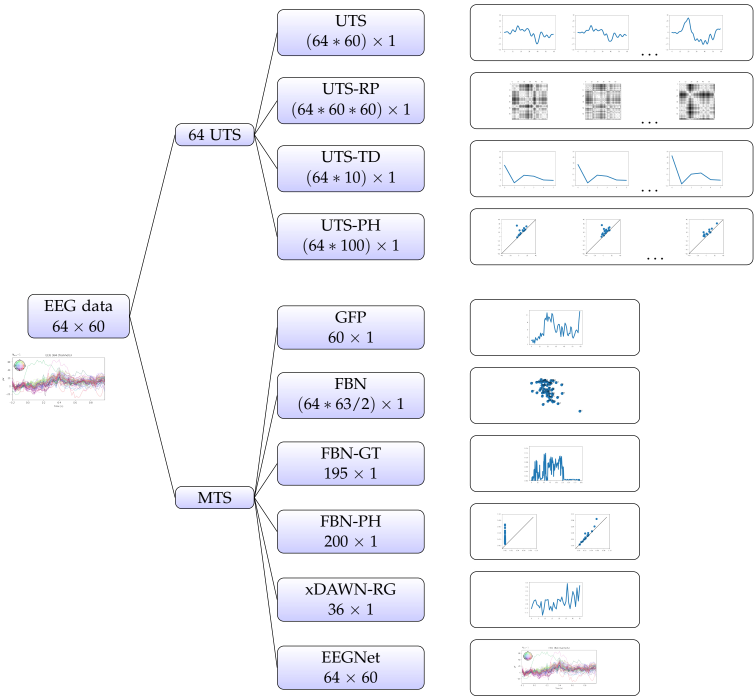


Figure 5.2.: Multivariate time series analysis. A WithMe data observation, an EEG multivariate time series across 64 electrodes and 60 time steps, can be represented with different types of features. We can consider the 64 univariate time series (UTS), that each correspond to a single participant, stimulus and electrode, extract features from each of them separately, and concatenate the information (top branches). It is also possible to calculate the features from the multivariate time series (MTS) itself, that rely on the relationship between the 64 channels (bottom branches).

5.2.2.1 *Univariate time series (UTS)*

Probably the most straightforward way to transform a multivariate time series into a vector is to concatenate the univariate time series across all features. For instance, a WithMe multivariate time series across 64 electrodes and 60 time steps (matrix of shape 64×60) can be represented as a vector of length $3\,840 = 64 * 60$.

5.2.2.2 *Recurrence plots of univariate time series (UTS-RP)*

A recurrence is a time the trajectory returns to a location it has visited before. For a univariate time series, the recurrence plot is a matrix of distances in the signal between every pair of points in time. A WithMe multivariate time series across 64 electrodes and 60 time steps (matrix of shape 64×60) can in this way be represented with a vector of length $230\,400 = 64 * 60 * 60$, corresponding to the flattened 60×60 recurrence plots, concatenated across 64 EEG electrodes. Recurrence plots have been employed in EEG analysis, for example for emotion recognition [17], or to differentiate between seizure-free, pre-seizure and seizure states in genetic absence epilepsy rats [260].

5.2.2.3 *Time-domain features of univariate time series (`UTS-TDF`)*

Instead of looking at the whole univariate time series for each channel (`UTS` pipeline, Section 5.2.2.1), we can also only extract some statistics about the time series, such as their maximum, minimum, mean and variance. Next to considering the largest peaks in the time series, we will include the peaks from within certain intervals, as these are related to the so-called event-related potentials (ERPs). An ERP is a stereotyped brain response to a specific sensory, cognitive, or motor stimulus. ERP waveforms consist of a series of positive and negative voltage deflections, which are related to a set of underlying components. Most components are referred to by a letter (N/P) indicating polarity (negative/positive), followed by a number indicating either the latency in milliseconds or the component's ordinal position in the waveform:

- N100 or N1: This is the first substantial peak in the univariate time series. It is a negative-going peak typically occurring about 100 milliseconds after a stimulus is presented, but may exhibit a peak anywhere between 80 ms and 120 ms (280 and 320 ms after the start of our time series, see Section 5.2.1). We therefore calculate the minimum of the waveform within this range (between time steps $t = 13$ and $t = 16$ in our time series).

- P100 or P1: This is a positive extreme occurring about 100 milliseconds after a stimulus is presented, but may exhibit a peak anywhere between 80 ms and 120 ms. We therefore calculate the maximum of the time series within this range (between time steps $t = 13$ and $t = 16$ in our time series).

- P200 or P2: This is the second substantial peak in the time series, which often occurs about 200 milliseconds after the stimulus onset. We calculate it as the maximum of the waveform between 150 and 275 ms, i.e., between time steps $t = 17$ and $t = 24$ in our time series.

- P300 or P3: This is the third substantial positive-going peak in the waveform, occurring about 300 milliseconds after a stimulus is presented. We calculate it as the maximum of the waveform between 250 and 500 ms, i.e., between time steps $t = 22$ and $t = 35$ in our time series.

A review of EEG/ERP applications can be found in [254]. In particular, multiple ERPs were found to be associated with mind-wandering [169], or different stages of attention [1]. Note that the EEG reflects thousands of simultaneously ongoing brain processes, making it challenging to see the brain response to the event of interest in the EEG recording of a single trial [36]. To see the brain's response to a stimulus, the experimenter commonly conducts many trials and averages the results, causing random brain activity to be averaged out and the relevant waveform to remain [79]. However, in this work we focus on applications that aim to predict the attention from the given EEG time series corresponding to some stimulus, and we therefore extract the given metrics for each EEG epoch. We represent each of the univariate time series with 10 time-domain features: maximum, minimum, mean, variance, skewness, kurtosis (similar to [237]) N1, P1, P2 and P3 peaks, and then concatenate the information across EEG electrodes. In this way, a WithMe 64×60 multivariate time series is represented with a vector of length $640 = 64 * 10$.

5.2.2.4 *Persistent homology of univariate time series (UTS-PH)*

The shape of the EEG wave has been shown to contain useful information about the state of the brain [179]. For this reason, we represent EEG time series for each channel with its persistent homology (PH) with respect to the lower-star filtration (Appendix 5.5.1) directly on the signal, i.e., on the function $f : \{t_1, \ldots, t_n\} \to \mathbb{R}$. Persistent homology with respect to such a filtration measures the relative height of the peaks of the EEG signal, but not their width, so that it is invariant to expansion and contraction in the time axis direction (Appendix 5.5.1, Figure 5.10; Appendix 5.5.2, Figure 5.12, row 5, columns 2 and 4). Moreover, addition of noise to the signal results in minor changes of PH (Appendix 5.5.2, Figure 5.12, row 5, columns 2 and 5). This makes PH an interesting candidate for overcoming individual differences across subjects [100].

A similar pipeline is employed for the epileptic seizure, autism and arrhythmia detection from EEG or ECG in [100, 222, 355, 356]. In this work, we represent the WithMe EEG univariate time series with their 0-dimensional 10×10 persistence images (Appendix 5.5.1), so that an EEG epoch for 64 channels and 60 time steps results in a vector of length $6\,400 = 64 \times 10 \times 10$.

Rather than computing PH directly on a univariate time series, one can first employ the so-called sliding window embedding to transform it to a point cloud, and then calculate PH. We do not use this approach here as it is most suitable for distinguishing between the non-chaotic and chaotic time series, since the loops in the embedded point cloud reflect cyclic behavior, see Section 5.5.1 for more details.

5.2.2.5 *Global field power (GFP)*

The representations above (Sections 5.2.2.1-5.2.2.4) focus on the features from the univariate time series. However, the interactions between the different time series (i.e., different brain regions reflected by the EEG channels) might contain (more) meaningful information. One of the simplest ways to summarize this relationship between the time series is via the global field power (GFP). GFP is a measure of the scalp field strength and corresponds to the standard deviation of the signal across electrodes at each time point [321]. Thus, it is a one-dimensional time series capturing the spatial variability of the signal across sensor locations. The WithMe 64×60 multivariate time series is thus represented as a vector of length 60. GFP has found applications in studies of perceptual, attentional, cognitive and drug-related aspects of information processing [230].

5.2.2.6 *Functional brain network (FBN)*

From a multivariate time series, one can construct a graph or network, with vertices or nodes which reflect the different univariate time series, and the edges which describe some relationship between between them. In the context of (WithMe) EEG multivariate time series, the nodes correspond to the different EEG electrodes or brain regions, and the weights of the edges correspond to some measure of connectivity between them, with the resulting graph commonly referred to as a functional brain network [327]. These functional brain networks provide a new understanding of the characteristics of the brain, since different cognitive or perceptual tasks require a coordinated flow of information within networks of functionally specialized brain areas [27].

Indeed, changes in the topology of EEG functional brain networks appear to accompany a series of neurological and psychiatric disorders, such as stroke damage [347], schizophrenia [166, 231], amyotrophic lateral sclerosis (ALS) [125] or Alzheimer's [164, 326, 371], and can therefore be used as diagnostic markers for these conditions [329]. Moreover, several studies have suggested that EEG interregional correlations are associated with conscious cognitive processing and active perception [334, 347].

The correlation between time series (i.e., connectivity between brain regions) can be calculated in many different ways, such as cross-correlation, coherence, and synchronization likelihood. In our computational experiment, we will consider the common Pearson product-moment correlation coefficients [165]. The distance d_{ij} between the two univariate time series i and j is then calculated as $d_{ij} = 1 - p_{ij}$, where p_{ij} is the Pearson correlation. As input for this FBN pipeline, we will consider the distance matrix itself, above the diagonal and flattened into a vector, similarly to [288]. The WithMe 64×60 multivariate time series thus results in a 64×64 distance matrix, that is then flattened into a vector of length $2\,016 = 63 * 64/2$.

5.2.2.7 *Graph theory of functional brain networks (`FBN-GT`)*

Instead of feeding the complete correlation or distance matrix to a machine learning algorithm (`FBN` pipeline, Section 5.2.2.6), it is common to analyze such a matrix (or functional brain network in the neuroscience context) using graph theory. Typically, the weighted graph, i.e. the correlation or distance matrix is thresholded at a prespecified level to produce the binary adjacency matrix that only indicates if a connection between vertices exists [202]. Then, the corresponding graph topology of the binary matrix can be characterized by calculating the graph metrics of interest that characterize the functional integration and segregation.

Indeed, significant differences across some graph theory metrics have been found between EEG brain networks for control subjects and patients of a range of neurological and psychiatric disorders [47, 125] (such as stroke, multiple sclerosis, Parkinson's, epilepsy or depression). Moreover, in the healthy brain, individual variability in cognitive functions, learning a new task, or the predisposition to learn have been correlated with specific patterns of network connectivity [184]. In our computational experiments, we consider the assortativity degree, average path length, edge connectivity, and for each node, its degree, betweenness, and eccentricity. To obtain the graph adjacency matrix, the normalized distance matrix (see `FBN` pipeline in Section 5.2.2.6) is thresholded at 0.1. For a WithMe EEG multivariate time series across 64 channels, i.e., a 64×64 functional brain network (distance matrix), this results in a vector of length $195 = 1 + 1 + 1 + 64 * (1 + 1 + 1)$.

5.2.2.8 *Persistent homology of the functional brain network (`FBN-PH`)*

Studying the correlations using graph theory or network science (`FBN-GT` pipeline, Section 5.2.2.7) suffers from methodological problems. Firstly, finding a proper threshold is one of the crucial issues, since the graph structure drastically changes depending on how to threshold a connectivity matrix. For example, most graph characteristics depend on the number of edges in the graph, and the estimated graph topology is therefore biased by the choice of the threshold. This hampers a meaningful comparison of graph topology between individuals or groups. Some of the proposed thresholding methods, such as the multiple comparisons correction and the sparsity control, assume that the strongly connected edges are only important; however, it is suggested that the weakly connected edges may also have discriminative information between networks [26, 203]. The choice of threshold has a major influence on the resulting graph [141] and inevitably leads to a loss of information. Determining the threshold can be based on the statistical significance by the false discovery rate or by fixing the graph metrics such as number of vertices and edges. However, these methods are fairly ad-hoc and everyone seem to use different thresholding techniques. This arbitrariness is demonstrated in [202, Figure 1], where it is shown that the number of edges, the number of connected components and smallworldness of a brain network substantially change depending on the threshold: by varying the threshold, the topology changed to random-like, small-world and clustered network. In addition, it was shown that the clustering coefficient, modularity, efficiency, efficiency-cost show and assortativity of a brain network change greatly across different thresholds (reflected by the network cost, i.e., the total number of edges) [171, Figures 3-7], and the same was shown to be true for efficiency, clustering coefficient, small-worldness, modularity, vertex and edge betweenness centrality, variance of vertex degrees, assortativity and synchronizability in [165, Figures 5-8]. Ideally, graphs should be characterized across a broad range of thresholds [305].

Moreover, in many real systems, dyadic relationships between pairs of vertices fail to accurately capture the rich nature of the system's organization, e.g., cognitive functions appear to be performed by a distributed set of brain regions and their interactions [137]. Furthermore, another drawback of the common graph theory approaches is that they require 2-dimensional embedding of structures that might otherwise be of higher dimension [29].

Persistent homology (PH) of a graph goes beyond graph-theoretic analysis by describing the architecture of a graph in more flexible ways, that investigates the persistence of relationships between graph vertices across multiple scales [11]. Instead of trying to determine one fixed optimal threshold, persistent homology allows us to look at the topological changes of graphs while increasing the threshold continuously. Persistent homology represents the weighted graph with a finite number of nested binary graphs over every possible threshold. In contrast to standard methods of graph or network analysis, persistent homology also encodes higher order connections and thus allows to go beyond pairwise connections; this is helpful for gaining global understanding of low-dimensional structures in graphs [327]. Indeed, experimental results in [148] show that compared with the existing methods, persistent homology can extract the topological features of brain networks more accurately and improves the accuracy of diagnostic and classification.

The first papers that deal with PH of brain networks [202, 203] demonstrate differences between the local connectivity structures in functional brain networks for attention deficit hyperactivity disorder (ADHD) and autism spectrum disorder (ASD), and later, in a depressed brain [184, 369]. Furthermore, PH metrics have been employed to investigate how the topological architecture of brain networks is related to cognitive function, behavior and personality [11, 213, 370]. In this chapter, we employ PH with respect to the rank filtration (Section 5.5.1) on the functional brain networks. This is useful in neuroscience applications, where correlations cannot be assumed to give a precise definition of the distances between graph nodes, and the PH defined in such a way remains unchanged under nonlinear monotonic transformations of the distances (see Appendix 5.5.1, Figure 5.11). More precisely, we concatenate both the 0- and 1-dimensional 10×10 persistence images (that respectively reflect the connected components or clusters, and loops) for each graph, so that a WithMe 64×64 functional brain network is represented with a vector of length $200 = 10 * 10 + 10 * 10$.

5.2.2.9 *xDAWN-RG (xDAWN-RG)*

As one of the baselines, we consider xDAWN-RG, one of the benchmark techniques for classification of multivariate bio-signals, like EEG, MEG or EMG. xDAWN-RG was the winner of the IEEE Neural Engineering Conference Brain Computer Interface (BCI) challenge [226], whose goal was to detect errors during a spelling task, given subject's EEG data. It consists of applying xDAWN spatial filters [299], calculating covariance matrix between the EEG channels to encode their statistical dependencies [23, 82], selecting the channels via Riemannian Geometry (RG) [18], and projecting the reduced covariance matrices in the tangent space [19, 20]. A WithMe 64×60 multivariate time series is transformed into a 8×8 covariance matrix, which is then projected into a vector of length $36 = (8 + 1) * 8/2$.

5.2.2.10 *EEGNet (EEGNet)*

As the main baseline, we consider EEGNet [199], a benchmark deep learning architecture for EEG signal processing and classification. EEGNet is a convolutional neural network (CNN) that learns the best representation for the given task directly from the multivariate time series, with each data observation corresponding to the matrix of univariate time series across channels. Therefore, the raw WithMe 64×60 multivariate time series is fed directly to the model.

5.2.3 *Attention score*

The focus of this chapter and the WithMe data acquisition (Section 5.2.1) is attention recognition. Remember that each experimental participant is shown 30 sequences of 10 Target or Distractor numbers on the screen, and is instructed to list the Targets in the correct order of appearance. We are thus interested to what degree the different pipelines (Section 5.2.2) can predict how well a participant was able to remember the Targets and ignore the Distractors from the EEG data. To do so, we define a simple scoring function, which reflects the digit recall accuracy, i.e., how well a participant was paying attention to the given stimulus.

Note firstly that the attention performance is not always conclusive, since the Targets and Distractors in a sequence are not necessarily unique: e.g., if a sequence contains digit 5 both as a Target and as a Distractor, and a participant reports 5, we do not know if the participant correctly remembered the target, or did not properly ignore the distractor. We assign such stimuli a value of -1.

We define the scoring function for the remaining, well-behaved stimuli to take values in $[0, 1]$. The score of 0 indicates perfect performance: `Target` stimulus is given in the participant's answer in the correct position, or the `Distractor` stimulus was properly ignored. On the other extreme, the score of 1 means that the participant did not remember the `Target`, or has listed the `Distractor` in their answer. The scores of $0.2, 0.4, 0.6$, or 0.8, that are only possible for the `Targets`, aim to capture that a participant remembered the `Target`, but at the wrong position; the value of the score indicates how wrong is the provided answer. Note, however, that e.g., a score of 0.2 for some `Target` stimulus does not necessarily imply that the subject was not paying "perfect" attention, since it might rather be that they were not attentive during a previous `Target` that they thus did not include in their answer. Since a stimulus might appear multiple times in a sequence, note also that the scoring function does not necessarily always correctly reflect the attention, and a participant might simply "get lucky" and report the number correctly.

Therefore, our simple scoring function is defined as follows:

$$\text{score}(T) = \begin{cases} 0 & T \in \text{answer, at the right position} \\ 0.2 & T \in \text{answer, wrong by one position} \\ 0.4 & T \in \text{answer, wrong by two positions} \\ 0.6 & T \in \text{answer, wrong by three positions} \\ 0.8 & T \in \text{answer, wrong by four positions} \\ 1 & T \notin \text{answer} \\ -1 & T \text{ appears in answer less than in } \texttt{Targets} \end{cases} \tag{5.1}$$

$$\text{score}(D) = \begin{cases} 0 & D \text{ appears in answer less than or same as in } \texttt{Targets} \\ 1 & D \text{ appears in answer more than or same in } \texttt{Targets+Distractors} \\ -1 & D \text{ appears in answer less than in } \texttt{Targets+Distractors}, \\ & \text{or } D \text{ is empty} \end{cases} \tag{5.2}$$

An example of a sequence, a participant's answer and their score for each stimulus is given in Table 5.1. Figure 5.3 shows that the large majority of the stimuli yield the perfect score=0: the `Target` is remembered at the correct position, or the `Distractor` is appropriately ignored. This is as expected since, as we discussed earlier, the literature suggests that a sequence of five digits should be fairly easy to remember for young adults.

Sequence	⑧ ⑥ ⑥ ⑥ ⑨ ○ ⑨ ⑦ ⑤ ④
`Targets`	86974
`Distractors`	66 95
Answer	86694

Table 5.1.: An example of a sequence of stimuli shown to each experimental participant on a computer screen, with the digits appearing one by one, `Targets` in black and `Distractors` in grey. For the given answer, the participant obtains the following scores on the 10 shown `Target` and `Distractor` stimuli: score(T_1) = 0, score(T_2) = 0, score(T_3) = 0.2, score(T_4) = 1, score(T_5) = 0, score(D_1) = -1, score(D_2) = -1, score(D_3) = -1, score(D_4) = 0, score(D_5) = 0.

5.2.4 *Experimental set-up*

The classification is done with a linear SVM on the features obtained in the pipelines, except for `EEGNet` that classifies the multivariate time series directly. According to a recent review [218], due to its good performance, SVM is among the most popular types of classification algorithms for EEG. Moreover, we also want to evaluate to what extent are the different pipelines able to deal with the person-to-person variability, as this is an important challenge of EEG data (Section 5.1). To this end, we consider three different experimental scenarios, that differ in the train and test data used for classification:

- Classification per participant: We start with the simplest scenario, when the model is trained on 70% of data (randomly chosen EEG epochs) for a single participant, and the test data corresponds to the remaining 30% of multivariate time series for that same participant.

- Classification on seen participants: Next, we train the models on 70% of randomly chosen multivariate time series, and test on the remaining data. In this case, the train and test data consists of all (and therefore, the same) participants.

- Classification on new participants: Finally, we train the models on all the EEG data from 70% of the experimental participants, and test on the complete data for the remaining 30% participants. Here, the test data consists of new participants compared to the train data.

For each of the three scenarios, we consider three different splits between train and test data, which are the same across different pipelines. The drop in accuracy from the test data consisting of seen and new participants (the last two scenarios above) can give an idea of how well a pipeline is able to avoid the issue of cross-subject

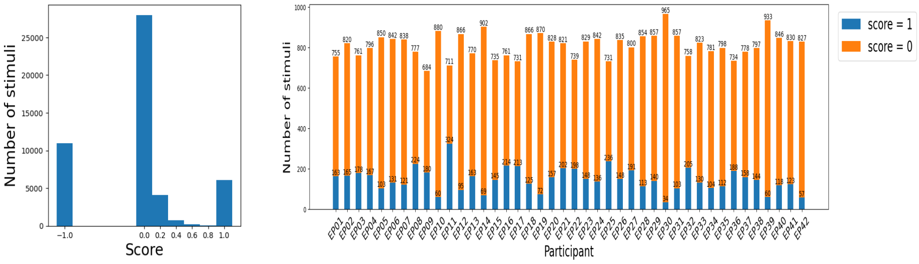


Figure 5.3.: Distribution of scores. The number of inconclusive stimuli with score=-1 is not negligible (left plot), but removing them does not lead to a major data loss, in particular since it allows to focus on the EEG epochs with a meaningful attention score. The number of EEG epochs with score in $\{0.2, 0.4, 0.6, 0.8\}$ is too small for a score prediction task (left plot), so that this data is also removed. In this way, score prediction amounts to classification between score=0 (attentive) and score=1 (inattentive) stimuli or EEG epochs.

variability. Note that the size of data differs greatly between the first, and second and third experimental scenario, since the former is limited to the EEG data from a single participant. The data size for each experiment is explicitly mentioned in Section 5.3.

Finally, since the pipelines `UTS-RP`, `UTS-TD` and `UTS-PH` that extract features from the univariate time series yield a large number of features in comparison to the number of data observations (e.g., 230 400 features for 1 200 data samples in the first experimental scenario where we focus on an individual subject), these models in general tend to overfit. For this reason, we limit these pipelines to the features from 5 most important EEG channels, identified by the `UTS` pipeline: we take these to be the EEG channels with the largest values of the linear SVM coefficients trained on the complete data. Removal of noisy of irrelevant channels in general makes the model less prone to overfitting [240].

For some of the classification problems, the data is very unbalanced: for instance, the number of EEG epochs with score=1 is quite small in comparison with score=0 (Figure 5.3), so that even a random guess yields a very high accuracy. For this reason, we consider a balanced accuracy, which is calculated as the average between the true positive rate or recall $\frac{TP}{TP+FN}$, and the true negative rate $\frac{TN}{TN+FP}$, where TP, TN, FP and FN are respectively the number of true positives, true negatives, false positives and false negatives. If the data is well balanced, the accuracy and balanced accuracy tend to converge to the same value. We opt for an adjusted classification accuracy in order to allow for a fair comparison across participants, which would not be the case if we oversampled some epochs for some participants, simultaneously avoiding data loss due to undersampling.

5.3 Results

5.3.1 *Score prediction*

The goal of this subsection is to evaluate to what extent can EEG signals predict the attention score (Section 5.2.3) that reflects how well a person has remembered the given `Target`, or ignored the given `Distractor` stimulus, shown on the computer screen. We limit the data to EEG multivariate series with score=0 (attentive) and score=1 (inattentive), in order to focus only on the stimuli for which we are fairly confident whether a participant was paying attention. Indeed, as we discuss in Section 5.2.3, the data samples with score=-1 are inconclusive, and score $\in \{0.2, 0.4, 0.6,$ or $0.8\}$ might often be misleading. Moreover, there is only a limited number of stimuli with a score of $0.2, 0.4, 0.6$, or 0.8 (Figure 5.3, left panel) so that any classifier would struggle to learn to recognize such signals. This results in a somewhat different number of appropriate EEG epochs across participants (Figure 5.3, right panel). To make the comparisons across participants fair, we randomly select 684 EEG epochs appropriate epochs for each participant (what is the minimum number of epochs with a conclusive score for participant EP09).

Figure 5.4 shows the balanced accuracy for score=0 vs score=1 classification for the different pipelines (Section 5.2.2), for the three different experimental scenarios, i.e., train and test data (Section 5.2.4). In general, most of the pipelines perform extremely poorly, and are not much better than a random guess with an accuracy of 0.5. It is only `xDAWN-RG` that is able to perform some classification (which does not drop for the more difficult problem when the test data consists of EEG epochs for unseen subjects), but it still only obtains the accuracy of around 0.6.

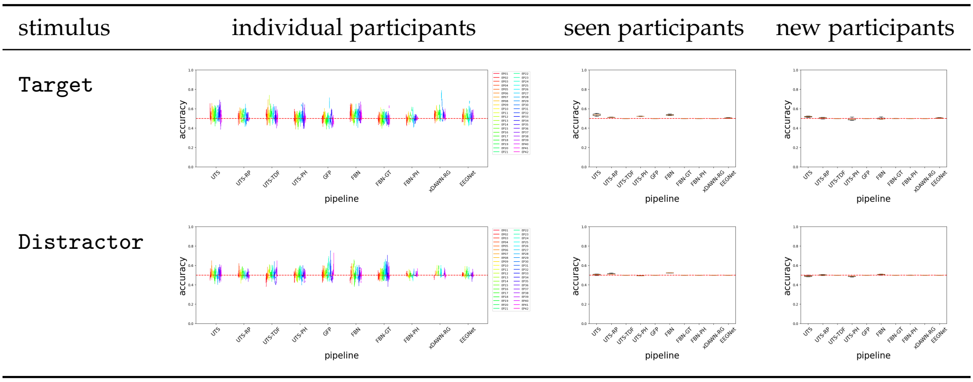


Figure 5.4.: Score 0 (attentive) vs 1 (inattentive) classification accuracy across the pipelines.

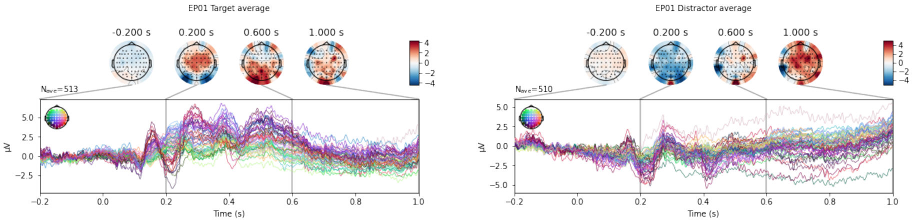


Figure 5.5.: An example of EEG signals across 64 EEG electrodes for participant EP01, averaged across `Target` (left) and `Distractor` (right) stimulus.

5.3.2 *Classification between `Target` and `Distractor` stimuli*

In this section, we perform the classification between `Target` and `Distractor` stimuli, since P300 ERP is expected to be observed when subjects see a `Target` stimulus, and the amplitude of the P300 has shown to be proportional to the amount of attentional resources engaged in processing a given stimulus [143]. Indeed, we show in Figure D.1 that the evoked response for an example WithMe participant exhibits a clear positive voltage deflection around 300ms post-stimulus in the parietal-occipital electrodes. This can also be observed in Figure 5.5, which shows an example of EEG multivariate time series for a participant, for the different type of stimuli. Classification between `Targets` and `Distractors` investigates whether the brain responds differently to the two different stimulus types, which is thus informative of attention, but does not depend on the particular choice of the attention score.

Note that the subjects do not always correctly identify the `Target` or `Distractor` stimulus, so that their EEG signals do not necessarily exhibit a behavior that might be representative of the different type of stimuli. To avoid this issue, we limit the classification only to stimuli that were perfectly remembered or ignored (score=0). Restricting the data in such a way results in somewhat different number of appropriate EEG epochs across participant, with the minimum number of 387epochs.

Figure 5.6 shows that a number of pipelines achieve a good classification accuracy of 75% or more. What is probably the most surprising is that the simplest pipeline `UTS` (Section 5.2.2.1), where the univariate time series across 64 EEG channels are simply concatenated into a large vector, often even outperforms the benchmark `xDAWN-RG` and deep learning `EEGNet` methods. However, this pipeline struggles more to maintain good performance on new participants. Overall, in this case, the pipelines that extract and then concatenate the features from each of the univariate time series separately (`UTS`, `UTS-RP`, `UTS-TDF`, `UTS-PH`) outperform the pipelines that focus on the relationship between the time series across EEG channels (`GFP`, `FBN`, `FBN-GT`, `FBN-PH`).

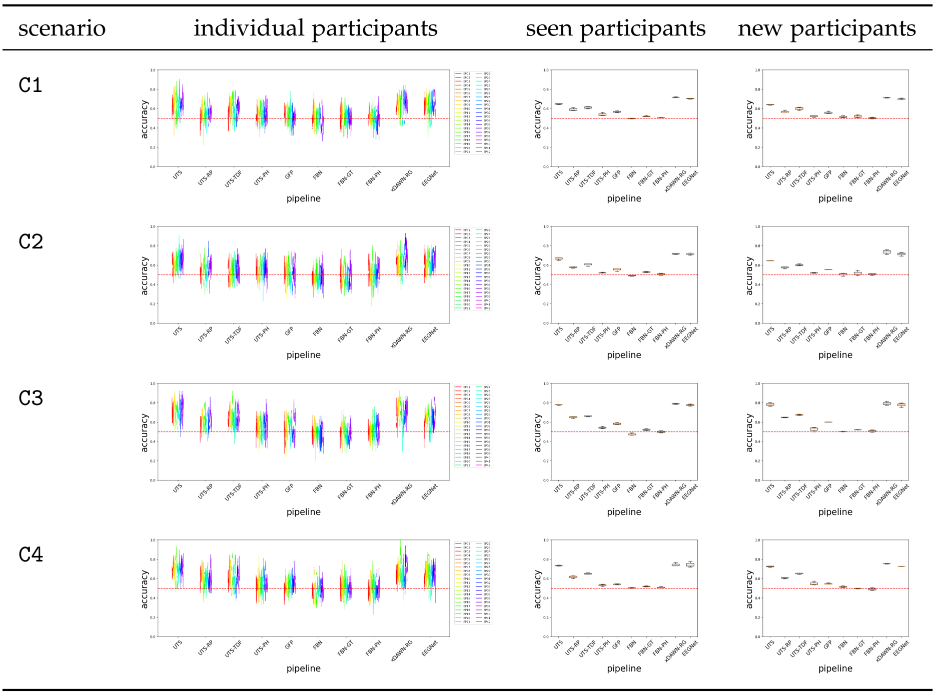


Figure 5.6.: `Target` vs `Distractor` classification accuracy across the pipelines.

Remember that the WithMe experiment is performed under four experimental conditions `C1`-`C4`, that indicate the presence of rhythmic or auditory support (Section 5.2.1). Whenever present, the rhythm and audio accompany the `Targets` only (and not the `Distractors`), so that one might wonder whether the classification above is a detection of rhythm and auditory clues rather than a differentiation between the `Target` and `Distractor` stimuli. To verify this, we could perform the classification separately under each of the conditions. However, we opted to keep the experiments simple and to exploit as much data as possible for learning, since similar WithMe `Target` vs. `Distractor` accuracy has been shown across the different experimental conditions (Table D.5). Moreover, we performed the `Target` vs. `Distractor` classification for condition `C1` (with neither rhythmic nor auditory support, and with only 25% of the original data) with `UTS`, obtaining a balanced accuracy of 0.72, that is not significantly lower than the accuracy of 0.74 when the conditions are considered together.

5.3.3 *Classification between conditions `C1`, `C2`, `C3` and `C4`*

Next to attention recognition, one of the goals of the WithMe experiment (Section 5.2.1) is to investigate to what extent can rhythmic and/or auditory clues (Figure 5.1, right panel) improve attention. Figure 5.7 shows that for the majority of the participants, the presence of rhythm alone (C2) does not improve their attention score during the modified digit span task, although there are some differences across subjects. However, the presence of auditory support (C3) commonly helps to achieve a better score, that is rarely improved further with the additional rhythmic support (audio and rhythm together, C4). This is consistent with earlier findings on the WithMe data [92], for three different scoring functions that look into the performances across complete sequences of digits shown on the screen (rather than for each of the 10 individual stimuli in a sequence).

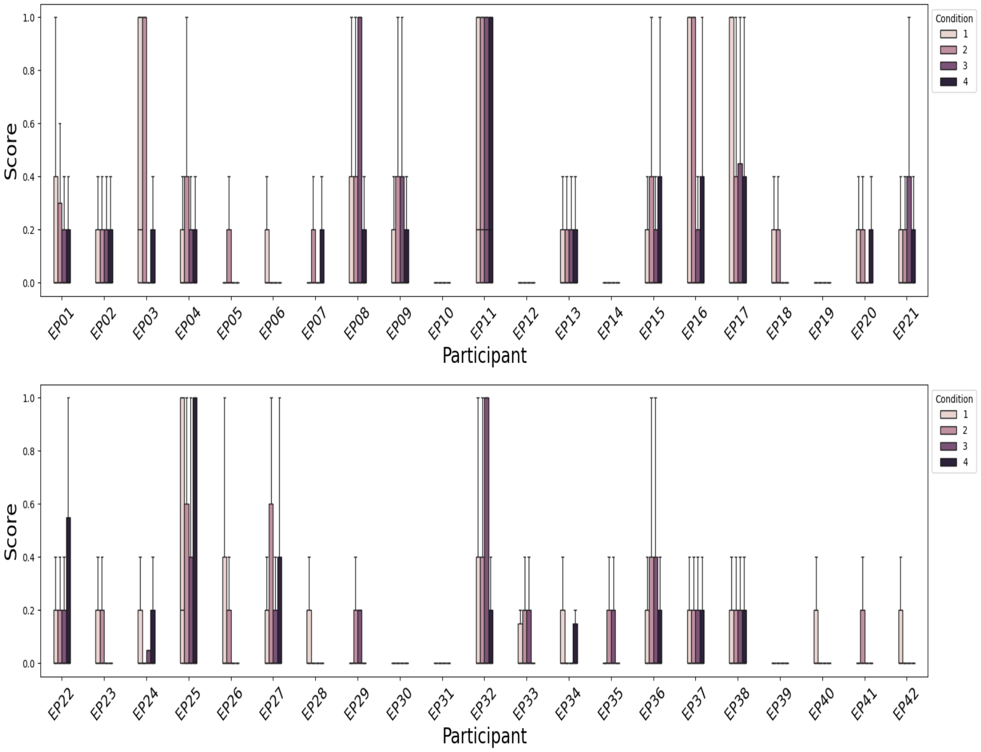


Figure 5.7.: Average scores (lower is better) across participants and experimental conditions indicate that the influence of rhythmic and/or auditory support on the attention score is different across participants. Rhythm (C2) deteriorates the performance more often than improving attention. Auditory support (C3), however, improves the attention score for the majority of participants.

In this section, we classify between the experimental conditions C1, C2, C3 and C4, to assess to what extent the EEG signals differ in the presence of rhythm and/or audio. An example of an EEG multivariate time series for a participant, across different conditions, is shown in Figure 5.8.

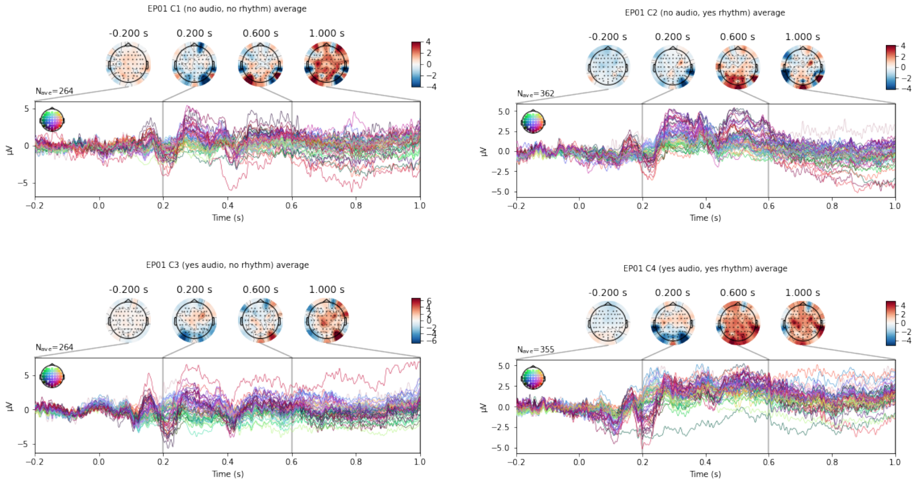


Figure 5.8.: An example of EEG signals across 64 EEG electrodes for participant EP01, averaged across experimental condition C1 (top left), C2 (top right), C3 (bottom left), and C4 (bottom right).

The performance of the different pipelines is rather good for this task (Figure 5.9). In particular, many different representations yield an accuracy above 60% (note that a random guess amounts to 25%) when the classification is performed separately for each participant. This time, the FBN pipeline that focuses on the correlation between the different brain regions performs particularly well, although it fails on the test data consisting of new, subjects that were not previously seen in the train data.

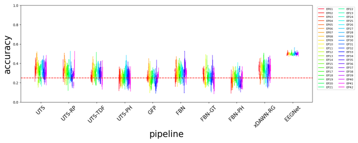
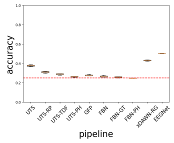
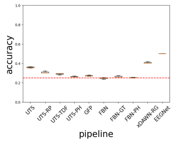
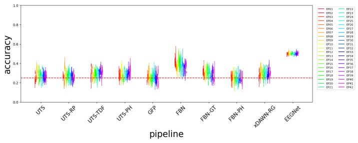
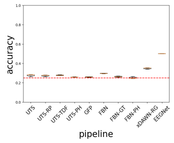
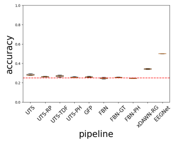
scenario	individual participants	seen participants	new participants
Target			
Distractor			

Figure 5.9.: C1 vs C2 vs C3 vs C4 classification accuracy across the pipelines.

Remember, however, that the rhythmic and/or audio support only accompanies the Target stimuli. To evaluate whether these rhythmic and auditory clues help attention, it is therefore particularly interesting to perform the classification between the conditions for the Distractor stimuli *only*. For the Distractor stimuli, there is no rhythm or audio that can change the EEG signal, but this support that is present during the `Target` stimuli in the sequence might help a person to correctly ignore the `Distractor`. However, the best performing pipeline for the classification of conditions, `UTS`, only achieves an accuracy of 0.28 (although we note that there is only 50% of data available in comparison to the previous experiment).

5.4 Discussion

5.4.1 *Effectiveness of EEG representations in capturing attention*

Attention can be captured with EEG, even for short visual stimulus. The performance for `Target` vs. `Distractor` classification we obtained is similar to the results from other comparable studies which obtain an accuracy between 56.5 and 84% [237, Table 3]. For example, similar accuracy from EEG data of 76.82% is obtained when classifying attentive and inattentive students in [214], and 84% and 81% for respectively the focused attention and working memory in [237]. Therefore, EEG-based BCI platforms have a good potential for utilization real-time classification and neurofeedback tasks, aiding in the diagnosis and training of individuals with attention deficits.

Overall, the representations of the EEG data that reflect the (features extracted from the) univariate time series perform better than the representations that focus on the relationship between these time series for different EEG channels. The former, including the recurrences in the time series, the common time-domain features, and the persistent homology (reflecting the local extrema) are thus all indicative of the use of attentional resources. The latter representations do at times obtain a superior performance on classification tasks for individual subjects, but they fail to obtain a good performance on previously unseen subjects. The communication between different regions in the brain is thus informative of processing visual stimuli (with rhythmic and/or auditory support), but is more sensitive to the cross-subject variability.

It is important to note that the insights might be strongly influenced by the particular problem (task at hand and the experimental set-up) and hyperparameters within the different pipelines. For instance, the WithMe experiment focuses on a novel digit-span paradigm with young students, and different EEG features might be most meaningful for other problems. The WithMe EEG time series reflect the brain signal during only 1.2 seconds of a simple visual task; more meaningful relationships between the different brain regions might be captured for longer resting-state time series. For example, [**vandecappelle2021eeg**] show that the performance of the state-of-the-art models for classifying auditory attention drops significantly when shorter windows are used: the accuracy decreases by 17.6% going from 10s to 1s. In the literature, most of the papers that focus on attention detection from EEG, aim to classify e.g., between long(er) reading or arithmetic task versus resting or sleeping state. Moreover, functional brain networks are more common for diagnostic purposes from long resting-state fMRI data, as EEG has less discriminating power due to its limited spatial resolution [369]. Cross-task classification accuracies, where a classifier is trained and tested on EEG features from different tasks, has been found to be significantly lower (even than a random guess) compared to within-task condition (44.8% and 87.1% respectively), since different tasks invoke highly dissimilar EEG patterns [182]. In addition, relative contribution of different features of stress classification model has been shown to change with the infant's age [197]. Our findings are also limited to our particular choice of the common Pearson correlation used to obtain the functional brain networks, whereas other measures of correlation might be more suitable in capturing attention from (raw) EEG. Furthermore, to address the issue of cross-subject variability, it might be a good idea to perform participant-based normalization. For example, [237] use decibel conversion to normalize with respect to the baseline resting-state EEG data with open and closed eyes. However, the WithMe data does not include such resting state baselines, and we can thus recommend collecting such data during the experiment.

We should note that our goal is not to obtain the best possible performance. A better performance can likely be achieved by combining some complementary representations. For example, the correlation matrix and persistent homology (similar to our FBN and FBN-PH pipelines) combined has been shown to outperform the individual approaches for autism detection from fMRI data [288]. However, our goal is to gain insights in how powerful the different representations are, so that we evaluate the performance of each representation separately.

5.4.2 *Good performance of raw time series*

One of the most surprising insights from our experimental results is that the UTS pipeline (Section 5.2.2.1), which simply concatenates the univariate time series across EEG channels into a large vector, obtains a very good performance on each classification task, often outperforming the very complex EEG-specific models, although we have not observed its usage in the literature. It can therefore be recommended as a good starting baseline, and in case of good performance can avoid further intricate pre-processing and representation techniques (that require expertise in the application domain). This is in contradiction to the common understanding that extracting relevant features for the considered application is a crucial step in EEG processing [367]. Moreover, we employ a simple SVM on the raw time series (in order to have a good indication of the discriminative power of the features), but better results can likely be achieved with more powerful learning models, such as multilayer perceptrons (MLPs) or other deep neural networks. A review of deep learning architectures in the analysis of EEG signals can be found in [86].

5.4.3 *Poor performance of persistent homology*

Although topological data analysis (TDA), and its main tool, PH, has found successful applications in neuroscience (see Section 5.2.2.4, Section 5.2.2.8, Appendix 5.5.1 for references and details), a closer look into the literature often points to a low effectiveness of these features, and a number of possible explanations.

The first TDA pipeline in this chapter, `UTS-PH` (Section 5.2.2.4), calculates 0-dimensional PH on every univariate time series that reflects the EEG data for a single participant, stimulus and EEG channel. This captures the local extrema of the time series, but in case of the WithMe data these might contain a lot of noise, since e.g., a time series corresponding to a `Target` stimulus might contain the EEG information about the forthcoming `Distractor`(s). This might also explain why the `UTS-TDF` where the extrema are taken from the interesting range in the time series, or the `UTS` representation that considers the complete time series (and lets the classifier focus on the important information) outperform the `UTS-PH` pipeline. Persistent homology on univariate time series might be more likely effective in applications where the important difference between data classes lies in some extreme values of the signal, such as epileptic seizure, autism and arrhythmia detection from EEG or ECG in [100, 222, 355, 356].

The second TDA pipeline in this chapter, `FBN-PH` (Section 5.2.2.8), calculates 0- and 1-dimensional PH on a functional brain 64×64 network reflecting the relationship between the time series across 64 EEG channels, for a single participant and stimulus. The poor performance in this chapter is consistent with the experimental results in [141, Figure 3], [288, Figure 4], [148], that point to limited or no success of PH features for brain functional networks. It is also important to note that PH on EEG-based functional brain networks has commonly been employed on correlations between the frequency domains, rather than the correlations between the time series themselves. In this work, we focus on the latter approach, since frequency-domains for short time series provide poor frequency resolution.

Moreover, PH-based representations have previously been shown to be successful on much longer, resting-state fMRI data (more informative than EEG), and for diagnostic purposes (likely an easier task compared to detecting attention during a 1.2-second long visual stimulus). And even in such applications, success is not guaranteed: for instance, [141] note only minor differences and small odd ratios between resting-state fMRI for diagnosing ADHD. There are only a few examples where PH has been used for neurotypical development [140, 367], but these look into the differences between PH on puberty and random MRI networks. In addition, it is more common in the literature to show a statistically significant difference between the patients and the control group, rather than on the more challenging classification or prediction tasks. The potential of PH to reveal the underlying processes from EEG during a short cognitive task might thus be limited.

We again note that the performance might be improved by combining the different representations. Indeed, a number of studies have suggested that persistent homology can extract features that are hardly noticed by other methods, so that their incorporation in deep learning models might yield better results. However, [288] provides a cautionary tale in this regard, as they show that the additional persistent homology features do not necessarily significantly improve the performance of deep learning models. The authors speculate that the poor performance might be attributed to the wide age group in their experiment, although persistent homology also underperforms on the WithMe data in this chapter, where the participants are within a narrow 5-year range.

A survey of some promises and pitfalls of TDA for brain connectivity analysis is provided in [53]. The field of applying topological data analysis, including persistent homology, to cognitive processes is an active area of research, and new studies and methods are continuously emerging. We encourage future research in this direction to help assess the effectiveness or added value of persistent homology in neuroscience applications.

5.4.4 *Auditory and rhythmic support*

Using only the WithMe behavioural data and participants' answers, [92] show the effect of auditory support, but no difference was observed between rhythmic and non-rhythmic sounds. These experiments focus on sequence-based scoring functions (that evaluate attention during a sequence of 10 digits), and we obtain similar findings for our stimulus-based attention score (Figure 5.7). To better understand the underlying processes of attention, as future work the authors in [92] suggest to also analyze the brain activation data such as EEG. Our results show that, although many pipelines can easily differentiate between the EEG data across experimental conditions C1-C4, there is little difference between EEG signals for the distractor stimuli that are not accompanied with rhythmic and/or auditory clues.

There is indeed prior evidence in the literature about the benefit of auditory support [49], but the results about rhythm are conflicting. On the one hand, theories of dynamic attending and predictive coding suggest that rhythms generate expectations that open up slots for attending, facilitating memorization and recall of targets. On the other hand, the accuracy of task performance has been shown to not depend on the synchronisation between the rhythm and target [116, 159]. A possible explanation for the rhythm not providing additional support might lie in the model by [174], which suggests that attentional resources are drawn from a general, but limited pool of resources: memory requires the major part of available resources, so that not enough resources can be assigned to process the rhythmic support. This explanation is however less likely since healthy adults have on average a digit span between 5 and 9, so that memorizing 5 targets should not put too big of a demand on the available cognitive resources.

A more likely explanation can be found in a few methodological issues. Firstly, in order not to draw attention to the support and following the directed attention hypothesis, the conditions C1-C4 do not follow a block experimental design but the order is pseudo-randomized (so that even two auditory or rhythmic conditions would rarely appear consecutively), and participants were not made aware of the sound or rhythm (whereas some research suggests that conscious attention to both sensory modalities is essential for performance improvement [343]). Indeed, even though the presence of rhythm did not improve attention in general, it did so for the participants who reported experiencing the rhythm as supportive [92]; however, it might be that these participants were more alert overall, what led to a better performance. Secondly, the priming of the rhythm with only 5 induction stimuli might have not been sufficient, and the induction with flashing empty circles might also be improved with moving stimulus such as a bouncing ball [92].

We stress, however, a difference in the of auditory and/or rhythmic support across participants. This added value has been shown to be influenced by the subject's bias for auditory or visual memory, and their previous task experience, both in earlier studies as well as recently for the WithMe data [92].

DATA AVAILABILITY The WithMe questionnaire, the participants' behavioural data and recalled answers are available at the Open Science Framework (OSF) repository: `https://osf.io/ntmy8/?view_only=88d951C394c7481dba00a1497d64797f`. The complete raw, unepoched and unprocessed WithMe EEG will be released later as part of a separate publication. The pre-processed unepoched data (42 .fif files, one for each experimental participant) is available on the figshare repository: `https://figshare.com/s/f3a0727f6ec6de76fd4b`. For easy reference, we also include the processed and epoched data (as a large .csv file), which also includes the attention scores, as supplemental material. The code used for the experiments in this chapter is publicly shared on `https://github.com/renata-turkes`. In particular, Jupyter notebooks `data_exploration.ipynb` and `classification.ipynb` provide user-friendly tools to respectively explore the data (further), replicate or extend our analyses.

5.5 Supplementary material

5.5.1 *Topological data analysis and neuroscience*

Topology has long been a central concept in neuroscience [120]. In particular, persistent homology (PH), has found many applications.

One reason behind the application of PH to neuroscience is that a significant portion of neuroscience research involves studying functions or graphs, that are particularly amenable to topological tools [89]. The functions can, for instance, correspond to the functional magnetic resonance imaging (fMRI) time series, and graph vertices can correspond to neurons, cell types, fMRI voxels, with the graph edges reflecting their relationship (e.g., synaptic connections between neurons). A survey of applications of topology in neuroscience can be found in [89]. Some examples of application of PH in neuroscience are analysis of: spatial structure of afferent neuron terminals in crickets [42], correlations of neurons in the hippocampus [138], neuron activity (firing event time series, so-called spike trains) [325], branching neuronal trees [176], brain artery trees [29, 35], morphological brain networks in deaf adults [186], fluorodeoxyglucose-positron emission tomography (FDG-PET) based brain networks in children with attention deficit hyperactivity disorder (ADHD) and autism [202, 203], functional magnetic resonance imaging (fMRI) volumes [297], fMRI brain networks [117, 272, 327, 329] (respectively during learning, drug-induced states, in patients with ADHD or schizophrenia), or structural covariance MRI (scMRI) or fMRI networks of persons with autism [261, 288].

There are a few different ways that PH can be calculated from EEG data (or any dataset of multivariate time series). For instance, one can compute PH on a point cloud, a geometric object obtained from a time series via the so-called Takens's time delay embedding, which is circular if the time series is periodic, under certain conditions [270]. Since 1-dimensional PH precisely describes the loops in the data, it provides insights into periodic and repetitive patterns in the time series, and promises to be beneficial in particular to distinguish periodic and chaotic time series [312]. In such applications, time series belonging to the same class do not have to have similar waveform and similar range, so that this PH-based model might be more meaningful than the typical approaches for time series classification (e.g., temporal or frequency features, or CNNs that learn the best features for the given task). It is possible to choose a summary of PH that is shape agnostic and resistant to dampening [269], what might be useful in dealing with cross-subject variability. This pipeline has been employed on EEG data for epileptic seizure [133] or cognitive fatigue detection [90], classification between eyes-open and eyes-closed signals [34], or between left and right-hand motor intentions for brain-computer interface (BCI) applications [8, 9]. A recent review of TDA applications to EEG analysis can be found in [367].

In this chapter, we apply PH both on univariate time series for a single EEG channel (i.e., functions, `UTS-PH`, Section 5.2.2.4) and on the networks resulting from the EEG multivariate time (i.e., graphs, `FBN-PH`, Section 5.2.2.8). In the remainder of this section, we discuss our choice of filtration and persistence signature, the input and output of PH, and illustrate their invariances under some transformations of a function or network (that in our application correspond to univariate EEG time series, or functional brain networks that reflect the relationships between them).

Next to the common Vietoris-Rips filtration, there is a number of different ways one can build a filtration, the input for PH. PD always registers the number of k-dimensional cycles (topological information), but the choice of filtration has crucial importance for the interpretation of birth and death values (additional geometric information), and for the type of data transformations that the resulting PH is invariant to. For example, the birth and death values with respect to the Vietoris-Rips filtration reflect respectively the sparsity of the neighbouring point cloud points and their radius or size, so that they are invariant under rotation or translation, but not under rescaling or outliers (Chapter 3). In this section, we focus our attention on the two filtrations that we will use in this work.

5.5.1.1 *Lower-star filtration*

Let $f : \mathbb{R} \to \mathbb{R}$ be a function. The lower-star filtration $\{K_r\}_{r \in \mathbb{R}}$ considers the sublevel sets

$$K_r = \{x \in \mathbb{R} \mid f(x) \leq r\}.$$

Every sublevel set K_r is either empty or a union of intervals, so the only non-trivial topological information they carry is their 0-dimensional homology, i.e., the number of connected components [69]. Since PH tracks how these connected components evolve within the filtration, it will capture the information about the local minima and maxima of f (Figure 5.10). Note that PH does not capture the width of the minima and maxima peaks, so that it is invariant to expansion and contraction in the x-axis direction.

5.5.1.2 *Rank filtration*

Let $G = (V, E)$ be a weighted graph or a network. In general, an algebraic representation of a weighted graph $G = (V, E)$ is a connectivity matrix $D = [d_{ij}]$, where d_{ij} is the weight of an edge $e_{ij} \in E$ between vertices $v_i \in V$ and $v_j \in V$, reflecting the relationship between them, which is typically some measure of distance, dissimilarity or inverse correlation. It is possible to build the standard Vietoris-Rips filtration of a graph, since it never relies on the point cloud coordinates, but rather only on the pairwise distances d_{ij} between the point cloud points (or graph vertices or nodes).

In applications, however, it is often the case that these graph distances are measured in somewhat arbitrary units, where one can usually assume a larger weight means a stronger interaction, but the precise values of the weights may not be meaningful [89]. This is particularly true for neuroscience applications, where correlations are believed to be given by an unknown monotonic function on an underlying distance in the relevant stimulus space, i.e., distances are only known up to rescaling by an unknown monotonic function [46, 138].

In such scenarios, it is useful to consider the rank filtration, which corresponds to the Vietoris-Rips filtration, but after the scalar values in the distance matrix are

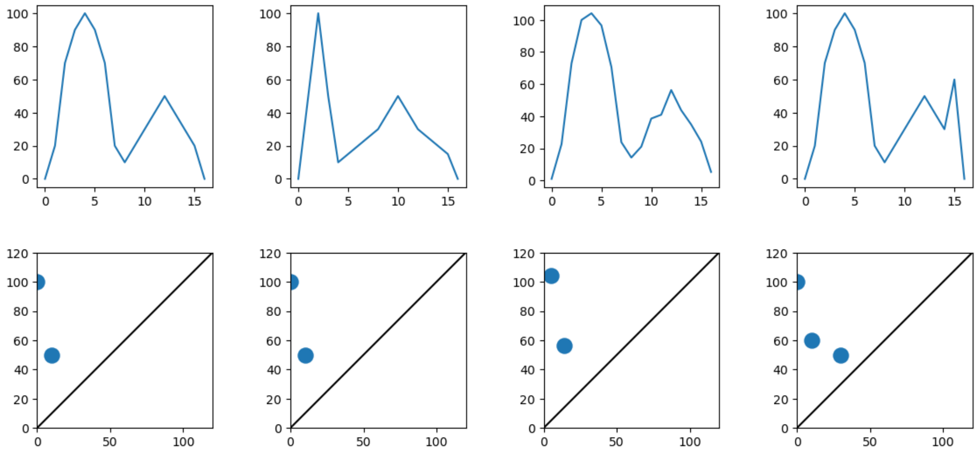


Figure 5.10.: Persistent homology of a function with respect to the lower-star filtration captures the information about the local maxima and minima of the function, and is invariant under expansion and contraction in the time axis direction. The figure shows four example functions $f : \mathbb{R} \to \mathbb{R}$ (top panel) and their respective 0-dimensional PDs (bottom panel). For the first example f_1, $\text{PD}_1 = \{[0, 100], [10, 50]\}$, reflecting the minimum values 0 and 10, and maximum values 100 and 50. Since the second function f_2 has the same extrema, $\text{PD}_2 = \text{PD}_1$, despite the width of the extrema being different. Function f_3 is a noisy version of f_1, resulting in a slightly different persistence intervals $\text{PD}_3 = \{[5.27, 104.17], [14.24, 56.32]\}$. Function f_4 has additional extrema, so that the PD sees additional persistence interval, $\text{PD}_4 = \{[0, 100], [10, 60], [30, 50]\}$.

replaced by their ordinal numbers or ranks. In other words, the rank filtration is a nested family of graphs (clique complexes of G), where each subsequent graph includes an additional edge $(i, j) \in E$ corresponding to the next-lowest distance d_{ij}. PH with respect to the weight rank filtration is then a topological invariant, as it remains unchanged under nonlinear transformations of the weights, provided that the ordering of weights is preserved (Figure 5.11).

5.5.2 *Different representations of an example multivariate time series*

In this appendix, we consider two toy examples of 5×100 multivariate time series. For the first example, the first univariate time series is a cosine signal with a few minima and maxima. The remaining univariate time series in this example are obtained by applying respectively the following transformations to the starting signal: a shift in the y-axis direction, "stretching" a part of the time series in the x-axis direction, and adding two different levels of noise. The second multivariate

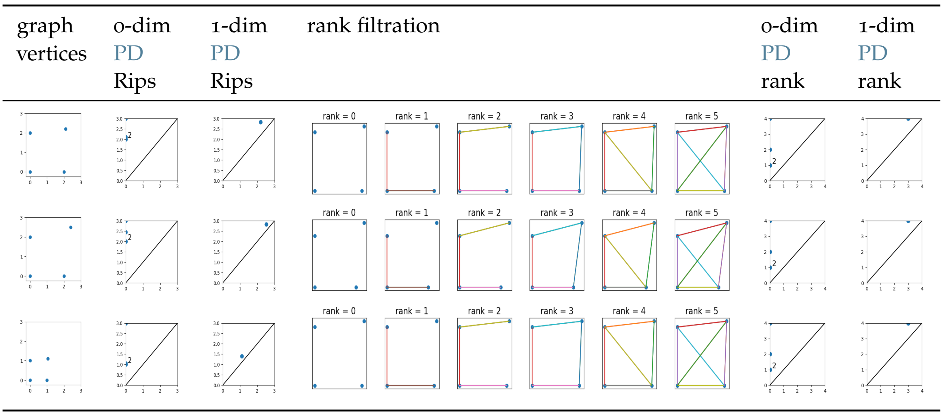


Figure 5.11.: Persistent homology of a graph with respect to the rank filtration captures the connectivity and holes, invariant under rescaling of distances by a monotonic function. The 0- and 1-dimensional PDs with respect to either the standard Vietoris-Rips or the rank filtration reveal respectively the four connected components and one hole. However, the persistence intervals (i.e., their birth and death values) with respect to the standard Vietoris-Rips filtration differ for the three example graphs, since they rely on the actual distances between graph nodes. The rank filtration instead only considers the order of distances, so that the PDs are the same for the three graph examples.

time series example consists of the same first univariate time series, with the last one being replaced with a completely noisy signal. Figure 5.12 visualizes the different representations of the multivariate time series considered in this work (Section 5.2.2), and gives some insights about the noise robustness of different features.

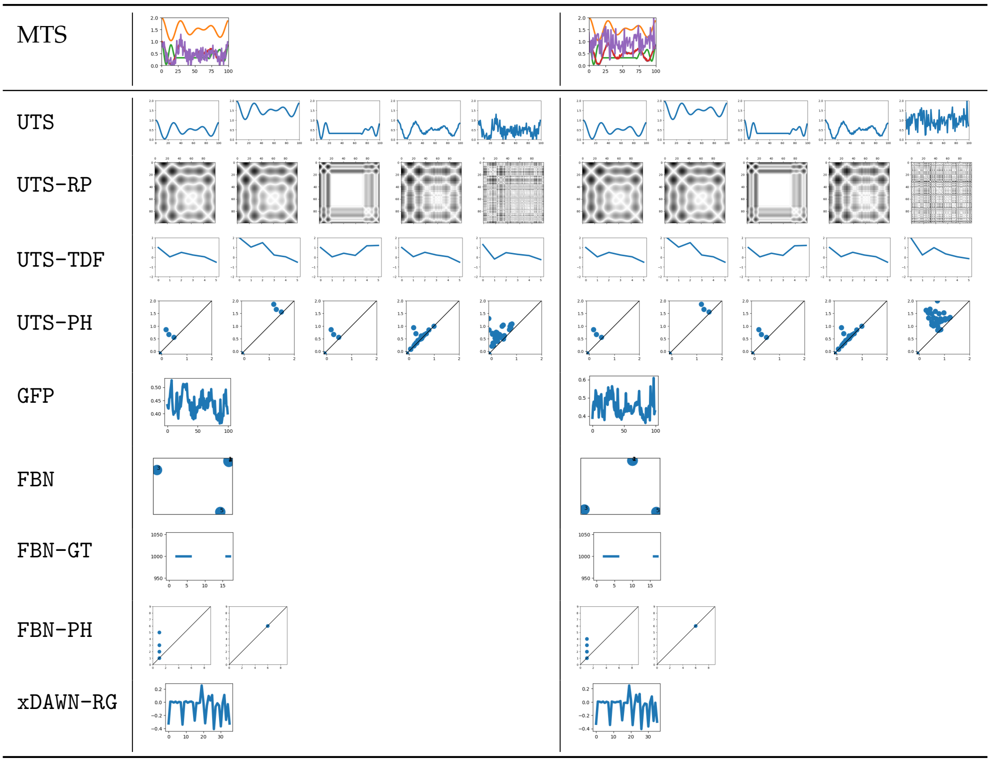


Figure 5.12.: Different representations of two example multivariate time series (MTS).

6

Lessons learned on myths about persistent homology

The insights from this thesis help us to “bust some myths” about persistent homology (PH), i.e., to point to some statements too often found in the literature that are at times not true, not precise enough, or risk misinterpretation, as discussed below.

PERSISTENT HOMOLOGY IS A TOPOLOGICAL SUMMARY. While PH is the main tool of TDA and likely developed in order to capture the homology of data, and while it indeed captures these topological features, it tracks how the features appear and disappear within a filtration, a nested family of spaces that approximate the data at different scales (Section 1.3). This additional information can reflect other geometric aspects of shape, such as curvature (Section 2.3) or convexity (Section 2.4). This is also highlighted with a sensitivity of PH to some affine transformations such as translation, rotation, stretch or shear (Figure 2.2, Figure 3.5, Figure 3.6, Table 3.4, Table 3.7, Figure B.8). Moreover, an intuitive discussion of the relationship between geometry, topology, homology and PH is provided in Chapter 1, and revisited here (see Figure 1.6 and Figure 6.1). See also Appendix A.6 where we discuss how PH can be used to detect the topological or geometric signal in a number of synthetic examples.

PERSISTENT HOMOLOGY IS ABOUT HOLES (AND THEREFORE ONLY SUITABLE TO DATA SETS WITH HOLES). Firstly, it is important to highlight that PH captures cycles in any homological dimension (that are at times referred to holes), including 0-dimensional PH which reveals the connected components or clusters, 1-dimensional holes or loops, but also cycles in higher dimensions such as 2-dimensional voids. Moreover, while PH can indeed reveal the number of holes (Section 2.2), as we note above, PH captures additional information about these cycles, that can reflect other geometric aspects of shape, such as curvature (Section 2.3) or convexity (Section 2.4).

Persistent homology is calculated with respect to Vietoris-Rips simplicial complex on Euclidean distances. This canonical choice of the filtration for PH is assumed and often not even mentioned in the largest portion of the literature, which hinders our understanding of the potential and flexibility of PH. What is more, many statements that have been shown to hold for such PH are then assumed true for PH in general, which is often far from the truth. We highlight this versatility by employing or explicitly acknowledging the Vietoris-Rips simplicial complex with respect to geodesic (Appendix A.4.1 and Appendix A.5.1) or ranked distances (Section 5.2.2.8), alpha and cubical complexes (Appendix A.4.1 and Appendix A.5.1), and the many different filtration functions throughout the thesis, such as binary, greyscale, density, radial, Distance-to-Measure, height, or tubular (see, for instance, Figure 1.9, Figure 3.1). We show the very different behaviors exhibited by PH across different filtrations, e.g., the different topological and geometric features that are captured (Chapter 2), or different sensitivity to noise (Chapter 3). In particular, see Remark 2, Appendix A.6, Section 3.3 Table 3.1, Figure 3.5, Figure 3.6, Table 3.4, Table 3.7, Appendix B.4.

Persistent homology is represented with a persistence diagram. Different signatures such as the common vectorized summaries PLs, PIs (Section 1.4, and used in experiments throughout the thesis) are commonly used in applications, since PDs are not amenable for statistical learning. These persistence signatures capture the same or similar information [1] , but lie in different metric spaces and therefore exhibit different behavior (Chapter 3). See in particular Table 3.2, Section 3.3 Figure 3.5, Figure 3.6, Table 3.4, Table 3.7, and also Appendix A.6. Moreover, in practical applications where there is some understanding of the signal, some simplified representation is sufficient, e.g., lifespans of 10 most persistent intervals (for detection of number of holes in a data set with maximum 9 holes, in Section 2.3), or the 2nd longest lifespan (for classification between convex and concave shapes in Section 2.4).

Persistent homology is stable. Stability theorems hold for filtration functions that satisfy some assumptions, and make a statement about a particular persistence signature and corresponding metric (Section 3.3). For example, it is a common choice in the literature to consider PIs or PL with the Euclidean metric, whereas these signatures are not stable with respect to Wasserstein W_2 metric.

Stability of persistent homology means that perturbation of persistent homology under noise is small. The stability theorems guarantee that

[1] The map that sends PDs to PLs is a bijection, so that we can recover the PD from its PL; in practice, however, there may be some loss of information since we limit the number of landscape functions, that are also further discretized. While PI typically captures similar information to PI, some information can be lost due to smoothing and discretization (multiple closeby persistence pairs might blend into the same grid cell), and the choice of the PI weight function can further highlight different parts of the PD.

the perturbation of PH is upper bounded by perturbation in the filtration function, which might be very large and thus of little use in cases where we think of the data being perturbed only slightly (see, for example, Table 3.1). This is, for example, the case for PH with respect to the Vietoris-Rips filtration, that is significantly perturbed in the presence of even a single outlier.

STABILITY OF PERSISTENT HOMOLOGY IMPLIES NOISE ROBUSTNESS IN A CLASSIFICATION TASK. For some data, the perturbation of some PH under some noise might be larger than the difference between PH across data classes, so that the PH features are not robust to such noise for the given classification problem (Section 3.4.3).

LONG PERSISTENCE INTERVALS ARE SIGNAL, SHORT INTERVALS ARE NOISE. Studying a number of different problems with a variety of PH pipelines has helped us explain how long intervals reveal topology, every interval can reveal geometry, and any interval can encode the signal for the given problem. This is particularly highlighted in Section 2.4, where we show that the persistence of one short interval detects convexity, and discussed in a lot of detail in Section 2.5 and Appendix A.6.2. Also, intervals of medium persistence are shown as most discriminative for an example application (Appendix B.4, see in particular Figure B.14).

PERSISTENT HOMOLOGY IS A FEATURE EXTRACTION TECHNIQUE. In many applications of PH, each data sample (e.g., a point cloud, graph or an image) is summarized with its PH features that are further used in statistical analysis, classification or regression, as we demonstrate in the application of PH in neuroscience in Chapter 5 (and the same approach is used in Chapter 2, Chapter 3 and Appendix B where we analyze some properties of PH). However, PH can also be employed as a data exploration technique, and can therefore also be used to e.g., study the structure of data before and after some encoding or transformation (Chapter 4), such as any dimensionality reduction technique.

PERSISTENT HOMOLOGY FACES IMPORTANT COMPUTATIONAL LIMITATIONS. While it is true that the computation of PH can be challenging for large data (the theoretical computational complexity of PH is discussed in detail in Appendix A.1.4), in Appendix A.2 we evaluate the computational resources in our experiments and conclude that most often PH pipelines run faster and consume less memory than the other baseline methods, namely SVM or MLP on distance matrices, or PointNet, bechmark deep learning architecture for point clouds. Moreover, there are numerous methods to tackle the computational issues: for example, Vietoris-Rips simplicial complex can be calculated with the Rips-specific and efficient library `Ripser`, or replaced with alpha or cubical complex (as in Section 2.2, Section 2.3, Section 2.4), and many PH libraries such as `Gudhi` or `Ripser` implement different techniques for the sparsification of simplicial complexes, or optimized matrix reduction algorithm to calculate PH.

Conclusions

The main goal of the work in this thesis is to gain a better understanding of the properties and behavior of persistent homology (PH), the main tool in topological data analysis (TDA), and in particular how these translate to practical applications. We start with a background on PH in **Chapter 1**, that provides an intuitive exposition of PH for a broad audience of computational scientists. We discuss how it relates to the notions of geometry, topology and homology, and introduce different examples of filtrations and signatures, the input and output of PH. This chapter ends with the stability theorems that guarantee the robustness of PH under minor perturbations of the input. In the first part, Chapter 2 and Chapter 3, we investigate what is seen with PH, and what the stability theorems mean in practice, also looking into to what extent they guarantee noise robustness of PH in a classification task. The second part, Chapter 4 and Chapter 5, focuses on two applications of PH, both as a data exploration technique where PH is applied to the data set, and as a feature extraction technique where each data sample in the data set is described with its PH information.

The primary contributions of the thesis lie in **Chapter 2 ("the what")**, where we look into the topological and geometric features that can be captured with PH, to gain insights into the types of problems PH can effectively address. We focus on the detection of the number of holes, curvature, and convexity. Theoretical evidence for the first two classes of problems has been established in the literature, and we prove a new result that guarantees that PH can detect convexity. To do so, we introduce a new filtration that calculates the distance from a line, that we call tubular filtration. We also demonstrate the practical utility of PH by showing that PH can solve all three problems for synthetic point clouds in $\mathbb{R}^2$ and $\mathbb{R}^3$, outperforming a few baselines, including PointNet, a deep learning architecture inspired precisely by the properties of point clouds. This is true even when there is limited training data and computational resources, and for noisy or out-of-distribution test data. For convexity detection, we also show the effectiveness of PH in a real-world plant morphology application.

Studying these three problems together also helps us to highlight how the choice of filtration and persistence signature influences the information that is captured with PH, and consequently, to formulate the guidelines for the use of PH in applications (Appendix A.6). Indeed, for the detection of the number of holes, curvature and convexity we respectively rely on: (i) lifespans of the 10 most persistent 1-dimensional cycles with respect to the Vietoris-Rips filtration on Euclidean distances, (ii) 0- and

1-dimensional persistence landscapes with respect to the Vietoris-Rips filtration on geodesic distances, and the (iii) lifespan of the second most persistent 0-dimensional cycle with respect to tubular filtration on a cubical complex, from nine different lines. Moreover, we discuss how employing the persistence diagrams for any homological dimension and the height filtration from any direction, it is possible to completely describe the shape (Remark 3). A similar result holds for the tubular filtration which, defined as a distance from a line, is more fine-grained than the height filtration that considers distances from a hyperplane (see Remark 2).[2] With these insights, we can revisit the relationship between geometry, topology, homology and PH from the Introduction, as PH has the potential to completely recover the geometry of the space (Figure 6.1). Due to the crucial role of shape classification in understanding and recognizing physical structures and objects, image processing and computer vision, to a variety of applications [211], the list of potential applications of PH is immense.

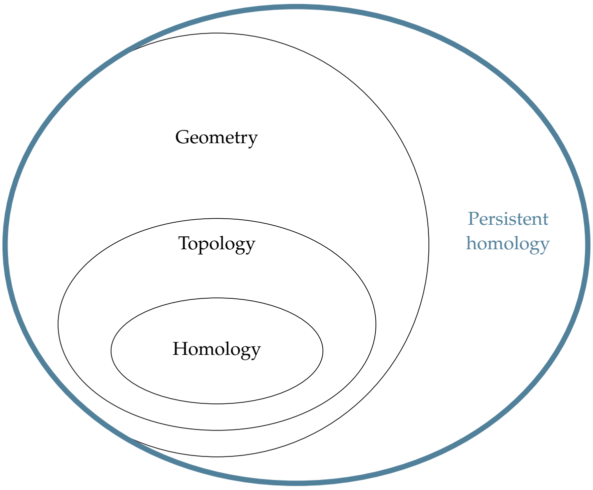


Figure 6.1.: Persistent homology is shown to be able to completely recover the geometry. Since it is possible to define arbitrary filtration function values on the data, persistent homology can capture additional information beyond geometry (such as color, temperature, intensity, etc.)

Furthermore, the results in this chapter advance the discussion about the importance of long and short persistence intervals, and their relationship to topology and geometry (Section 2.5, Appendix A.6.2). In the existing literature on PH, it is often implied that long persistence intervals capture the topological signal, whereas short intervals are noise. In recent years, however, it has been shown that the many short persistence intervals, when considered together, hold significant distinguishing power for specific applications. We highlight that *all* cycles can encode geometric

[2] The proof of this results is a part of ongoing work with Adi Onus and Nina Otter, where we study the properties of PH with respect to the new tubular filtration, which is another direct extension of this chapter.

information, such as the information about their size (with respect to the Vietoris-Rips and related filtrations, as in Sections 2.2 and 2.3) or their position (with respect to the height or tubular filtration, as in Section 2.4). 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 2.4, where we show that a single interval detects convexity. We conclude that the 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.

In **Chapter 3 ("the why")**, we study the implications of stability of PH, the main theoretical result in TDA, for applications in practice. We carry out experiments on the MNIST data set of images of handwritten digits, under different types of image transformations or noise. These results help us highlight the importance of the choice of (i) filtration, (ii) persistence signature, and (iii) data for the stability of PH and its noise robustness in a classification task. Firstly, stability results ensure that the distance between two PDs is upper bounded by the distance between the filtration functions (and not on the input data), and impose some assumptions on the filtration functions. The experimental results demonstrate that sensitivity to noise differs greatly across filtrations, and that in the presence of any type of noise, the performance of PH features for classification drops significantly for some of these choices. Secondly, it is often neglected in the applications of PH that some persistence signatures are proven not to be stable. Even when shown to be stable, these persistence signatures lie in different metric spaces, influencing the type of perturbations of PH that are emphasized. The experiments show a clear relative difference in the extent to what different persistence signatures are sensitive to various types of noise, and a difference in the noise robustness in a classification task. And thirdly, even if the stability theorem holds for the given filtration and signature, it might not yield noise-robust features in a classification task. Whether the perturbation of PH under noise is small enough (to not disturb the classification accuracy) depends on the distance between filtration functions across different data classes.

In summary, in this chapter we quantify the sensitivity to various forms of noise PH across different filtrations and persistence signatures, and help explain how the stability of PH to noise does not imply the noise robustness of PH features for any data set, or classification problem. Understanding these variations informs the selection of appropriate PH parameters for specific applications, improving the reliability and performance of PH in real-world scenarios where noise or measurement errors are present in the data.

The final two chapters explore some applications of PH. In **Chapter 4 ("the how")**, we employ PH to better understand the hyperdimensional computing (HDC) encoding that sends the input data into a hyperdimensional (HD) space, of dimension 10 000. We compare the PH of the input data and the HD data across several synthetic and real-world Euclidean data sets, and conclude that HDC approximately preserves the structure, with some local geometry that may be lost. In **Chapter 5 ("the when")**, PH is employed for attention detection from electroencephalography (EEG) data. Positioning the poor performance of PH features in the existing applications of PH in neuroscience, points to the type of applications where PH may (not) have an advantage. Firstly, our PH features calculated on univariate time series (functions) capture the local extrema and are therefore more likely to be effective in applications where the important difference between data classes lies in some extreme values of the signal, such as epileptic seizure, autism and arrhythmia detection. Secondly, due to the limited spatial resolution of EEG data, the PH features calculated on the multivariate time series, i.e., on the functional brain networks that reflect the relationship between different regions in the brain, are more suited to for diagnostic purposes from long resting-state fMRI data.

The insights help us to "bust some myths" about PH, i.e., some statements too often found in the literature that are at times not true, not precise enough, or risk misinterpretation:

- PH is a topological summary.
- PH is about holes (and therefore only suitable to data sets with holes).
- PH is calculated with respect to Vietoris-Rips simplicial complex on Euclidean distances.
- PH is represented with a PD.
- PH is stable.
- Stability of PH means that perturbation of PH under noise is small.
- Stability of PH implies noise robustness in a classification task.
- Long persistence intervals are signal, short intervals are noise.
- PH is a feature extraction technique.
- PH faces important computational limitations.

Chapter 6 discusses each of these statements in detail, where we point to the specific results in the thesis that provide an understanding on how these statements can be untrue or misleading (for the given problem, or the choice of input and output of persistent homology).

In conclusion, in the thesis we move away from the dominant directions in the literature, going beyond the canonical choices of filtrations and persistence signatures, to highlight how PH offers remarkable flexibility that makes it a versatile descriptor of different aspects of shape, with different behaviors. The broad range of filtrations and signatures is at the same time an important limitation, as it requires expertise and makes it more difficult to effectively use in applications; what is more, experts too still do not have a good understanding of the power and potential of PH. While we provide some insights into these questions that consequently inform some guidelines for applications, further research into the interpretability of PH is crucial.

Firstly, further work can be done to understand the other type of topological and geometric features (e.g., orientability, singularities, eccentricity, relative position) that can be captured with PH.

In this context, saliency maps that assign importance scores to each point in the input point cloud, or each pixel in an image, are also an important tool that can explain their significance in a PH-based analysis. In Appendix B.7.4, we visualize the point saliency maps for a few example point clouds using the pull-back geometry approach; more detailed studies on different data sets (including real-world data) is a promising avenue for further work. The study of representative cycles is another promising approach in this regard, as it provides an understanding of a given persistence interval in terms of the underlying data. In the context of geometric data, this takes the form of an "inverse problem", constructing geometric structures corresponding to each persistent interval in the original input data. However, representative cycles are not unique, what causes ambiguity and can lead to many different interpretations, prompting the need for further improvements. Moreover, there is enormous potential in the more complex, but relatively unexplored scenario of multiparameter persistence, that calculates PH with respect to multiple filtrations at the same time. Since we highlight how different filtrations can capture different aspects of shape, multi-parameter persistence offers further flexibility and more nuanced and detailed information about the data.

To enhance the interpretability of stability results, future work should aim to refine the upper bounds in PH stability theorems by relating them to relevant distance $d(X, X')$ between data X and its noisy variant X'. A good example is the upper bound for the distance between PDs with respect to the Vietoris-Rips filtration, as it clearly guarantees that the PD remains unchanged under rigid transformations (rotation, translation). Comparing these bounds across filtrations can help guide the optimal filtration choice for specific problems and noise types.

The solid theoretical framework and the versatility of PH make it a promising approach for many further applications. When used as a feature extraction technique, it can be expected to work well when the the data classes differ in their topology and/or geometry (such as their connected components or clusters, holes, curvature, or convexity), which is not true for any application (Chapter 5). However, it is of crucial importance to decide on a good choice of a filtration (or, filtrations) and persistence signatures; we provide some guidelines throughout the thesis (see in particular Appendix A.6 and Chapter 3). Alternatively, as a data exploration technique, PH can provide insights about the topology and/or geometry of the given data (as in Chapter 4, when we analyze to what extent does the HDC encoding preserves the structure), or any object of interest. For example, we can track PH of the weights of a deep neural network during training, or PH of their decision boundaries through the layers, to gain some understanding of the learning process (i.e., to "unbox the black box"). Although there is some research in this direction in recent years, we believe that the potential and flexibility of PH is yet to be realized.

To exploit the versatility and potential of persistent homology in practice, further work is also needed in improving the persistent homology software. What is mainly lacking in the existing libraries are the readily available implementations of different filtrations, that are primarily responsible for the various aspects of shape that can be captured with persistent homology. Furthermore, these libraries should also be able to directly tune the choice of filtration and signature (that optimizes the performance for the given problem), and ensure seamless integration with other machine learning frameworks. Some good existing examples in the literature are PersLay, a simple neural network layer [56], or ATOL, an unsupervised vectorization method [304] that learn the best vector summary of persistence diagrams; further extensions to learning the complete PH pipeline (i.e., including the best filtration) would make significant contributions. Together with its improved interpretability discussed above, topological data analysis can in this way be made into a technology that is widely accessible to non-specialist practitioners.

APPENDIX

A

On the effectiveness of persistent homology: Supplementary material

This appendix consists of the supplementary material that accompanies Chapter 2, i.e., the corresponding publication:

> Renata Turkeš, Guido Montúfar, and Nina Otter, *On the effectiveness of persistent homology*, Advances in Neural Information Processing Systems 35 (NeurIPS 2022): 35432-35448.

The first part includes the detailed proof of our new result that guarantees that 0-dimensional persistent homology (PH) on the newly introduced tubular filtration detects convexity (Theorem 2.2). It also lists the existing theoretical results about the detection of the number of holes and curvature, and discusses the computational complexity of the proposed PH pipelines. The second part provides additional details about the experiments on synthethic 2D and 3D point clouds in Chapter 2 (such as the choice of parameters or computational resources). The findings inspire some guidelines for the use of PH in applications, and the final part demonstrates the detection of a concavity measure on a real-world FLAVIA data set of images of plant leaves.

A.1 Theoretical results

In this section, we provide the proof of our Theorem 2.2 that guarantees that PH can be used to detect convexity. We then formulate Theorem A.1 and Theorem A.2 from the literature, that summarize known theoretical guarantees that PH can detect the number of holes and curvature. At the end of the section, we also discuss some known results about the theoretical computational complexity of PH.

A.1.1 *Convexity*

In our pipeline for the detection of convexity we consider tubular filtrations, akin to the concept of tubular neighborhoods in differential topology. Given a subspace X of $\mathbb{R}^d$, and a line $\alpha \subset \mathbb{R}^d$, we consider all the points in X that are within a specific distance r from the line. By varying r, we then obtain a filtration of X. We note that while a tubular neighborhood is defined with respect to any curve, here instead we focus on the special case of (closed) neighborhoods with respect to a line.

Definition 2.1 (Tubular filtration) Given a line $\alpha \subset \mathbb{R}^d$, we define the **tubular function with respect to** α as follows:

$$\begin{aligned} \tau_\alpha \colon \mathbb{R}^d &\to \mathbb{R} \\ x \ &\mapsto d(x, \alpha)\,, \end{aligned}$$

where $d(x, \alpha)$ is the distance of the point x from the line α. Given $X \subset \mathbb{R}^d$ and a line $\alpha \subset \mathbb{R}^d$, we are interested in studying the sublevel sets of τ_α, i.e., the subsets of X consisting of points within a specific distance from the line. We define

$$X_{\tau_\alpha, r} = \{x \in X \mid \tau_\alpha(x) \leq r\} \ = \{x \in X \mid d(x, \alpha) \leq r\}\,.$$

We call $\{X_{\tau_\alpha, r}\}_{r \in \mathbb{R}_{\geq 0}}$ the **tubular filtration with respect to** α.

We first formulate and prove our main theorem below, and then discuss the need for tubular filtration in Remark 2.

Remark 1 (Different notions of components). In the proof of Theorem 2.2, we need to work with path-connected components. We note that while in the main part of this manuscript we always only use the term "components", more precisely one would need to distinguish between "connected components" and "path-connected components". For the purposes of the majority of the spaces that we consider in this work, the two notions are equivalent. Thus, we often simply refer to these as "components".

Theorem 2.2 (Convexity detection with PH) *Let $X \subset \mathbb{R}^d$ be triangulable*[1]*. We have that X is convex if and only if for every line α in $\mathbb{R}^d$ the persistence diagram in degree 0 with respect to the tubular filtration $\{X_{\tau_\alpha,r}\}_{r\in\mathbb{R}_{\geq 0}}$ contains exactly one interval.*

Proof. We first note that the persistence diagram in degree 0 of $\{X_{\tau_\alpha,r}\}_{r\in\mathbb{R}_{\geq 0}}$ exists, since the singular homology in degree 0 (and with coefficients in a field) of $X_{\tau_\alpha,r}$ is finite-dimensional for any $r \in \mathbb{R}_{\geq 0}$; the existence of the persistence diagram then follows from [65, Theorem 2.8][2].

Assume that X is convex. By definition, we have that for all $p_1, p_2 \in X$ the straight-line segment between p_1 and p_2 is contained in X. Let α be any line in $\mathbb{R}^d$ and $r \in \mathbb{R}$. By elementary properties of Euclidean spaces (similarity of triangles, see Figure A.1), we have that if $d(p_1, \alpha) \leq r$ and $d(p_2, \alpha) \leq r$, then also $d(q, \alpha) \leq r$ for any point q on the line segment between p_1 and p_2. By the definition of tubular function, this means that $p_1, p_2 \in X_{\tau_\alpha,r}$ implies that $q \in X_{\tau_\alpha,r}$. Therefore the straight-line segment between p_1 and p_2 is contained in $X_{\tau_\alpha,r}$, which means that $X_{\tau_\alpha,r}$ is convex, and thus path-connected. We therefore have that for any line α, the persistence diagram in degree 0 of $\{X_{\tau_\alpha,r}\}_{r\in\mathbb{R}}$ contains a single interval.

Assume now that X is concave. Then by definition there exist $p_1, p_2 \in X$ and a point q on the straight-line segment between p_1 and p_2 such that $q \notin X$. Since X is closed, we have that there exists $\epsilon > 0$ such that $B(q, \epsilon) \subset \mathbb{R}^d \setminus X$ (Figure A.2). Let α be the line passing through p_1 and p_2, and let $0 \leq r \leq \epsilon$. We then have that $d(p_1, \alpha) = d(p_2, \alpha) = 0 \leq r$, so that $p_1, p_2 \in X_{\tau_\alpha,r}$. We claim that the subset $X_{\tau_\alpha,r}$ is not path-connected.

Let us assume otherwise, i.e., that that $X_{\tau_\alpha,r}$ is path-connected. Equivalently, there is a path connecting p_1 and p_2 and which is contained in $X_{\tau_\alpha,r}$. Then such a path would have to intersect the hyperplane β passing through q and orthogonal to line α. To see why, we first note that the complement $\mathbb{R}^d \setminus \beta$ of the hyperplane is disconnected with two connected components, each containing one of the two points p_1 and p_2. If the path would not intersect the hyperplane, it would be contained in the complement of the hyperplane, but not entirely contained in one of the connected components, which yields a contradiction to the path being connected. By construction, this point of intersection $z \in X$ lies on the path between p_1 and p_2, and therefore $z \in X_{\tau_\alpha,r}$, or equivalently, $\tau_\alpha(z) = d(z, \alpha) \leq r$. Since z is also contained in the hyperplane β orthogonal to the line α and passing through q, we have that $d(z, q) = d(z, \alpha) \leq r$, i.e., $z \in B(q, r) \subset \mathbb{R}^d \setminus X$, which is a contradiction to $z \in X$. Therefore, for any

[1] Namely, there exists a simplicial complex K and a homeomorphism $h\colon |K| \to X$ from the geometric realization of K to X.

[2] We note that while in this proof we need to consider singular homology, when computing PH in practice one works with either simplicial or cubical homology (Remark 4). For the types of spaces that we consider in our work, all homology theories are equivalent. See [172] for a discussion of the equivalence between simplicial and cubical homology, and [153, Chapter 2] for the equivalence of singular and simplicial homology.

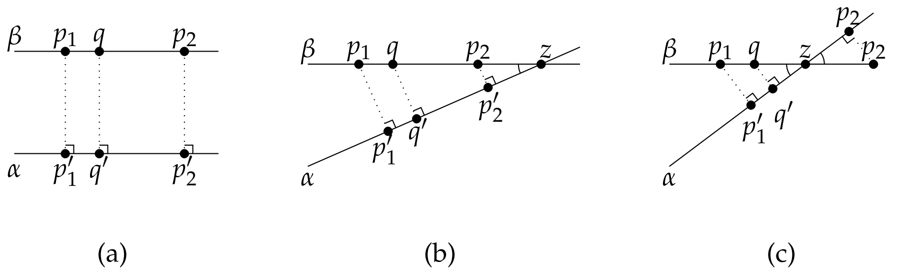


Figure A.1.: The distance $d(p, \alpha)$ from a point p to a line α is defined as the distance between p and the projection p' of p to line α (denoted with dotted lines in the figure). For any two points $p_1, p_2 \in \mathbb{R}^d$, $q \in \mathbb{R}^d$ on a line segment between p_1 and p_2, and a line α in $\mathbb{R}^d$, we have that the line β passing through p_1 and p_2 is either (a) parallel, or (b)-(c) intersects the line α in a point $z = \beta \cap \alpha$. In the former case, per definition of parallel lines, we have that $d(p_1, \alpha) = d(q, \alpha) = d(p_2, \alpha) = d(\beta, \alpha)$. In the latter cases, we have a similarity of triangles $\triangle p_1 p_1' z$, $\triangle q q' z$ and $\triangle p_2 p_2' z$, since they all have a right angle, and share the angle $\angle(\beta, \alpha)$. Since $d(q, z)$ lies between $d(p_1, z)$ and $d(p_2, z)$, the triangle similarity implies that $d(q, \alpha)$ lies between $d(p_1, \alpha)$ and $d(p_2, \alpha)$.

$0 \leq r \leq \epsilon$, the set $X_{\tau_\alpha, r}$ is not path-connected, so that the 0-dimensional PD on the tubular filtration with respect to α contains at least two intervals.

□

Remark 2 (Relationship with the height filtration). To illustrate the need for the tubular filtration, we discuss how it compares to the height filtration that is well-established in the literature. For a given shape $X \in \mathbb{R}^d$ and a unit vector $v \in S^{d-1}$, the height function $\eta_v : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined via the scalar product, $\eta_v(x) = x \cdot v$. If we consider the hyperplane that is orthogonal to the vector v and passes through the origin $(0, 0, \ldots, 0) \in \mathbb{R}^d$, the sublevel set $X_{\eta_v, r}$ corresponds to the area in X above or in the hyperplane and below or at height $r \in \mathbb{R}$, and the complete area of X below the hyperplane (where the scalar product is negative) (Figure A.3, column 1). Note that it is possible to recenter the shape at the origin, so that the hyperplane does not need to pass through the origin.

We note that 0-dimensional PH with respect to the height filtration can help us detect *some* concavities in $\mathbb{R}^d$, what is the case for panel (a) in Figure A.3, where we clearly see multiple path-connected components in green. However, for shapes in $\mathbb{R}^2$ where a source of concavity is a hole within the shape, such as the annulus-like shape in row 2 of Figure A.3, the sublevel sets with respect to the height filtration will only see a single path-connected component, see panel (d) in Figure A.3. Indeed, there is no unit vector v for which the sublevel set $X_{\eta_v, r} = \{x \in X \mid \eta_v(x) = x \cdot v \leq r\}$ contains

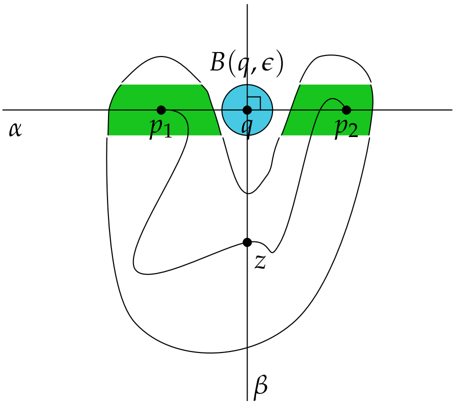


Figure A.2.: For a concave shape X, there exists a tubular filtration line α so that the resulting 0-dimensional PD sees multiple path-connected components (in green). Note that the path in the figure can exist, but it cannot lie completely in the particular sublevel set (in green).

more than one path-connected component. If we adjust the definition of the filtration function and let $\eta'_v(x) = |x \cdot v|$, the sublevel set $X_{\eta'_v,r}$ corresponds to the area in X above and below the hyperplane, and within height $r \in \mathbb{R}$ in both directions. In other words, $X_{\eta'_v,r}$ is the area of X within a given distance from the hyperplane. 0-dimensional PH with respect to the absolute height function η'_v enables us to detect any concavity in $\mathbb{R}^2$, see (b) and (e) in Figure A.3, since the sublevel set consists of multiple path-connected components (in green). In $\mathbb{R}^2$, the tubular filtration corresponds to the absolute height filtration. However, neither the height nor the absolute height filtration can detect concavity in higher dimensions. Indeed, consider a sphere-like shape as an example concave shape in $\mathbb{R}^3$ (Figure A.3, row 3). The sublevel sets $X_{\eta_v,r}$ and $X_{\eta'_v,r}$ will result in a single path-connected component, see the green areas respectively in panels (g) and (h) in Figure A.3. On the other hand, the area within a distance from some *line* (points in X that are within a given tube) can result in two (disconnected) disks on polar opposites on the sphere, see panel (i) in Figure A.3.

However, we note that there are likely alternative approaches that can rely on PH with respect to the (absolute) height filtration to detect concavity. One possibility would be to also consider homology in higher dimensions (although, we note that 0-dimensional PH is faster to compute, see Appendix A.1.4). For shapes in $\mathbb{R}^2$, it is sufficient to consider the 0- and 1-dimensional PH with respect to the height filtration. Indeed, although the annulus-like shape (Figure A.3, row 2) does not see multiple path components with respect to any height filtration function, there is a 1-dimensional hole which points to a concavity. This, however, does not generalize to higher dimensions: Indeed, a sublevel set of a sphere with respect to any height filtration will result in a spherical cap which has trivial homology in dimensions

0, 1 and 2, see panel (g) in Figure A.3. On the other hand, the sublevel set of the absolute height function on the sphere will always consist of a single path-connected component, but we can capture 1-dimensional holes, see panel (h) in Figure A.3. Another interesting example of a concave surface in $\mathbb{R}^3$ is a ball with a dent on the north pole, i.e., a crater. This concavity cannot be detected with 0-dimensional PH with respect to any (absolute) height filtration. For the example surface, an interesting direction would be looking at the surface "from the top" (horizontal hyperplane), but PH would start by seeing a circle, and then the crater itself — always a single path-connected component. However, 1-dimensional PH with respect to this filtration would capture a hole, implying that the surface is concave.

Another possibility could be to study (the computable) multiparameter 0-dimensional PH by scanning shapes from multiple directions *simultaneously*. For the sphere, 0-dimensional PH would capture the two path-connected components with the bi-filtration that looks at the shape with respect to the horizontal hyperplane denoted in the panel (h) in Figure A.3, and the orthogonal hyperplane passing through the shape. We note that while the theory and computations for multiparameter persistence are hard, there have been some recent advances, see, e.g., [332]. This is similar to slicing the shape with a hyperplane, and then studying the single-parameter 0-dimensional PH of the slice. The alternative approaches that we briefly discuss here are an interesting avenue for further work.

Remark 3 (Relationship with the Persistent Homology Transform). There is a lot of work done on studying the so-called Persistent Homology Transform (PHT), which is given by considering the PH on the height filtration with respect to all unit vectors [88, 136, 339]. Such a topological summary that has been shown to be a sufficient statistic for probability densities on the space of triangulable subspaces of $\mathbb{R}^2$ and $\mathbb{R}^3$ [339].

For practical purposes it would not be feasible to have to consider all unit vectors in the PHT. Luckily, there are known results on the sufficient number of directions [88]. In computational experiments in [339] on the MPEG-7 silhouette database of simulated shapes in $\mathbb{R}^2$ [196, 315] and point clouds in $\mathbb{R}^3$ obtained from micro-CT scans of heel bones [41], the PHT is approximated by looking respectively at 64 evenly spaced directions and 162 directions constructed by subdividing an icosahedron. Furthermore, 0- and/or 1-dimensional PH with respect to the height and/or related radial filtration has also been used as a 3-dimensional shape descriptor in [58], for analysis of brain artery trees [29], or classification of MNIST images of handwritten digits [129].

We note that the theoretical results related to the Persistent Homology Transform focus on a complete description of shapes, whereas here we are interested in investigating to what extent PH can detect convexity and concavity.

height	absolute height	tubular
scalar product $\eta_v(x) = x \cdot v$	distance from hyperplane $\eta'_v(x) = \lvert x \cdot v \rvert = d(x, \alpha)$	distance from line $\tau_\alpha(x) = d(x, \alpha)$

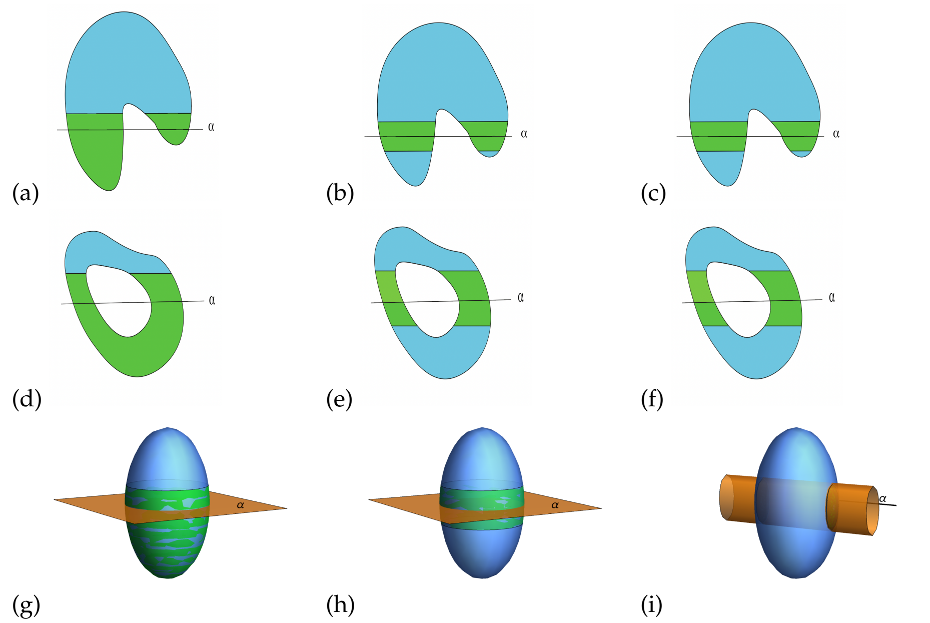


Figure A.3.: Sublevel sets of the height (column 1), absolute height (column 2) and tubular function (column 3) for three example concave shapes in $\mathbb{R}^2$ and $\mathbb{R}^3$. Concavity is detected with multiple 0-dimensional persistence intervals, which reflect the multiple path-connected components in the filtration. The sublevel set (in green) has multiple path-connected components for the height function only for the shape in row 1, for the absolute height only for shapes in rows 1 and 2, whereas for the tubular function this is the case for any concave shape in $\mathbb{R}^d$.

Remark 4 (Detection of convexity with PH in practice). To calculate PH in practice, the sublevel sets K_r need to be approximated with simplicial or cubical complexes (see Section 1.2). The PH pipeline in our computational experiments for convexity detection (Section 2.4) relies on cubical complexes, but it is possible to do so also with the Vietoris-Rips complex relying on geodesic distances. For details, see Appendix A.5.1, where we conclude that it is important to ensure that the singular homology of each of the sublevel sets is properly reflected with the homology of the complex. For convexity detection, this means that the complexes of concave shapes should also see multiple connected components (that do not connect with a simplex).

Furthermore, we note that, to simply differentiate between convex and concave shapes (binary classification problem), it would be sufficient to only consider 0-dimensional *homology* of the intersection of X with the line α, i.e.,

$$X_{\tau_\alpha,0} = \{x \in X \mid \tau_\alpha(x) = d(x,\alpha) \leq 0\} = X \cap \alpha,$$

which is easier to compute than the multi-scale PH. This intersection is a line segment for convex shapes, and for concave shapes it is a union of disconnected segments on a line. Therefore, convex shapes will have $\beta_0(X \cap \alpha) = 1$, and concave shapes will have $\beta_0(X \cap \alpha) > 0$ (one could think of this as persistence intervals $[0, +\infty)$ that all have the same lifespan). However, in practical applications, it is often useful to capture a more detailed information about concavity. For example, for the plant morphology application we consider in Appendix A.7, the goal is to capture a continuous measure of concavity (regression problem). Then, the (different) lifespans of the second (or also third, fourth, ...) most-persisting connected component can provide important additional information. Moreover, in applications X is typically finite, e.g., a point cloud, so that one would still need to approximate $X \cap \alpha$ (points on a line segment) with a complex in order to calculate the homology (see paragraph above). In other words, we would need to choose an appropriate scale $r \in \mathbb{R}$ that would ensure that the complex faithfully reflects the homology of $X \cap \alpha$, which is a non-trivial task.

A.1.2 *Number of holes*

In our pipeline to detect the number of holes, we use the alpha complex, for which several theoretical guarantees have been proven.

The Nerve Lemma (see, e.g., [112]) guarantees that the alpha complex of a set of points has the same homology-type as the space obtained by taking unions of balls of a certain radius centered around the points. Whether this union of balls has the same homology-type as the space from which the points are sampled depends on properties of the sample. If the sample is dense enough, then it has been shown that, for a suitable value of the scaling parameter, the alpha complex has the same number of holes as the original space, for instance under the assumption on the space being a smooth manifold [255]. For ease of reference, we reproduce here the result from [255].

Theorem A.1 (Number of holes)**.** *Let M be a compact smooth manifold, and X a set of points sampled uniformly at random from M. Then there exists $r \in \mathbb{R}$ such that the homology of the alpha simplicial complex $\alpha(X, r)$ is isomorphic to the singular homology of the underlying manifold M.*

Proof. [255, Theorem 3.1] implies that there exists $r \in \mathbb{R}$ such that the singular homology of the $\cup_{x \in X} B(x, r)$ is isomorphic to the singular homology of M. By the

Nerve Lemma we then know that the simplicial homology of the alpha complex $\alpha(X,r)$ is isomorphic to the singular homology of $\cup_{x\in X}B(x,r)$. □

The alpha complex is known to approximate the Vietoris–Rips complex, in the sense that the respective persistence modules are interleaved, see, e.g. [112].

A.1.3 *Curvature*

Here we reproduce the theoretical guarantee provided in [46].

Theorem A.2 (Curvature)**.** *Let M be a manifold with constant curvature κ, and D_κ be a unit disk on M. Let further X be a point cloud sampled from X, according to the probability measure proportional to the surface area measure. Then, PH of X recovers κ.*

Proof. Given κ, [46, Theorem 14] establishes an analytic expression for the persistence ($p = d/b$) of triangles to the curvature κ of the underlying manifold. This function is continuous and increasing, so that persistence recovers curvature. □

A.1.4 *Computational complexity*

In this section, we discuss how our pipelines are affected by the size n of the point cloud and the dimension d of the embedding space (which are also related, since typically exponentially more points are needed to properly sample a shape in higher dimensions).

There exist several efficient algorithms for the computation of PH, many coming with heuristic guarantees on speed-ups for the computation (see survey [258] for an overview, and references therein). For the purposes of this discussion, we will focus on the standard algorithm, which has a computational complexity which is cubical in the number N of simplices contained in the filtered complex [384], i.e., the computational complexity is $\mathcal{O}(N^3)$. Thus, to better understand how our pipelines generalize to higher-dimensional point clouds, in the following we explain how the sizes N of the different types of complexes that we consider are affected by the size n and dimension d of the point cloud.

NUMBER OF HOLES For the detection of the number of holes, in our experiments we relied on alpha simplicial complex. In the worst case, the size N of the alpha complex is $\mathcal{O}\left(n^{\lceil d/2 \rceil}\right)$.

CURVATURE We used the Vietoris-Rips simplicial complex for curvature detection. The size N of the Vietoris–Rips complex is $\mathcal{O}\left(n^{k+2}\right)$, where k is the maximum PH degree that we are interested in computing.

In general, the choice between the alpha and Vietoris-Rips simplicial complex therefore depends on the point cloud size n and dimension d, and the highest homological dimension k that is of interest for the given problem at hand. For point clouds with a given number n of points, the alpha complex is better suited in lower dimensions ($d = 2, 3$), and provided that the point cloud is embedded in Euclidean space.

CONVEXITY To detect convexity, we relied on cubical complexes. For data sets that have an inherently cubical structure, using cubical complexes may yield significant improvements in both memory and runtime efficiency [351]. This is particularly true for high dimensional data, since the ratio between the size of the Vietoris-Rips simplicial complex compared to a cubical complex is exponential in dimension d [71].

In our construction, given a point cloud in $\mathbb{R}^d$ and a fixed $c \in \mathbb{N}$, we bin the points into c^d cubes of the same volume. In the worst case, the size N of the cubical complex of the resulting structure is $\mathcal{O}(3^d c^d)$ (and thus does not depend on the number n of point cloud points).

In addition, it is important to note that our convexity detection pipelines only uses the 0-dimensional persistent homology, which has a reduced complexity since one only needs to construct the complex up to dimension 1. It is fairly easy to compute 0-dimensional PH in near-linear time with respect to the number N of simplices by using union-find data structures [101, 112, 351]. For this reason, it is an important advantage of our pipeline that it only relies on 0-dimensional PH, without needing to calculate PH in higher homological dimensions.

Detecting convexity, however, poses additional challenges. Testing convexity is fundamentally a hard problem in high dimensions, related to the hardness of computing convex hulls in high dimensions, and unfortunately we cannot hope for free lunch. In our PH convexity detection pipeline, unlike for the detection of the number of holes or curvature, we calculate PH across multiple tubular-filtration lines, whose number also grows with the dimension d since sufficiently many filtrations need to be considered (and the same would be the case - we would have to consider multiple tubular directions, if we considered simplicial instead of cubical complexes, see Figure A.13). This could be circumvented by considering (quasi-)random lines. To conclude, specifying a desired computation budget and number of filtrations in advance (leading to a corresponding accuracy tradeoff), our PH pipeline can be used to obtain fast estimates of convexity. It can also be used to compute a continuous measure of convexity (as we demonstrate on the real-world FLAVIA data set of leaf images in Appendix A.7), or convexity at a given resolution, depending on the resolution of the filtration, which in some cases may be more useful than the binary label (convex or concave).

A.2 Experimental details

A.2.1 *Reproducibility and computer infrastructure*

The data and code developed for this research are made publicly available at `https://github.com/renata-turkes/turkevs2022on`. All our computations were conducted using a 2.7Ghz vCPU core from a DGX-1 + DGX-2 station.

A.2.2 *Hyperparameter tuning and training procedure for the individual pipelines*

In this section, we provide more details about the pipelines that were compared in the computational experiments:

- SVM on persistent homology features (PH),
- simple machine learning (ML) baseline - SVM on distance matrices,
- fully connected neural network (NN) on distance matrices, and
- PointNet (PointNet) on raw point clouds.

For each pipeline, we list the hyperparameters that were tuned. To ensure a fair comparison of the different approaches, we used the same train and test data across all the pipelines. We used `sklearn GridSearchCV` based on cross validation with 3 folds and random splits, and returned the hyperparameter values that resulted in the highest accuracy for classification problems (Section 2.2 and Section 2.4), or the lowest mean squared error for regression problems (Section 2.3). We also list relevant software and licenses.

PH The general steps to extract PH features are visualized in Figure 1.7. To calculate PH in Section 2.2 and Section 2.4 we use `GUDHI` [331], and in Section 2.3 we use `Ripser` [28, 335], which are persistent homology libraries in Python, available under the MIT (GPL v3) license. For the DTM filtration (Section 1.3.2) in Section 2.2, we choose $m = 0.03$, so that $0.03 \times 1\,000 = 30$ nearest neighbors are used to calculate the filtration function. Grid search is performed to choose the best persistence signature and classifier or regressor as described below.

- In Section 2.2 and Section 2.3, we select between:
  - simple signature of 10 longest lifespans,

  - persistence images with resolution 10×10, bandwidth $\sigma \in \{0.1, 0.5, 1, 10\}$, and weight function $\omega(x, y) \in \{1, y, y^2\}$, and
  - persistence landscapes with resolution of 100, and considering the longest 1, 10 or all persistence intervals.

- We use SVM (`sklearn SVC` and `SVR` for classification and regression respectively) on the PH signature, with the regularization parameter $C \in \{0.001, 1, 100\}$. The latter tunes the trade off between correct classification of training data and maximization of the decision function's margin.

ML In our experiment, the input for the simple machine learning (ML) pipeline is the normalized matrix of pairwise distances between point-cloud points. For a given point cloud $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, the corresponding distance matrix is the $n \times n$ matrix $D \in \mathbb{R}^{n \times n}$, with entries $D_{ij} = d(x_i, x_j)$ corresponding to the Euclidean distance for the detection of the number of holes (Section 2.2) or convexity (Section 2.4), and hyperbolic, Euclidean or spherical distance for curvature detection (Section 2.3). We take the entries above the diagonal flattened into a vector. Since the dimension of distance matrices scales with the square of the number of points, we work with subsamples of 100 distinct random points from each point cloud. Similarly as above, we use cross validation to choose the SVM regularization parameter among $C \in \{0.001, 1, 100\}$.

We note that while a distance matrix can be taken as input to a classifier, it depends on the particular and arbitrary labeling of the points in the point cloud and hence it does not account for the label symmetry of point clouds.

NN The normalized distance matrices are also fed to the multi-layer perceptrons (MLPs). We consider the following hyperparameters:

- depth in $\{1, 2, 3, 4, 5\}$ (only for NN deep),
- layer widths in $\{64, 256, 1\,024, 4\,096\}$,
- learning rate in $\{0.01, 0.001\}$,

that are selected through a grid search, with each parameter setting trained for 2 epochs. We use a soft-max activation function, and cross entropy and mean squared error as loss functions for classification (Section 2.2 and Section 2.4) and regression (Section 2.3) problems, respectively. Batch normalization (with zero momentum) and a drop out (with a rate of 0.5) are applied after every (input or hidden) layer.

POINTNET PointNet [282] is a neural network that takes point clouds as inputs, and is inspired by the invariance of point clouds to permutations and transformations. It incorporates fully-connected MLPs to approximate classification functions, and convolutional layers to capture geometric relationships between features.

In our experiments, we rely on the PointNet model from `keras` [145] under Apache License 2.0. This model implements the architecture from the original PointNet paper [282], which is supplemented with a publicly available code [281], licensed under MIT. We use grid search to tune:

- number of filters in $\{32, 64\}$,

- learning rate in $\{0.01, 0.001\}$.

For each of the problems we consider, the model is trained from scratch using the training set described in the corresponding section. Unlike in the original paper, we do not augment the data during training by randomly rotating the object or jittering position of each point by a Gaussian noise, in order to ensure a fair comparison with the other pipelines.

A.3 Additional experimental details for number of holes

A.3.1 *Data transformations*

To test the noise robustness of the different pipelines, in Section 2.2 we consider the test data consisting of the original point clouds, or point clouds under different transformations (Figure 2.2). A detailed description of the data transformations is given in Table A.1, and the transformations are visualized on an example point cloud in Figure A.4. To define reasonable values for the data transformations, we took inspiration from the affNIST[3] data set of MNIST images under affine transformations.

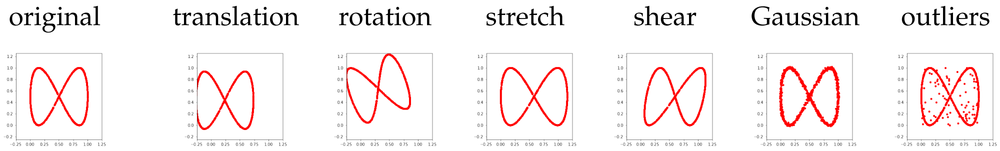


Figure A.4.: Noise on an example point cloud.

[3] `https://www.cs.toronto.edu/~tijmen/affNIST/`

Transformation	Definition of transformation
rotation	Clockwise rotation by an angle chosen uniformly from $[-20, 20]$ degrees clockwise.
translation	Translation by random numbers chosen from $[-1, 1]$ for each direction.
stretch	Scale by a factor chosen uniformly from $[0.8, 1.2]$ in the x-direction. The other coordinates remain unchanged, so that the point cloud is stretched. Stretching factor of 0.8 results in shrinking the point cloud by 20%, and the factor of 1.2 makes it 20% larger.
shear	Shear by a factor chosen uniformly from $[-0.2, 0.2]$. A shearing factor of 1 means that a horizontal line turns into a line at 45 degrees.
Gaussian noise	Random noise drawn from normal distribution $\mathcal{N}(0, \sigma)$ with the standard deviation σ uniformly chosen from $[0, 0.1]$ is added to the point cloud.
outliers	A percentage, chosen uniformly from $[0, 0.1]$, of point cloud points are replaced with points sampled from a uniform distribution within the range of the point cloud.

Table A.1.: Data transformations.

A.3.2 *Pipeline*

Figure A.5 visualizes the PH pipeline. To reduce the computation times, we approximate point clouds at each scale with the alpha simplicial complex (Section 1.2.2, Section A.1.4). The DTM filtration on the point-cloud points is defined as the average distance from a number of nearest neighbors. Therefore, outliers appear only late in the filtration, so that their influence is smoothed out to a great extent. For the example point cloud in the figure, the 1-dimensional PD consists of four persistence intervals with non-negligible persistence or lifespan (PD points far from diagonal) which correspond to the four big holes, and many short persistence intervals that correspond to holes that are seen at some scales due to noise. This is clearly reflected in the vector of sorted lifespans of the 10 most persisting cycles.

A.3.3 *Performance across multiple runs*

Table A.2 provides a detailed overview of the results for the detection of the number of holes, when the experiment is repeated multiple times. The accuracy for PointNet varies for different runs, but in any case, we can clearly see that PH performs the best for each individual run. Note also that the performance of ML, NN shallow and

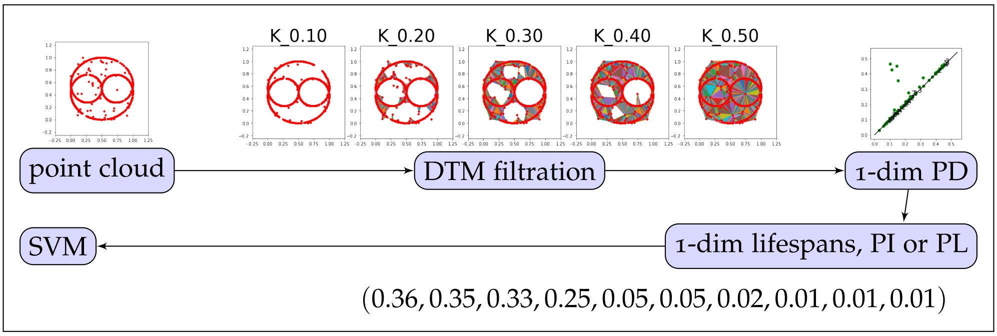


Figure A.5.: Persistent homology pipeline to detect the number of holes.

NN deep does not drop under affine transformations, since they take the normalized distance matrices as input.

A.3.4 *Training curves*

Figure A.6 shows the training curves for the NN and PointNet pipelines. The training set performance of MLPs (shallow and deep) continues improving over epochs, but the validation set performance quickly saturates and stops improving after a few epochs. PointNet performs well on this task, already after a short number of training epochs. We do not include training curves for the PH and ML pipelines, as these are based on SVMs.

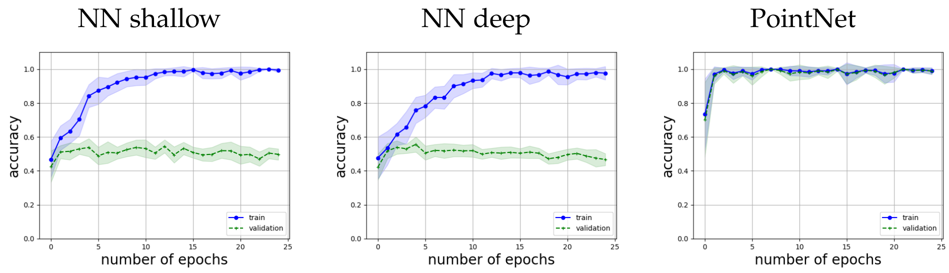


Figure A.6.: Training curves for the detection of the number of holes.

A.3.5 *Learning curves*

Figure A.7 shows the learning curves for every pipeline; i.e., the test accuracy of the trained pipelines depending on the total amount of training data. This serves to evaluate the data efficiency of the different methods. The PH-approaches perform well even for a small number of training point clouds. PointNet also has good performance, although it requires more training data. The other approaches (NN shallow, NN deep, and ML) have poor performance, which does not improve when more training data is available. An explanation for this is that these methods are based on distance matrices and hence cannot directly take advantage of the permutation symmetry of point clouds.

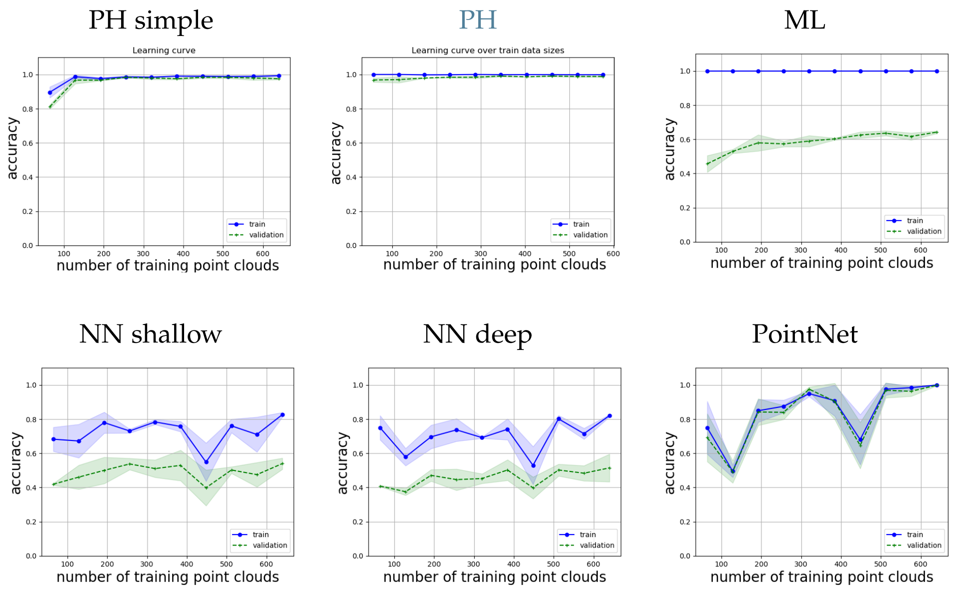


Figure A.7.: Learning curves for the detection of the number of holes.

A.3.6 *Computational resources*

Figure A.8 visualizes the computational efficiency and memory usage. We see that PH pipeline also performs better with respect to these criteria in comparison to the other methods. The hyperparameter tuning of the PH pipeline does take time (as we consider a wide range of parameters for the different persistence signatures), but Figure 2.2 shows that even PH simple, where the SVM is used directly on the lifespans of the 10 most persisting cycles (without any tuning of PH-related parameters) performs well.

We note that the difference in the memory usage for data comes from the different types of input that are considered by different pipelines: PDs (lists of persistence intervals) for PH simple and PH, 100 × 100 distance matrices for ML and NNs, and 1000 × 3 point clouds for PointNet (Appendix A.2.2).

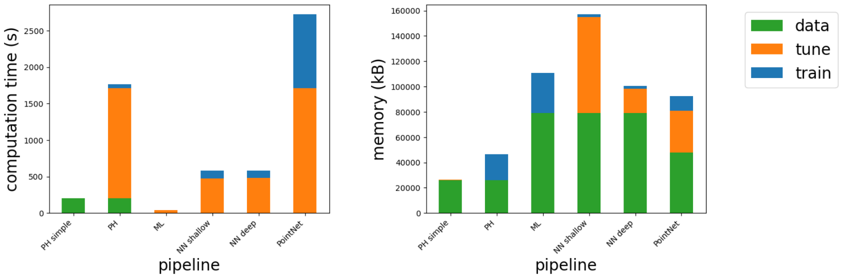


Figure A.8.: Computational resources for the detection of the number of holes.

Transformation	Run	PH simple	PH	ML	NN shallow	NN deep	PointNet
original	run 1	0.94	**1.00**	0.67	0.52	0.50	**1.00**
	run 2	0.94	**1.00**	0.67	0.51	0.50	**1.00**
	run 3	0.94	**1.00**	0.67	0.56	0.50	**1.00**
	mean	0.94	**1.00**	0.67	0.53	0.50	**1.00**
	std dev	0.00	0.00	0.00	0.03	0.00	0.00
translation	run 1	0.94	**1.00**	0.67	0.52	0.50	0.23
	run 2	0.94	**1.00**	0.67	0.51	0.50	0.17
	run 3	0.94	**1.00**	0.67	0.56	0.50	0.21
	mean	0.94	**1.00**	0.67	0.53	0.50	0.20
	std dev	0.00	0.00	0.00	0.03	0.00	0.03
rotation	run 1	0.94	**1.00**	0.67	0.52	0.50	0.86
	run 2	0.94	**1.00**	0.67	0.51	0.50	0.57
	run 3	0.94	**1.00**	0.67	0.56	0.50	0.78
	mean	0.94	**1.00**	0.67	0.53	0.50	0.74
	std dev	0.00	0.00	0.00	0.03	0.00	0.15
stretch	run 1	0.97	**0.98**	0.64	0.49	0.47	0.85
	run 2	0.97	**0.98**	0.64	0.48	0.45	0.70
	run 3	0.97	**0.98**	0.64	0.52	0.51	**0.98**
	mean	0.94	**0.98**	0.64	0.50	0.48	0.84
	std dev	0.00	0.00	0.00	0.02	0.03	0.14
shear	run 1	0.95	**1.00**	0.66	0.54	0.51	0.94
	run 2	0.95	**1.00**	0.66	0.51	0.50	0.72
	run 3	0.95	**1.00**	0.66	0.56	0.51	0.96
	mean	0.95	**1.00**	0.66	0.54	0.51	0.87
	std dev	0.00	0.00	0.00	0.02	0.01	0.13
Gaussian	run 1	0.94	**1.00**	0.68	0.54	0.50	**1.00**
	run 2	0.94	**1.00**	0.68	0.51	0.50	**1.00**
	run 3	0.94	**1.00**	0.68	0.56	0.51	**1.00**
	mean	0.94	**1.00**	0.68	0.54	0.50	**1.00**
	std dev	0.00	0.00	0.00	0.02	0.01	0.00
outliers	run 1	0.82	**0.93**	0.62	0.55	0.49	0.70
	run 2	0.82	**0.93**	0.62	0.51	0.50	0.51
	run 3	0.82	**0.93**	0.62	0.54	0.41	0.44
	mean	0.82	**0.93**	0.62	0.53	0.47	0.55
	std dev	0.00	0.00	0.00	0.02	0.05	0.13

Table A.2.: Accuracy across multiple runs for the detection of the number of holes.

A.4 Additional experimental details for curvature

A.4.1 *Pipeline*

Before we visualize the PH pipeline, we give an illustrative figure that provides some intuition on why PH can detect curvature (Figure A.9).

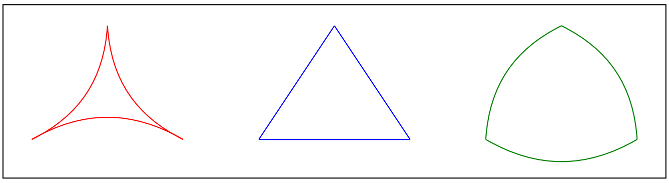


Figure A.9.: Intuition behind curvature detection with PH. For triangles embedded on a manifold with constant negative (left, in red), zero (middle, in blue) and positive (right, in green) curvature, the length of triangle edges clearly reflect the underlying curvature. Since persistence captures the length of these edges (when the triangle vertices merge into a component), PH can be used to detect curvature.

The PH pipeline to detect curvature (Section 2.3) is visualized in Figure A.10. The example point cloud shown in the figure is in the Euclidean plane. We start by calculating the Euclidean distance matrix, and then construct the Vietoris-Rips filtration from these distances, which approximates the point cloud at different scales. 0-dimensional PD registers one persisting cycle reflecting the single component of the disk (which we ignore, since it is shared by every disk in the data and thus does not contribute to the classification), and many other components which have a short lifespan as they get connected to other point-cloud points early in the filtration. There are no persistent 1-dimensional holes since disks are contractible, but there are many holes with short persistence. PDs are then transformed to a vector summary such as a PI.

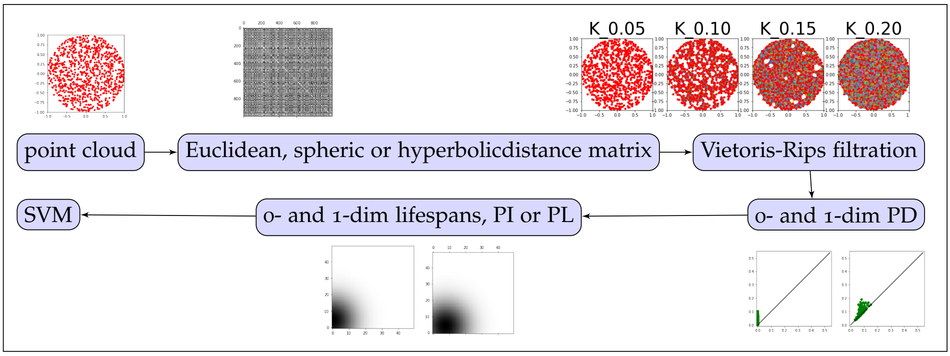

K_0.05
K_0.10
K_0.15
K_0.20
point cloud
Euclidean, spheric or hyperbolicdistance matrix
Vietoris-Rips filtration
o- and 1-dim PD
o- and 1-dim lifespans, PI or PL
SVM

Figure A.10.: Persistent homology pipeline to detect curvature.

Run	0-dim PH simple	0-dim PH simple 10	0-dim PH	1-dim PH simple	1-dim PH simple 10	1-dim PH	ML	NN shallow	NN deep	PointNet
run 1	**0.06**	0.21	0.08	0.34	0.29	0.18	0.34	0.42	0.46	12.28
run 2	**0.06**	0.21	0.08	0.34	0.29	0.18	0.34	0.43	0.46	0.25
run 3	**0.06**	0.21	0.08	0.34	0.29	0.18	0.34	0.66	0.43	578.28
mean	**0.06**	0.21	0.08	0.34	0.29	0.18	0.34	0.50	0.45	196.94
std dev	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.02	330.31

Table A.3.: Mean squared error across multiple runs for curvature detection.

A.4.2 *Performance across multiple runs*

Table A.3 shows that (0-dimensional) PH outperforms the other machine- and deep-learning approaches for curvature detection, across multiple experimental runs. The poor performance of PointNet is not surprising, as it takes raw point clouds as input, i.e., their coordinates and Euclidean distances, and is therefore the only pipeline that does not exploit the spherical and hyperbolic distances for point clouds lying on manifolds with positive or negative curvature (for details, see Appendix A.2.2).

A.4.3 *Computational resources*

Figure A.11 visualizes the computational time and memory usage of the different pipelines for this task. The superior performance of the PH pipelines in comparison to other methods (Figure 2.4) can come at a high cost with respect to the usage of computational resources. However, the simple 0-dim PH pipeline (that only focuses on the lifespans of the PH cycles), which achieves the best predictive power (Figure 2.4), is the most efficient.

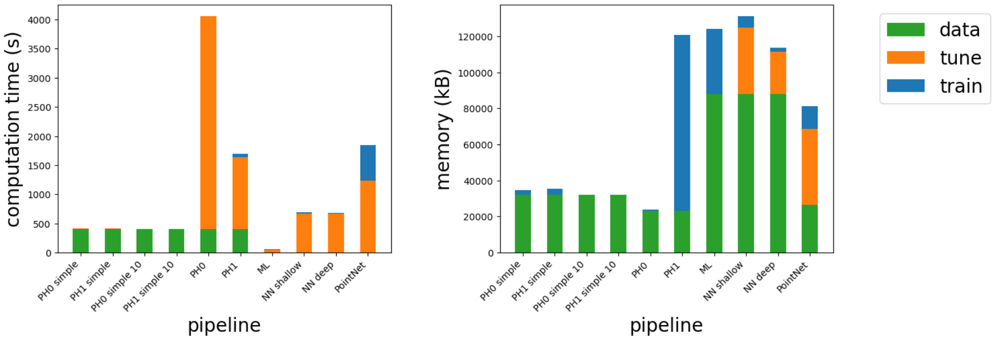


Figure A.11.: Computational resources for the detection of curvature.

A.5 Additional experimental details for convexity

A.5.1 *Pipeline*

A visual summary of the PH pipeline used for convexity detection (Section 2.4) is given in Figure A.12. Every shape has at least one 0-dimensional cycle, i.e., connected component. For the given example point cloud, PD on the cubical complexes weighted by the tubular filtration from the line passing through the bottom of the image will have a second persistent connected component. A positive persistence of the second most persisting cycle for at least some line indicates concavity.

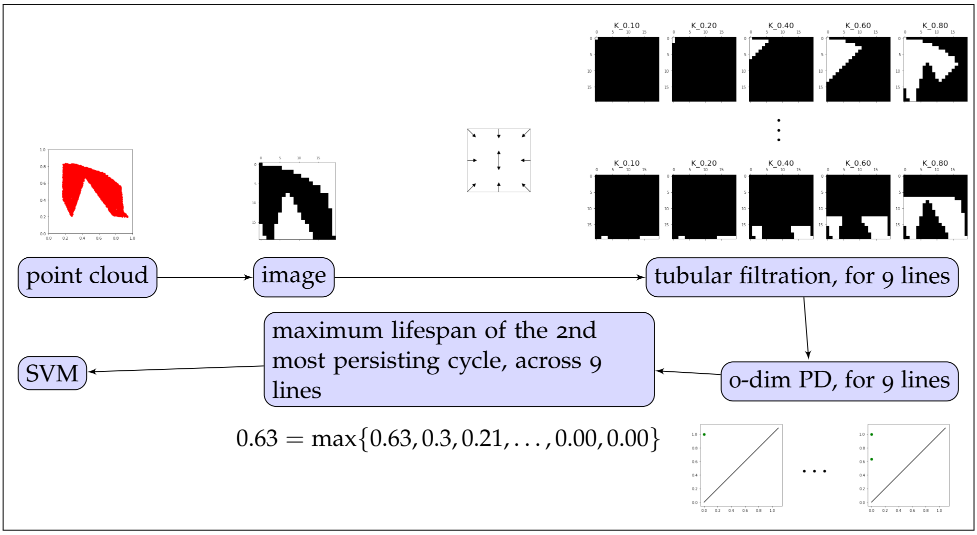

K_0.10
K_0.20
K_0.40
K_0.60
K_0.80
⋮
point cloud
image
tubular filtration, for 9 lines
maximum lifespan of the 2nd most persisting cycle, across 9 lines
0-dim PD, for 9 lines
SVM
0.63 = max{0.63, 0.3, 0.21, . . . , 0.00, 0.00}
. . .

Figure A.12.: Persistent homology pipeline to detect convexity.

We note here that the convexity could also be detected with PH with respect to the Vietoris-Rips filtration, with some important adjustments. Indeed, [68, Theorem

2] provides a guarantee that PH of any function f and shape M can be estimated using an algebraic construction based on Rips complexes from a point cloud X which is a geodesic dense-enough sample of M (and Theorems 3 and 4 in this paper obtain guarantees in scenarios where both function values and pairwise distances are approximate, i.e., defined on the point cloud). To do so to detect convexity, we cannot employ the standard (so-called vanilla) Vietoris-Rips simplicial complex that relies on the distance function, since all point cloud points show immediately at $b = 0$ in the filtration (that all soon get connected into a single component), so that it never sees the two connected components in concave shapes, at any scale $r \in \mathbb{R}$, which are captured with cubical complexes. Filtering the point cloud points by their height (yielding a so called weighted Rips filtration) might capture the multiple connected components, but these components can get connected with an edge as soon as they are born, if they are close to each other with respect to Euclidean distance (Figure A.13). This can be resolved by considering the geodesic distance (the length of the shortest path along the manifold, or a graph), which will allow the multiple connected components to persist longer in the filtration. Indeed, the geodesic distance between points in the "disconnected" regions in concave shapes (the clusters) is larger than the Euclidean distance, so that these only get connected later in the filtration.

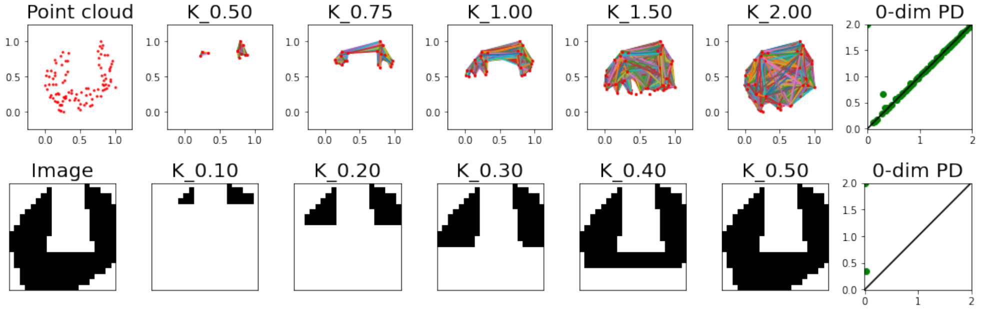


Figure A.13.: Convexity detection with PH on simplicial and cubical complexes. The concavity can be detected with the weighted Vietoris-Rips simplicial complex, with the tubular filtration function on the vertices (top row). In this figure, we consider the tubular function with respect to the horizontal line at the top of the point cloud. The filtration function on edges is defined according to the Euclidean distances, but in a way that ensures that an edge can only appear in the filtration after both vertices incident to this edge appear in the filtration (for details, see [10]). However, these multiple connected components can still be connected with an edge, if they are close in the Euclidean space. This could be circumvented by considering the weighted Vietoris-Rips which relies on the geodesic distances (which are expensive to compute), or by considering cubical complexes instead (bottom row), where the connected components remain separate until they merge with the rest of the shape.

Experimental setting	**Run**	**PH**	**ML**	**NN shallow**	**NN deep**	**PointNet**
train = regular test = regular	run 1	**1.00**	0.74	0.72	0.72	**1.00**
	run 2	**1.00**	0.74	0.77	0.60	**1.00**
	run 3	**1.00**	0.74	0.75	0.75	**1.00**
	mean	**1.00**	0.74	0.75	0.69	**1.00**
	std dev	0.00	0.00	0.02	0.08	0.00
train = random test = random	run 1	**0.85**	0.56	0.60	0.46	0.40
	run 2	**0.85**	0.56	0.55	0.52	0.61
	run 3	**0.85**	0.56	0.54	0.59	0.71
	mean	**0.85**	0.56	0.56	0.52	0.57
	std dev	0.00	0.00	0.03	0.06	0.16
train = regular test = random	run 1	**0.78**	0.59	0.59	0.59	0.51
	run 2	**0.78**	0.59	0.56	0.60	0.47
	run 3	**0.78**	0.59	0.57	0.57	0.57
	mean	**0.78**	0.59	0.57	0.59	0.52
	std dev	0.00	0.00	0.02	0.08	0.05
train = random test = regular	run 1	**0.96**	0.54	0.55	0.49	0.54
	run 2	**0.96**	0.54	0.54	0.42	0.57
	run 3	**0.96**	0.54	0.56	0.52	0.52
	mean	**0.96**	0.54	0.55	0.48	0.54
	std dev	0.00	0.00	0.01	0.05	0.02

Table A.4.: Accuracy across multiple runs for convexity detection.

The important thing to keep in mind is to choose a filtration that will see disconnected components for concave shapes. We choose cubical complexes as they are more straightforward and do not involve the calculation of geodesic distances. Indeed, as the authors of [68] note, geodesic distances are not known in advance and have to be estimated through some neighborhood graph distance, and computing full pairwise geodesic distances is expensive [188, 234] (e.g., there are deep learning efforts to estimate these geodesic distances on point clouds, such as [155, 280]).

Finally, we also note that the calculation of geodesic distances for curvature detection (Section 2.3) was straightforward since the point clouds were sampled from unit disks from manifolds with constant curvature, which enabled us to directly rely on the analytical formulas for geodesic distance.

A.5.2 *Performance across multiple runs*

For any experimental run, PH is better able to distinguish between convex and concave shapes than the other machine- and deep-learning pipelines (Table A.4).

A.5.3 *Computational resources*

Results related to the computational efficiency of the different approaches (trained on regular, and tested on regular shapes) are summarized in Figure A.14. In this case, the PH pipeline significantly outperforms the other methods, since it relies on a scalar summary of a point cloud (the maximum lifespan of the second most persisting connected component, across 9 tubular filtration function lines, see Section 2.4 and Figure A.12). On the other hand, PointNet relies on raw point clouds and therefore has a very high memory consumption, since point clouds have 5 000 points for this task (compared to 1 000 points for the detection of the number of holes, or 500 points for curvature detection).

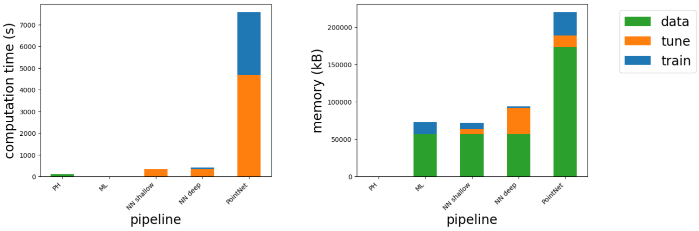


Figure A.14.: Computational resources for the detection of convexity.

A.5.4 *Mislabeled point clouds*

In order to gain a better understanding of the performance and limitations of our PH pipeline, we look at some examples of mislabeled point clouds. Figure A.15 shows a few point clouds sampled from concave shapes that are erroneously classified as convex by PH pipeline (trained on regular, and tested on random shapes). The figure also clearly suggests that considering additional lines for the tubular filtration function would resolve these issues.

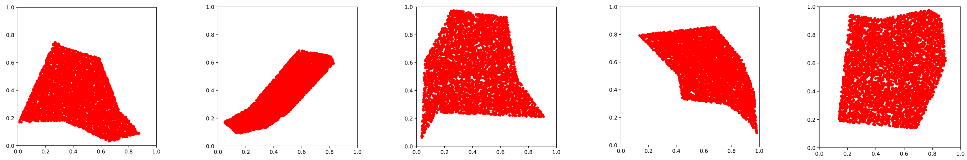


Figure A.15.: Examples of point clouds from concave shapes incorrectly classified as convex.

A.6 Guidelines for persistent homology in applications

Our results demonstrate that PH can be successful in applications for which detecting the number of holes, curvature and convexity is important. Based on our findings, we delineate guidelines for the choice of filtrations and signatures, the input and output of PH pipelines, and draw a better understanding of the topology and geometry properties that are captured by long and short persistence intervals (see Figure A.16).

We again note here that we use the alpha simplicial complex for the detection of number of holes in order to improve the computational efficiency, but that the same can be done with the standard Vietoris-Rips filtration. In addition, we discuss in Appendix A.5.1 that convexity can alternatively be detected with the weighted Vietoris-Rips filtration, filtered by the tubular function, and relying on geodesic distances.

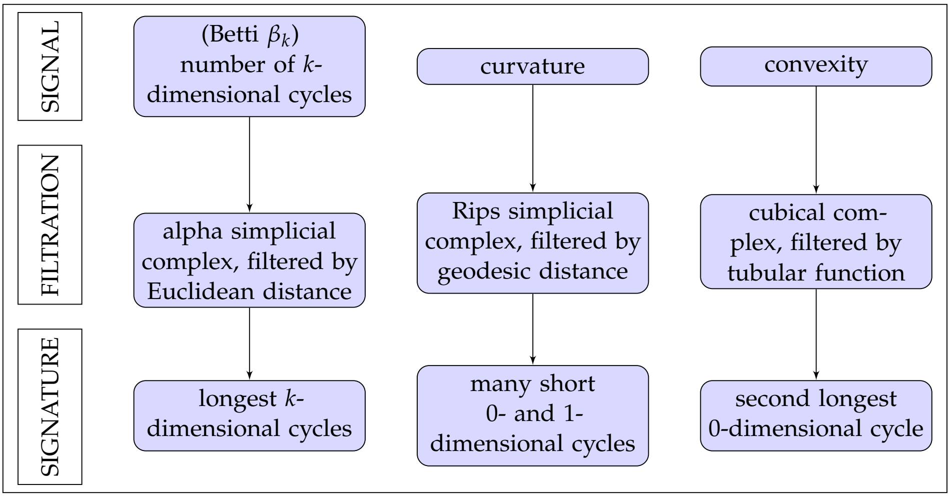

SIGNAL
FILTRATION
SIGNATURE
(Betti β_k) number of k-dimensional cycles
curvature
convexity
alpha simplicial complex, filtered by Euclidean distance
Rips simplicial complex, filtered by geodesic distance
cubical complex, filtered by tubular function
longest k-dimensional cycles
many short 0- and 1-dimensional cycles
second longest 0-dimensional cycle

Figure A.16.: Persistent homology can be useful in applications where k-dimensional cycles, curvature or convexity are important features. The choice of filtration and persistence signature, including the focus on the long and/or short persistence intervals, depends on the signal of the particular application.

A.6.1 *Adjustments of persistent homology pipeline for related applications*

Some obvious adjustments to the guidelines from Figure A.16 can be made for applications related to the ones that we consider here. Some possible adjustments include the following.

- If it is expected that the data set is noisy, the suggested filtration function should be weighted by density to achieve robustness to noise (as in Section 2.2).
- In our experiments, we focused our attention on the PH information relevant to the individual problem at hand, but for other related applications, one might

need to consider a different type of information given by PH. For example, if we do not only aim to distinguish between convex and concave shapes, but rather to capture more information about the possibly many concavities, we should not restrict our attention only to the second most persisting cycle, nor consider the maximum across filtration function directions. Instead, it would be useful to take all PH intervals into account for such an application.

- If there are multiple sources of differences in the data, it can be a good idea to combine the different pipelines. For example, if two classes can be differentiated with some concavities, 0-dimensional PH on the tubular filtration will be useful, but if it is also the shape curvature that can help make a distinction, this information can be concatenated with 0- and 1-dimensional PH on Vietoris-Rips filtration.

A.6.2 *Discussion of persistent homology pipeline for other applications*

step 1: signal Figure A.16 and the discussion above clearly indicate that, when faced with a new problem, it is essential to first try to identify the important information, the signal. To illustrate this more clearly, we list some examples of very different types of signal in Appendix A.6.2.1, Appendix A.6.2.2, Appendix A.6.2.3. Once there is some understanding of the signal, the next steps are to choose the filtration and signature accordingly.

step 2: filtration The aim is for the filtration to capture the signal. For instance, the Vietoris-Rips filtration encodes the size of cycles, while the height or tubular filtration encodes their position. The choice of filtration also influences which type of geometric properties will be captured by long or short persistence intervals. To illustrate the importance of the choice of filtration for the interpretation of long and short intervals, we consider the example point cloud in Figure 1.7. PH with respect to any meaningful filtration can detect the topology of the underlying shape, i.e., the two holes. However, for the height filtration function from the top of the image, the small circle would have a longer lifespan of the two (as it is born earlier in the filtration), and the large circle can have a seemingly very short lifespan (as it is only born at the bottom of the image). For the Vietoris-Rips filtration it is the opposite (a small cycle has short persistence), and PD on the height filtration from the bottom of the image would see cycles of comparable persistence.

step 3: signature The choice of persistence signature and the corresponding metric further influences the emphasis on long or short persistence intervals. The Wasserstein distances [54] between PDs place more importance on long persistence, and the same is true for L_p or l_p distances between other common choices of persistence signatures, with the standard choice of parameters. However, similarly to our discussion above in A.6.1, one might want to focus only on short intervals, e.g., by considering only the intervals with lifespan below a certain threshold (so that the distances would be computed between this simplified PH information). Some alternative ways to prioritize short intervals, or intervals of any persistence, or even birth or death value, is via persistence images with the appropriate choice of the weighing function [2], or stable ranks with appropriate densities [60]. Note, however, that the stability results also depend on the choice of filtration, persistence signature and metric [337]. Finally, we note that there has recently been a lot of effort in trying to train neural networks to learn what the best PH signature is for specific types of applications [56, 241, 304, 330].

In the remainder of this section, we consider a few hypothetical applications to discuss the relevance of signal, filtration and signature that we hope will be useful for practitioners. In particular, the examples highlight that the importance of long and short persistence intervals depend on the particular application domain. In this context, it is sensible to try to understand the nature of information that is captured with PH (e.g., topological or geometric, any of which might or not be important). The examples thus help us to refine an ongoing discussion in the field about the information detected by intervals of a specific length in a PD:

- **Long persistence intervals as signal.** Indeed, this is true in examples from Figure A.17 (when long intervals capture important topology) or Figure A.20 (where long intervals capture important geometry). However, an example in Figure A.21 (together with results in Section 2.3 and Section 2.4) highlights that important information can be encoded in short intervals.

- **Short persistence intervals as noise.** Figure A.21 is an example where important information is captured by short intervals (or in this case, a short interval). This can also be seen in the experimental results in Section 2.3 and Section 2.4.

- **Long intervals capture topology.** An example in Figure A.20 highlights that long intervals, next to topology, also capture geometric information.

- **Many short intervals capture geometry.** An example in Figure A.21 (together with results in Section 2.4) shows that even a single short interval can capture (important) geometry.

A.6.2.1 *Topology is important, geometry is irrelevant*

In some applications, it might be useful to make no distinctions between a circle, a circle with a dent, the circle under translation or scaling, or a square (Figure A.17). In this case, "shape" is understood through the lens of topology — more precisely, what we are interested in is what is called "homology-type" —, where one object can be deformed into another by bending, shrinking and expanding, but not tearing or gluing. Indeed, to a topologist, a coffee mug and a donut have the same shape.

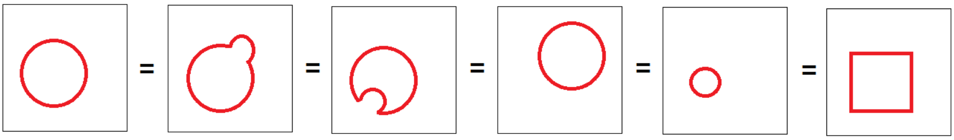


Figure A.17.: An example of an application where topology is the signal. We consider all the shapes to be the same, i.e., to represent the same class of data, as they all have one connected component and one hole.

It is possible to obtain the same PH summaries for all of the shapes from Figure A.17. Indeed, 1-dimensional PDs with respect to the standard Vietoris-Rips filtration on a unit circle and a unit square sampled with same density (reflected in the birth values) could respectively be $\{(0.1, 1)\}$ and $\{(0.1, 1.41)\}$, since the death value reflects the size of the hole. However, we can focus on the cardinality $|\text{PD}|$ of PDs, that here only encodes topological information. Alternatively, we could rather consider PDs calculated on cubical complexes filtered by the binary or grayscale filtration.

Let us further assume that point clouds with multiple holes might be present in the data, but that the only relevant information is the presence of holes, and not their number (Figure A.18). An example of such application could be classification between chaotic and periodic (biological) time series, since the circularity of the so-called Taken's embedding point cloud reflect periodicity of the underlying time series [269]. In this case, we can only focus on the maximum persistence $\max\{l = d - b \mid (b, d) \in \text{PD}\}$. Although PD captures a lot of topology and geometry, this choice of summary obviously ignores a lot of this information.

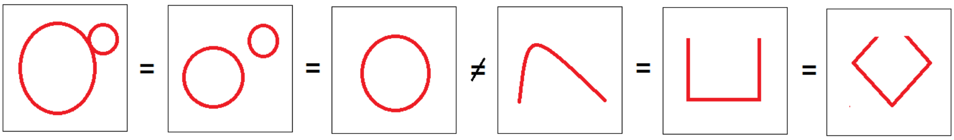


Figure A.18.: An example of an application where the presence of holes is the signal. The first three shapes in the left part of the figure belong to the same class as there is at least one hole present, whereas the remaining three shapes belong to another class with no holes.

A.6.2.2 *Topology is irrelevant, geometry is important*

For other type of applications, the shapes from Figure A.17 might be representatives of different classes of objects (Figure A.19). Since they all have a single connected component and a single hole, the topological information has no use in discriminating between the classes. However, the geometric information about their size and position is useful.

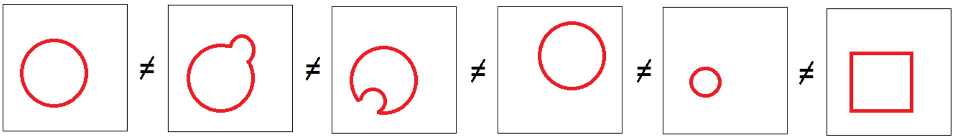


Figure A.19.: An example of an application where geometry is the signal. In this case, every shape in the figure represents a different object, i.e., they all belong to different data classes.

important geometry encoded in long intervals Consider an example where every shape in the data set has only two holes (Figure A.20), and PH with respect to the Vietoris-Rips filtration. The two longest intervals reflect these two holes (topological information), but their lifespans reflect their size, and it is this geometric signal that can help discriminate between the shapes.

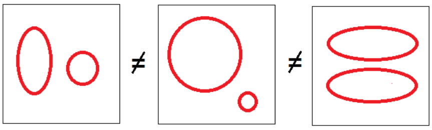


Figure A.20.: An example of an application where the size of the holes is the signal. The three shapes all have two holes, but their size is meaningful for this application, so that they all belong to different data classes.

important geometry encoded in a single interval with the shortest persistence We consider a hypothetical cancer-detection application. Let us assume that the data set consists of medical images of some cells in the human body, which look like a certain number of holes (e.g., a grid-like structure). Now imagine that the only difference between the healthy and cancerous cells is the presence of a tiny hole somewhere in the image (which might correspond to some developing cancerous tissue) (Figure A.21). For PH with the Vietoris-Rips filtration, the lifespan of each cycle registers its size, but it is the very short persistence of the tiniest holes which would be the most important for this application, as it would be this local geometry signal that would allow to discriminate between the two classes of data, i.e., to detect the presence of cancer.

For example, PH for healthy and cancerous cells can respectively have lifespans $(20, 20, 20, 20, 0)$ and $(20, 20, 20, 20, 0.05)$. The stability theorems imply that the difference between the PH on the healthy and cancerous cells is "small" (or more precisely,

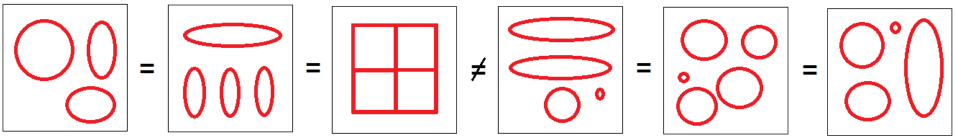


Figure A.21.: An example of an application where the presence of a tiny hole is the signal. The first three shapes in the left part of the figure reflect the images of healthy cells, whereas the remaining three shapes indicate developing cancerous tissue.

it is limited by the difference in their filtrations), but this difference is important for this problem and hence not "noise."

PH can be successful for this task even if the number and size of holes varies across images of healthy cells. In this case, the lifespans for PH of healthy cells could, e.g., be $(20, 15, 12, 25, 0)$, $(13, 21, 15, 17, 0)$, and $(14, 15, 27, 20, 0.05)$, $(19, 21, 15, 17, 0.05)$, for cancerous cells. Here, the distance between the PH for healthy and cancerous cells is overwhelmed by the distance between the long cycles, that reflect irrelevant information for the problem. However, we could consider a PH signature that only focuses on short intervals, or choose PIs that give a greater weight to short intervals. Alternatively, if there is a number of labeled images available, the difference with respect to the short persistence interval can be learned.

In the same way, it might be the case that images can only be distinguished with a hole of medium persistence, and therefore the importance of different lifespans depends on the application, i.e., data set. If we know this a priori, we can use PIs and give the greatest weight to the intervals with the most distinctive persistence.

A.6.2.3 *Topology and geometry are important*

To conclude our guidelines, we consider an example of an application in which both topological and geometric information are important. Let us consider a classification problem where the shapes in Figure A.22 represent different objects, i.e., different data classes. In this case, it is topology and geometry together that provide useful information. The standard choice of PH on the Vietoris-Rips filtration can help to distinguish between these objects. The PH signature should consider all persistence intervals, since, as discussed in Section 2.5, geometry (reflecting the size of cycles) is captured in every persistence interval, while the topology is reflected in the number of long-enough intervals.

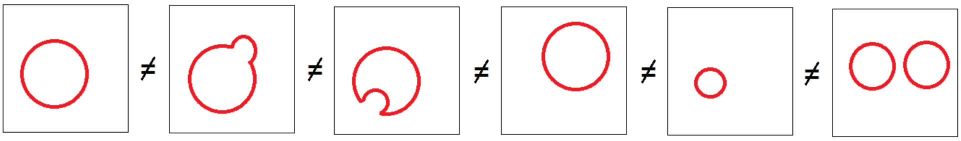


Figure A.22.: An example of an application where topology and geometry are both signal.

A.7 Persistent homology detects convexity in FLAVIA data set

In this section, we employ PH on the FLAVIA data set which consists of 1 907 1200×1600 images of plant leaves [364]. Figure A.23 shows a few examples of images in this data set. The goal of these experiments is to show that PH can be effective on real-world data, but also to illustrate the above guidelines about the appropriate choice of filtration and signature for a given application, and the importance of long and short intervals (Appendix A.6). We focus on convexity detection, as this is the main contribution of our work.

We classify the leaves according to following measure of convexity:

$$c(X) = \frac{\text{area}(X)}{\text{area}(CH(X)))}, \tag{A.1}$$

where $CH(X)$ is the convex hull of image X. The convexity measure (A.1) is the most widely used in the literature [385], and has been shown to be useful for plant species recognition [175]. Note that we only use the above formula to properly label the data set (Figure A.23), but that, deriving convexity information in such a way involves employing a convex hull algorithm.

In the simpler scenario of a binary classification between convex or concave shapes (i.e., signal is the simple: convex - yes or no), we could rely on the same pipeline as in Section 2.4, where we consider the lifespan of the second most persisting connected component, and then store the maximum such value across all 9 tubular filtration lines (Figure A.12, Figure A.24). This is sufficient information, since we are only interested in whether PH sees multiple connected components - source of a concavity, for *at least one* line. However, the convexity measure (A.1), the signal in our application, provides a more detailed level of information (regression problem), so that we keep the lifespan of the second most persisting connected component *for all* lines, in order to capture information about sources of concavities seen with respect to any of the lines.

Moreover, since the convexity measure is calculated relative to the size of the leaf, the tubular filtration *directions* remain the same, but 8 filtration lines pass through the corners of the leaf rather than the corners of the image (Figure A.12), and the lifespans are normalized relative to the area of the leaf (total number of black pixels in the binary image). In this way, PH depicts information about concavities for any line and relative to the leaf size, and it is invariant under translation and scaling. We use 30×30 cubical complexes (on binary images), to capture a higher level of detail for the leaves of different convexity, in comparison with the 20×20 resolution for the cruder differences in our synthetic data set in Section 2.4. Figure A.24 visualizes the tubular filtration for the 9 different lines, and the resulting 0-dimensional PDs, for an example leaf image (bottom image from Figure A.23).

Linear regression on the FLAVIA data set, trained on 70% of random images, with each image represented with the 9-dimensional vector of lifespans of the second most persisting component across all tubular filtration lines, obtains a mean square error of 0.00065. The regression line in Figure A.25 shows that PH is effective in classifying the FLAVIA leaves according to a measure of convexity. The convexity of some thin leaves (such as the image in the third panel in Figure A.23) gets overestimated with our PH pipeline, since concavity is not captured well with our crude resolution, that could easily be improved.

Furthermore, even more detailed information can be captured if the lifespans of the third, fourth, ... most persistent connected component would be kept, because some leaves have more than two sources of concavity for a single line, that result in more than two connected components. For example, the 0-dimensional PD of the image in Figure A.24 has more than two persistence intervals for some tubular filtration lines. The accuracy can thus be improved by considering the lifespans of *all* short intervals (and across all lines), and again, by considering more tubular filtration lines (Section 2.4). This clearly illustrates how the choice of filtration and signature, the input and output of PH, should be guided by the signal in the given application. Moreover, it shows that one short interval can be sufficient for some applications, but that in other cases, many short intervals might store the needed additional level of (geometric) information.

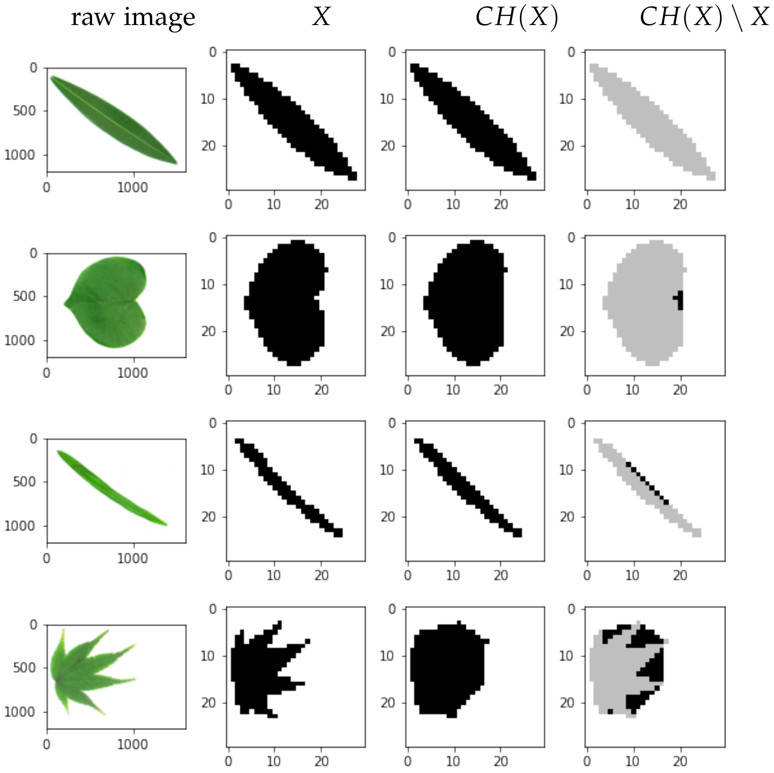


Figure A.23.: A few example images from the FLAVIA leaf data set. The images are shown, from top to bottom, with a decreasing label, i.e., convexity measure $c(X)$, 1.00, 0.98, 0.89, 0.71. Note that the second image from the top is more convex than the third image, since the considered convexity measure $c(X)$ is given relative to the area of the leaf. The PH lifespans on the tubular filtration with respect to 9 different lines are respectively from top to bottom, [0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00], [0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00], [0.00, 0.00, 1.09, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00], [0.00, 2.52, 10.08, 3.78, 0.00, 1.26, 0.00, 0.00, 0.00].

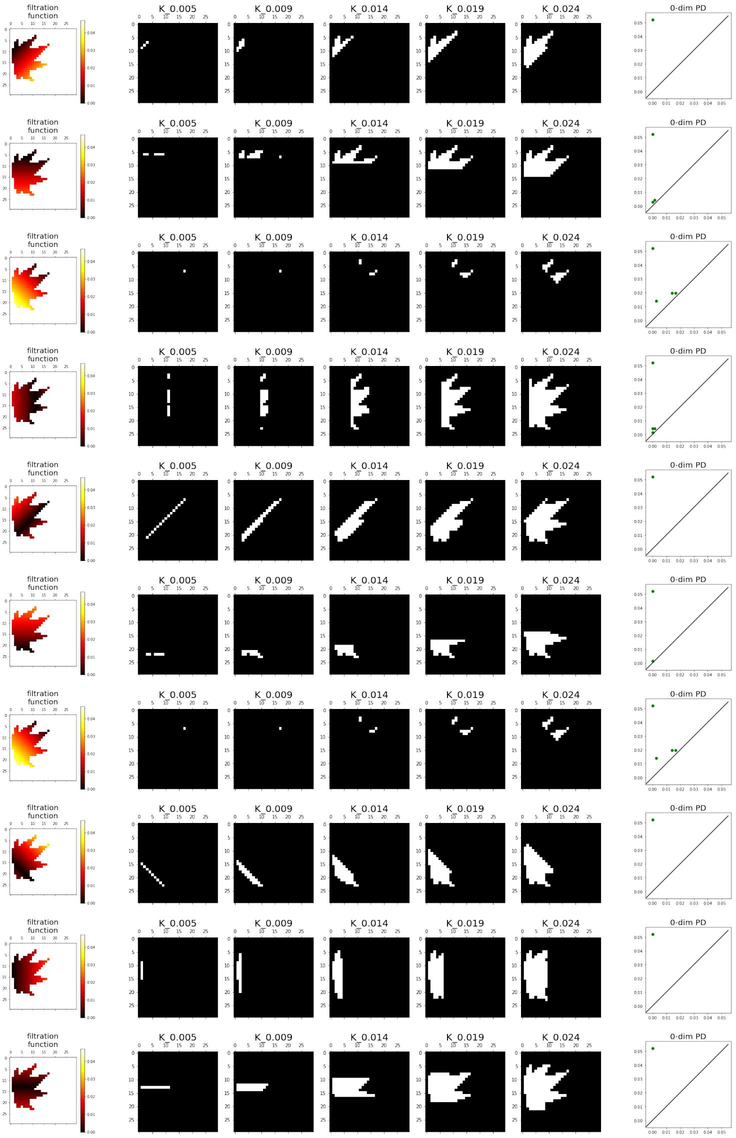


Figure A.24.: The tubular filtration for the 9 considered lines, and the resulting 0-dimensional PDs for an example image. The concavity is detected with multiple connected components that are seen for a few lines.

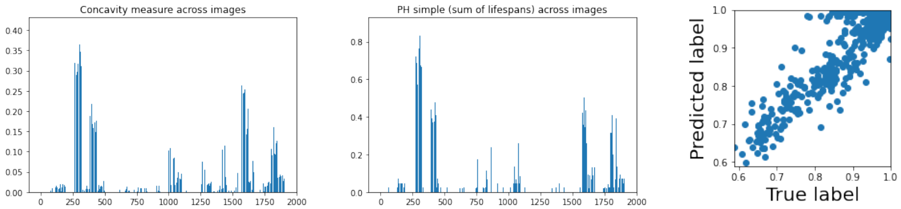


Figure A.25.: Results on the FLAVIA data set. The first two plots show that there is a good correspondence between the concavity measure $(1 - c(X))$ (left panel) and the simple PH signature that only considers the sum of lifespans of the second most persisting connected component, across the tubular filtration lines (middle panel). The regression line on lifespans from all 9 tubular filtration lines shows good performance of our PH pipeline.

B

Pull-back geometry of persistent homology encodings

Empirical studies that investigate the properties of persistent homology (PH) (Chapter 1), such as its sensitivity to perturbations or ability to detect a feature of interest, commonly rely on training and testing an additional model on the basis of the PH representation. To gain more intrinsic insights about PH, independently of the choice of such a model, we propose a novel methodology based on the pull-back geometry that a PH encoding induces on the data manifold. The spectrum and eigenvectors of the induced metric help to identify the most and least significant information captured by PH. Furthermore, the pull-back norm of tangent vectors provides insights about the sensitivity of PH to a given perturbation, or its potential to detect a given feature of interest, and in turn its ability to solve a given classification or regression problem. Experimentally, the insights gained through our methodology align well with the existing knowledge about PH. Moreover, we show that the pull-back norm correlates with the performance on downstream tasks, and can therefore guide the choice of a suitable PH encoding.

This appendix is based on the following publication:

> Shuang Liang, Renata Turkeš, Jiayi Li, Nina Otter, and Guido Montúfar, *Pull-back geometry of persistent homology encodings*, arXiv:2310.07073 (2023).

It is related to Chapter 2 and Chapter 3, as it investigates what kind of features or perturbations can be detected with persistent homology. The project was supported in part by NSF CAREER 2145630, NSF 2212520, DFG SPP 2298 grant 464109215, ERC Starting Grant 757983, and BMBF in DAAD project 57616814.

B.1 Introduction

In applications of PH to data classification or regression tasks, it is common to employ a model, such as a support vector machine (SVM) or a neural network, on the PH features. Such performance-based testing comes with two main drawbacks. Firstly, additional time and effort are needed to choose the classifier or regression model and tune their hyperparameters: this typically involves a grid search over all PH parameters, *but also* over models (e.g., SVM and neural networks), and over the model's parameters (e.g. regularization parameter of SVM, and a much larger list of hyperparameters for neural networks), which also requires training and testing for each combination of the three groups of parameter values. Secondly, the conclusions drawn regarding the effectiveness of PH are contingent upon the choice of the model and its specific parameters.

In this work, we propose a novel methodology aimed at gaining a more intrinsic understanding of PH encodings, irrespective of a particular classification or regression model. Here, a PH *encoding* denotes the mapping from data to a vectorized PH representation. We use the pull-back geometry induced by the PH encoding map to investigate its sensitivity to any particular data variation. A data variation is represented by a vector field in the data space, and can therefore be understood as an umbrella term that includes both perturbation vector fields reflecting data perturbations (e.g., translation or dilation of a point cloud) and gradient vector fields resulting from data features (e.g., a label indicating the presence of an anomaly or disease).

The Jacobian of an encoding mapping characterizes the behavior of the encoding in response to data variations. Specifically, the rank and eigenvectors of the Jacobian characterize the number of independent data variations and the most significant data variations captured by the encoding, respectively. The average pull-back norm of a vector field quantifies the sensitivity of PH to the corresponding data variation, and thus it helps assess to what extent PH is sensitive to a given perturbation or how effective it is at detecting a given feature (which in turns translates to its ability to solve a given problem). Furthermore, optimizing the average pull-back norm can guide the choice of a suitable PH encoding (choice of filtration, PH representation, and their parameters): one only needs to evaluate the pull-back norm over the different choices of PH parameters. This approach eliminates the need to train and select a classifier on top of PH features, at the same time providing insights that are more intrinsic to the underlying problem. Indeed, if the performance of a particular model on PH features is poor, one can hardly make any claims about the *PH representation itself* (since the problem could be that the model is poor). On the other hand, if the pull-back norm of the vector field is close to zero, we are more confident that the representation cannot recognize the given perturbation or feature. We provide a schematic diagram in Figure B.1 that illustrates the pipeline of our proposed method and compares it with performance-based methods.

We center our attention on a widely used PH representation known as the persistence image (PI) (Section 1.4.2), which is an image-like representation of the input data in terms of multiscale homological features. Our methodology, however, extends to other PH representations and, more broadly, to any differentiable encoding whose representation space can be endowed with a Riemannian manifold structure. In our experiments, we illustrate this generality by applying our approach to the PointNet encoding, a benchmark deep learning model for point clouds. We note that the insights about the (PH) encodings obtained through our approach depend on the specifics of the data set. Nonetheless, by evaluating different datasets one may be able to draw certain conclusions that hold with some generality: for instance, conclude that a particular encoding captures a particular feature in datasets of a particular type. This is an interesting prospect that can be facilitated by our proposed approach.

MAIN CONTRIBUTIONS

- We present an approach that can be used to investigate persistence images and their induced pull-back geometry on the manifold of input data sets in terms of the rank, spectrum, and eigenvectors of the Jacobian, as well as the pull-back norm of tangent vectors on the data manifold (Section B.3)[1].

- We show how the above approach can be used to identify which data perturbations are captured by the encodings and which are ignored on given data sets. We also show how this facilitates an intrinsic comparison of PH encodings built with different filtrations. We experimentally demonstrate the insights gained via our approach align well with the existing knowledge about PH (Section B.4).

- We show how the above approach can be used to quantify to what extent a PH encoding can recognize a data feature of interest on given a data set (*sex* feature in a data set of brain artery trees). We also show how this quantitative evaluation can guide the selection of hyperparameters for the encodings. Finally, we show that the pull-back norm is predictive for the performance on a downstream task (Section B.5).

RELATED WORK Our discussion falls within the general subject of interpreting a complex nonlinear map by investigating the effect that local input perturbations have on the output. This is conceptually related to topics such as sensitivity analysis [306], interpretable machine learning [307], sensitivity of outputs to input perturbations [238], activation maximization [318], relevance propagation [239], adversarial robust-

[1] Data and code developed in this research are available at `https://anonymous.4open.science/r/persistent-homology-0915`

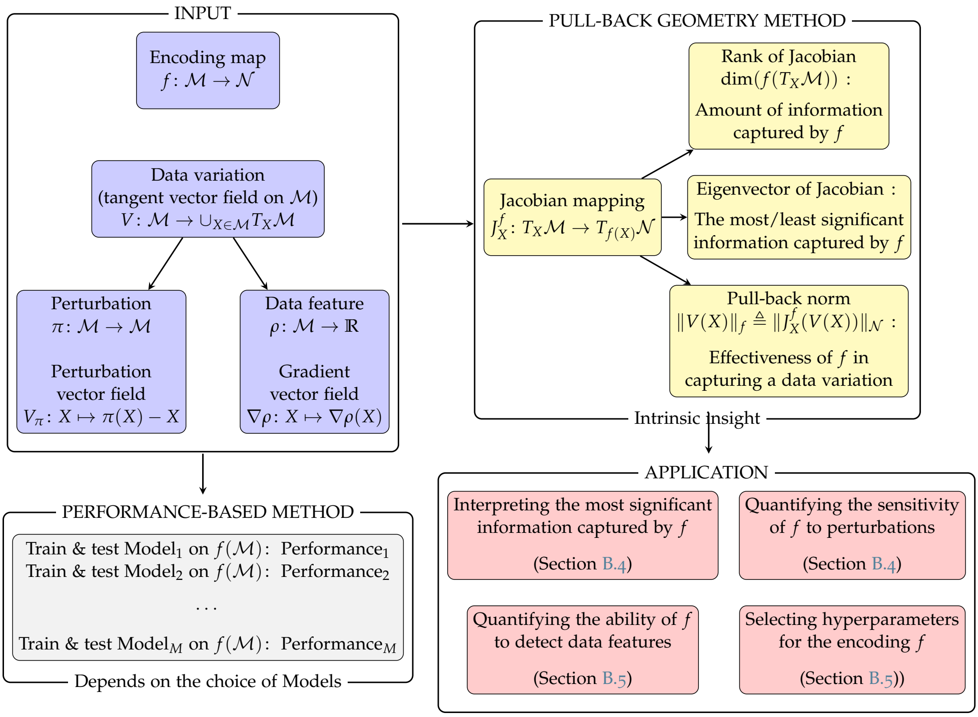

INPUT
Encoding map
$f: \mathcal{M} \to \mathcal{N}$
Data variation
(tangent vector field on $\mathcal{M}$)
$V: \mathcal{M} \to \cup_{X \in \mathcal{M}} T_X\mathcal{M}$
Perturbation
$\pi: \mathcal{M} \to \mathcal{M}$
Perturbation vector field
$V_\pi: X \mapsto \pi(X) - X$
Data feature
$\rho: \mathcal{M} \to \mathbb{R}$
Gradient vector field
$\nabla\rho: X \mapsto \nabla\rho(X)$
PULL-BACK GEOMETRY METHOD
Jacobian mapping
$J_X^f: T_X\mathcal{M} \to T_{f(X)}\mathcal{N}$
Rank of Jacobian
$\dim(f(T_X\mathcal{M}))$:
Amount of information captured by f
Eigenvector of Jacobian :
The most/least significant information captured by f
Pull-back norm
$\|V(X)\|_f \triangleq \|J_X^f(V(X))\|_\mathcal{N}$:
Effectiveness of f in capturing a data variation
Intrinsic insight
PERFORMANCE-BASED METHOD
Train & test Model$_1$ on $f(\mathcal{M})$: Performance$_1$
Train & test Model$_2$ on $f(\mathcal{M})$: Performance$_2$
...
Train & test Model$_M$ on $f(\mathcal{M})$: Performance$_M$
Depends on the choice of Models
APPLICATION
Interpreting the most significant information captured by f
(Section B.4)
Quantifying the sensitivity of f to perturbations
(Section B.4)
Quantifying the ability of f to detect data features
(Section B.5)
Selecting hyperparameters for the encoding f
(Section B.5))

Figure B.1.: Schematic pipeline of our proposed method (comparing it with performance-based testing).

ness [118], interpretable controls in implicit generative models [151], or function parametrizations in artificial neural networks [106, 191].

[154] investigated the Riemannian geometry on the data manifold that is induced by representations learned using artificial neural networks and show, in particular, that the metric tensor can be found by backpropagating the coordinate representations learned by the network. This shares similarities with our approach, as the induced geometry is essentially the pull-back geometry from representation space by the Jacobian map. [228] proposed a graph representation for neural networks using a singular value decomposition of the weight matrices. While they tackle the nonlinearity by studying linear maps contained in the nonlinear map consecutively, our emphasis lies in the local linear approximation of the nonlinear map.

There exist a few studies that investigate the sensitivity of PH representations to perturbations and their ability to recognize specific features, as we do in this work. For example, [337] study the sensitivity of a number of PH representations to different types of transformations (such as rotation, translation, change of image brightness or contrast, as well as Gaussian, salt and pepper noise). However, the main method to assess sensitivity is the performance of an SVM trained on the representations of the original data and tested on the representations of data under transformations, and thus requires training and testing, and depends on the choice of a particular classifier. [46] showed in theory and experiments that persistence landscapes can be used to detect curvature of an underlying set based on a sampled point cloud. [338] conducted investigations towards identifying fundamental types of tasks for which PH representations might be most useful. They showed that beside curvature and number of holes, PH representations can be used for detecting convexity. However, they focus on three specific tasks (detecting number of holes, convexity, and curvature), whereas we study the alignment of the representations with *any* given feature defined on the space of input data sets. Moreover, in that work, the performance of PH is experimentally evaluated via the SVM accuracy, whereas our approach does not require training and testing of any model.

More generally, in the context of inverse problems in persistence theory, there are several lines of work that study conditions under which persistence diagram maps are surjective or injective; see the survey of [259] and references therein. Within this context, our work can be seen as providing a framework related to the study of the injectivity of specific PH encodings. The work of [365] deals with local explainability using topological representations. In contrast, we deal with the representations. [227] measured the dissimilarity between representations learned by neural networks trained on PH encodings and networks trained on raw data. They experimentally demonstrate that networks learn considerably different representations when processing PH encodings instead of raw data. [295] compared the expressivity of PH against the Weisfeiler-Lehman hierarchy of graph isomorphism tests, and explored the potential of PH to capture certain graph structures and characteristic properties. Finally, an important line of work in the study of PH encodings is concerned with developing computable notions of optimal representative cycles for persistent homology classes, see, e.g., the survey of [209].

B.2 Persistent homology encoding

In our work, we will focus on persistence images (PIs). Let PD be a k-dimensional persistence diagram in birth-death coordinates. One converts this to a multiset $\eta(\mathrm{PD})$ in birth-lifespan coordinates by applying the linear map $\eta(b,d) = (b,l)$ with $l = d - b$ to each birth-death pair (b,d). Given a kernel function $g_{(b,l)}(x,y)$ on $\mathbb{R}^2$ and a weighting function $\alpha(b,l)$, the persistent surface is the function $\psi\colon \mathbb{R}^2 \to \mathbb{R}$ defined by

$$\psi(x,y) = \sum_{(b,l)\in\eta(\mathrm{PD})} \alpha(b,l) g_{(b,l)}(x,y).$$

A persistence image (PI) is a finite-dimensional representation of ψ obtained as follows. One splits a subdomain of ψ by a $P \times P$ grid of regions. Then the PI of resolution P is the matrix whose (i,j)-th entry or pixel is the integration value of ψ over the (i,j)-th region. Note that, since the death time d cannot be smaller than the birth time b, the birth-death pairs (b,d) always lie above the diagonal line, i.e.,

$\text{PD} \subseteq \{(x,y) \in \mathbb{R}^2 : y \geq x \geq 0\}$. The transformed birth-lifespan pairs (b,l) lie in the first quadrant, i.e., $\eta(\text{PD}) \subseteq \{(x,y) \in \mathbb{R}^2 : x \geq 0, y \geq 0\}$ (see an illustration in Figure B.2).

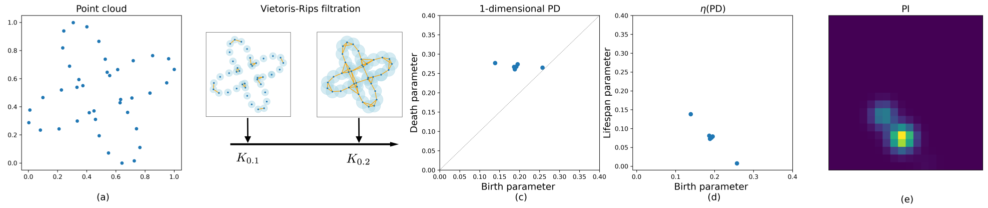


Figure B.2.: The pipeline for constructing a persistence image described in Section B.2. From left to right: (a) input point cloud; (b) Vietoris-Rips filtration built on the point cloud; (c) 1-dimensional persistence diagram; (d) birth-lifespan pairs (transformed 1-dimensional persistence diagram); and (e) persistence image.

For constructing PIs, one needs to choose 1) the resolution P, 2) the kernel function $g_{(b,l)}(x,y)$ and its associated parameters and 3) the weighting function $\alpha(b,l)$. One of the main difficulties in working with PIs is that there is no canonical way to choose these hyperparameters [2]. [2] studied the effects of PI parameters on the performance of certain classifiers (K-medoids classifiers) that take PIs as inputs. However, we note this approach heavily depend on the choice of downstream model. This motivates us to investigate what kind of information of the input data is intrinsically captured by the PI under different choices of the hyperparameters.

A motivation for considering PIs is that they provide differentiable PH representations [see 206], and that, with an appropriate choice of metric, the space of PIs has a Euclidean structure, which simplifies computations (see Section B.3.1). We will later consider derivatives of the mapping from input data to PIs. These can be obtained using existing automatic differentiation packages and libraries, such as topologylayer [128] and Gudhi [331]. We provide further details about this in Appendix B.7.2.

B.3 Pull-back geometry methodology

We consider an encoding map $f \colon \mathcal{M} \to \mathcal{N}$, where $\mathcal{M}$ is a space of point clouds and $\mathcal{N}$ is the space of persistence images (Section B.3.1). We conceptualize input data variations (Section B.3.2) and the resulting changes of the encoding output (Section B.3.3).

B.3.1 *Input and output space*

We let $\mathcal{M}$ be the space of point clouds in $\mathbb{R}^D$ that contain exactly N points,

$$\mathcal{M} = \{X \subset \mathbb{R}^D \colon |X| = N\}.$$

A point cloud X is finite subset in $\mathbb{R}^D$ with cardinality $|X| = N$. A point cloud may be regarded as an unordered list of points, determined only up to permutation. It can also be regarded as a probability distribution on $\mathbb{R}^D$. Hence we can equip $\mathcal{M}$ with the 2-Wasserstein distance [see, e.g. 274]:

$$d_W(X, Y) = \min_{\omega \in \Omega(X,Y)} \left(\sum_{x \in X} d_E^2(x, \omega(x)) \right)^{\frac{1}{2}} .$$

Here $\Omega(X, Y)$ denotes the set of bijections $\omega \colon X \to Y$ between the sets X and Y, and d_E denotes the Euclidean distance on $\mathbb{R}^D$. The 2-Wasserstein distance induces a metric topology on $\mathcal{M}$. Further, $\mathcal{M}$ can be endowed with a Riemannian manifold structure with $\dim(\mathcal{M}) \triangleq m = D \times N$. For simplicity of presentation, in the following we treat $\mathcal{M}$ as an Euclidean space. Nonetheless, our discussion is consistent with the Riemannian manifold structure and applies in that level of generality (see Appendix B.7.1 for details).

We let $\mathcal{N}$ be the space of persistence images of fixed resolution P. Thus we can interpret $\mathcal{N}$ as a submanifold embedded in $\mathbb{R}^{P \times P}$ endowed with the canonical Euclidean distance, with $\dim(\mathcal{N}) \triangleq n = P^2$. [2] Here again, other choices of metric on the space of PIs are possible [3].

B.3.2 *Data variations*

To characterize local variations of the input data, we consider tangent vectors on the data manifold. We conceptualize the intuitive concepts of perturbations and feature variations in terms of corresponding vector fields on the data manifold.

The *tangent space* at $X \in \mathcal{M}$, denoted $T_X\mathcal{M}$, is the vector space of all vectors emanating from X and tangential to the data manifold $\mathcal{M}$. The dimension of $T_X\mathcal{M}$ is equal to the dimension of the data manifold, $\dim(T_X\mathcal{M}) = \dim(\mathcal{M})$. Each *tangent vector* $v \in T_X\mathcal{M}$ characterizes a local variation of a single point cloud X. A *vector field* specifies a variation for each point cloud in $\mathcal{M}$. More specifically, a vector field V on $\mathcal{M}$ is a smooth map $V \colon \mathcal{M} \to \sqcup_X T_X\mathcal{M}$, assigning to each X in $\mathcal{M}$ a tangent vector $V(X) \in T_X\mathcal{M}$.

A *perturbation* is a modification of a point cloud in the data manifold, e.g., by rotation or shearing. This can be described by a map $\pi \colon \mathcal{M} \to \mathcal{M}$ taking data $X \in \mathcal{M}$ to a perturbed data $\pi(X) \in \mathcal{M}$. The perturbation vector field V_π associates to each

[2] We note that if one considers the 1-wasserstein distance on the space of PDs, and any of the L_1, L_2 or L_∞ norms on the space of PIs, then PIs are known to be stable, see [2, Theorem 5]. On the other hand, PIs, together with the L_2 norm, are unstable if one instead considers p-Wasserstein distances on the space of PDs [2, Remark 6].

[3] This can be of interest to try to establish more general stability results for PIs. An example would be a Wasserstein distance between persistence images assigning an appropriate cost to b and l directions.

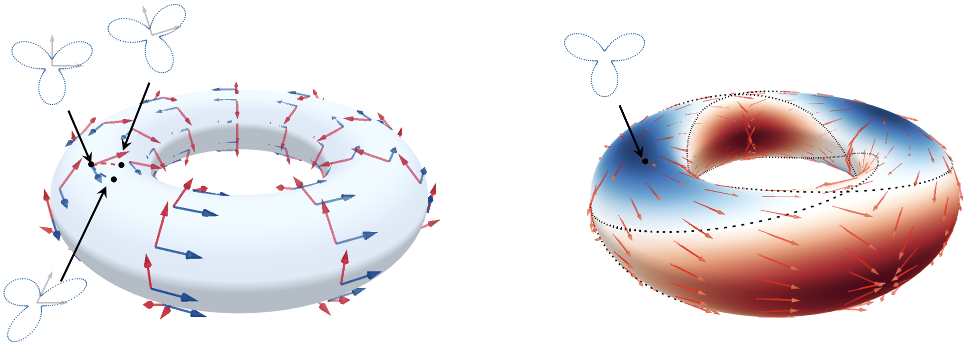


Figure B.3.: The space of point clouds forms a manifold, which in this figure is depicted as a torus; each point on this manifold is a point cloud. Left: vector fields on the data manifold correspond to variations of the point clouds; in this illustration, the red arrows correspond to "rotation" and the blue arrows to "shearing". Right: a continuous feature on the data manifold induces a gradient vector field; the figure illustrates a binary feature, where the dashed line is the class boundary, and the continuous feature value represents the probability of the data point belonging to the "red" class.

$X \in \mathcal{M}$ a tangent vector $V_\pi(X)$ capturing the difference between $\pi(X)$ and X (see Figure B.3, left).[4]

Definition B.1 (Perturbation vector field)**.** Let $\mathcal{M}$ be a manifold, $T_X\mathcal{M}$ the tangent space at $X \in \mathcal{M}$, and $\pi\colon \mathcal{M} \to \mathcal{M}$ a perturbation map. The perturbation vector field induced by π is defined as

$$V_\pi : \mathcal{M} \to \sqcup_X T_X\mathcal{M}; \quad X \mapsto V_\pi(X) = \pi(X) - X.$$

A *feature* ρ is a real-valued smooth function defined on the data manifold, $\rho\colon \mathcal{M} \to \mathbb{R}$, assigning a feature value to each X. Discrete-valued (categorical) features can be converted to continuous ones by considering probability distributions or logits of the feature values. For instance, the "cat-or-dog" feature can be converted to a continuous feature $\rho(X) = \text{Prob}(X \text{ is cat}) \in [0, 1]$.

The gradient of a feature introduces a vector field on the data manifold. The gradient vectors point in the direction of steepest increase of the feature, with magnitude indicating the rate (see Figure B.3, right).

[4] Figure B.3 is a schematic illustration. It is not intended to imply that the kind of depicted point clouds indeed form a torus.

Definition B.2 (Gradient vector field)**.** Let $\mathcal{M}$ be a manifold, $T_X\mathcal{M}$ the tangent space at $X \in \mathcal{M}$, and $\rho\colon \mathcal{M} \to \mathbb{R}$ a real-valued feature. The gradient vector field of ρ is the vector field on $\mathcal{M}$ defined as

$$\nabla\rho : \mathcal{M} \to \sqcup_X T_X\mathcal{M}; \quad X \mapsto \nabla\rho(X),$$

such that $\frac{\nabla\rho(X)}{\|\nabla\rho(X)\|} = \operatorname{argmax}_{v \in T_X\mathcal{M}:\|v\|=1} |\frac{\partial}{\partial v}\rho(X)|$ and $\|\nabla\rho(X)\| = \max_{v \in T_X\mathcal{M}:\|v\|=1} |\frac{\partial}{\partial v}\rho(X)|$. Here $\frac{\partial}{\partial v}$ is the directional derivative along v.

Definition B.1 and Definition B.2 are given for the case that $\mathcal{M}$ is a Euclidean space. We provide definitions of perturbation vector fields and gradient vector fields for the case of general Riemannian manifolds in Appendix B.7.1.3. Further, we provide details on how to estimate such vector fields using finite data sets in Appendices B.7.5.3 and B.7.5.4.

B.3.3 *Encoding variations*

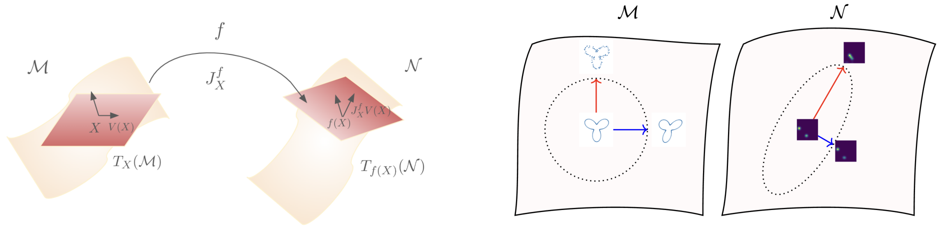


Figure B.4.: A visualization of the Jacobian map and the pull-back norm. Here f denotes an encoding map from the input space $\mathcal{M}$ to the output space $\mathcal{N}$. Left: the Jacobian of the encoding sends tangent vectors in the tangent space $T_X\mathcal{M}$ of $\mathcal{M}$ to tangent vectors in the tangent space $T_{f(X)}\mathcal{N}$ of $\mathcal{N}$. Right: the pull-back norm of a tangent vector on $\mathcal{M}$ measures by what amount the output of the encoding would change in response to the variation of the input by that tangent vector. In this schematic illustration, the pull-back norm of the red vector ("noising") is larger than the pull-back norm of the blue vector ("shearing").

Having characterized data variations in terms of vector fields, the next step is to describe the behavior of the encoding map f in response to these variations. Specifically, we are going to introduce the average pull-back norm of a vector field to quantify the sensitivity of the encoding map to the corresponding data variation. At the outset of this subsection, we emphasize that whether or not it is desirable to have an encoding that is sensitive to a particular data variation depends on the specific practice scenario and whether this variation is perceived as valuable information or as noise that one would like to filter out in the encoding.

JACOBIAN The *Jacobian* of an encoding map f, denoted by J_X^f, is a linear transformation between tangent spaces that characterizes the local behavior of f. While a tangent vector $v \in T_X\mathcal{M}$ describes one type of data variation at X, the image tangent vector $J_X^f(v)$ describes the resulting variation of the encoding $f(X)$,

$$J_X^f : T_X\mathcal{M} \to T_{f(X)}\mathcal{N}; \quad v \mapsto J_X^f(v).$$

We may write this linear transformation in terms of a Jacobian matrix $J_X^f \in \mathbb{R}^{n\times m}$ with respect to a basis. If there is no risk of confusion, we will omit the super-/subscripts f and X. We provide a visualization for the Jacobian map in the left panel of Figure B.4.

The rank of the Jacobian is the dimension of the image of $T_X\mathcal{M}$ under the Jacobian map, $\text{rank}(J_X^f) = \dim(J_X^f(T_X\mathcal{M}))$. It corresponds to the number of degrees of freedom of the data that are captured by the encoding. For instance, $\text{rank}(J_X^f) = \dim(T_X\mathcal{M})$ indicates that f is sensitive to all local data variations, whereas $\text{rank}(J_X^f) = 0$ means that f is approximately invariant under all local variations and thus approximately constant near X.

PULL-BACK NORM To measure the encoding's effectiveness in capturing a data variation, we introduce the average pull-back norm of a vector field. The *pull-back norm* of a tangent vector $V(X)$ at X is defined as[5]

$$\|V(X)\|_f = \|J_X^f(V(X))\|_{\mathcal{N}} = \sqrt{V(X)^T \cdot G_X^f \cdot V(X)}.$$

Here $\|\cdot\|_{\mathcal{N}}$ denotes the vector norm in output space $\mathcal{N}$ and $G_X^f = (J_X^f)^T J_X^f$ is the Gram matrix of the encoding f at X. While in the above definition, we consider the Euclidean metric for the output space of PIs, our approach can be applied for other choices of metric as well. We present the definition of pull-back norm for any differential encoding mapping between Riemannian manifolds in Appendix B.7.1.4. We also provide a visualization for the pull-back norm in the right panel of Figure B.4.

The pull-back norm of $V(X)$ measures the sensitivity of f to the variation $V(X)$ at X. To measure the sensitivity across different inputs, we take the average with respect to a distribution on $\mathcal{M}$. In practice, we use the empirical distribution of a given data set $\mathcal{D} = \{X_i\}_{i=1,\dots,|\mathcal{D}|}$.

Definition B.3 (Average pull-back norm)**.** The average pull-back norm of a vector field V with respect to an encoding map f and a data set $\mathcal{D} = \{X_i\}_{i=1,\dots,|\mathcal{D}|}$ of cardinality $|\mathcal{D}|$ is

$$\|V\|_f = \frac{1}{|\mathcal{D}|} \sum_{X\in\mathcal{D}} \|V(X)\|_f.$$

[5] Strictly speaking this is a semi-norm, as it may vanish for non-zero tangent vectors.

Please note that in Definition B.3, $V(X)$ denotes a tangent vector at X in the space of point clouds. Specifically, a vector $V(X) \in T_X\mathcal{M}$ corresponds to a "vector field" on X which assigns a vector to each point $x \in X$ in the point cloud X. We say that an encoding can detect a data variation characterized by a vector field V if the encoding is sensitive to V, which alludes to the average pull-back norm of V.

SINGULAR VALUE DECOMPOSITION To gain a more fine-grained insight into the properties of an encoding, we consider the singular value decomposition (SVD) of the Jacobian matrix,

$$J = \tilde{Q}\Lambda Q^T.$$

Here $\tilde{Q} \in \mathbb{R}^{n\times n}, Q \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and $\Lambda \in \mathbb{R}^{n\times m}$ is a diagonal matrix containing in its diagonal the singular values in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min(m,n)}$. This sequence of ordered singular value is the *spectrum* of the Jacobian J. Accordingly, the Gram matrix has eigendecomposition $G = J^T J = Q\Lambda^2 Q^T$. We denote the right singular vectors, i.e., the columns of Q, by $q_1, \ldots, q_m$. We will refer to these q_i's as the eigenvectors of the encoding. Any tangent vector $v \in T_X\mathcal{M}$ can be written as $v = \sum\langle v, q_i\rangle q_i$, and its pull-back norm as $\|v\|_f = \sqrt{\sum \lambda_i^2 \langle v, q_i\rangle^2}$. With this, the pull-back norm is decomposed as two parts: the spectrum of Jacobian and the alignment between v and eigenvectors, which is described by the inner product. In particular, the pull-back norm is large if v is aligned with q_i's that have large singular values. This also implies that the eigenvectors with top largest eigenvalues can be regarded as the data variations that the encoding considers most "important". We offer a visualization in Appendix B.7.3.

COMPARISON BETWEEN ENCODINGS Later we will compare different encodings by examining their sensitivity to specific data variations. To place different encodings on the same scaling level, we consider the normalized average pull-back norm $[\sum \|V(X)\|_f / \lambda_1^f]/|\mathcal{D}|$. We provide details about this normalization technique in Appendix B.7.5.2. Another way of comparing different encodings is via the Bures-Wasserstein distance [33] between their Gram matrices. The Bures-Wasserstein distance quantifies the alignment between the eigendecompositions of the Gram matrices. For two positive definite matrices A and B, the Bures-Wasserstein distance is computed as:

$$d_{\mathrm{BW}}(A, B) = \left[\mathrm{Tr}A + \mathrm{Tr}B - 2\mathrm{Tr}(A^{1/2}BA^{1/2})^{1/2}\right]^{1/2}.$$

For matrices that are not strictly positive definite we use the same definition after adding a small multiple of the identity matrix.

B.4 Identifying what is recognized

In this section, we seek to identify, for fixed PH encodings, which data variations are recognized and which are ignored. Specifically, we investigate the total amount of data variations that are captured by PH encodings (Section B.4.1), and among those captured variations we interpret the most significant ones (Section B.4.2). Then, we quantify the "importance" for any data variation (Section B.4.3), and measure the dissimilarity of the captured information across different PH encodings (Section B.4.4). It is important to notice that, even though our approach eliminates potential inference biases induced by downstream models, our results still significantly depend on the data set under consideration. Therefore, we emphasize that this section is dedicated to investigating "what is recognized" by PH encodings *within specific data sets*.

SYNTHETIC DATA Throughout this section we consider a synthetic data set of point clouds in $\mathbb{R}^2$ sampled from curves in the Radial Frequency Pattern (RFP) family. A point cloud of this type is shown in Figure B.2. The curve $\text{RFP}_{(a,w)}$ is parametrized by $\rho(\theta) = 1 + a\cos(w\theta), \theta \in (0, 2\pi]$. Loosely speaking it represents the shape of a flower with w petals of size characterized by a. We take w in $\{3, 4, \ldots, 10\}$ and 10 values of a evenly distributed on the interval $[0.5, 0.9]$. For each curve $\text{RFP}_{(a,w)}$, we evenly sample $N = 150$ points to obtain a point cloud, which is then scaled to the unit square $[0, 1]^2$ (see examples in Appendix B.7.5.3). Notably, each curve in the RFP family has the same topology. This allows us to validate the ability of PH to capture information beyond topology. The RFP data set has also been used in studying the importance of specific shape features in shape recognition and object perception [310].

PH ENCODINGS We investigate PH encodings constructed on 3 different filtrations: Vietoris-Rips (Rips) filtration, DTM filtration, and Height filtration (with respect to the hyperplane with normal vector $[1, 0]^T$) (see Section 1.3 for definitions). For each filtration we extract the 1-dimensional PDs, and convert them to PIs with the same PI parameters. In the following discussion, we sometimes refer to these encodings by the name of the filtration on which they are constructed, denoting for instance the PH encoding constructed on Rips filtration simply as Rips. For reference we also include a PointNet encoding. PointNet [282] is a deep neural network architecture designed for processing point clouds directly. We train the network to predict the number of petals w, achieving a test accuracy of 100%. We take the output of the second-to-last layer of the trained PointNet as the output of the PointNet encoding. More details concerning the filtration and PI parameters, and the PointNet encoding are provided in Appendix B.7.5.3.

B.4.1 *Spectrum of the Jacobian*

As explained in Section B.3.3, the rank of the Jacobian indicates the maximum amount of information, in a dimension sense, that can be captured by the encoding. The spectrum, i.e., the sequence of ordered singular values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, provides a fuller picture, indicating to what extent different eigenvectors are highlighted.

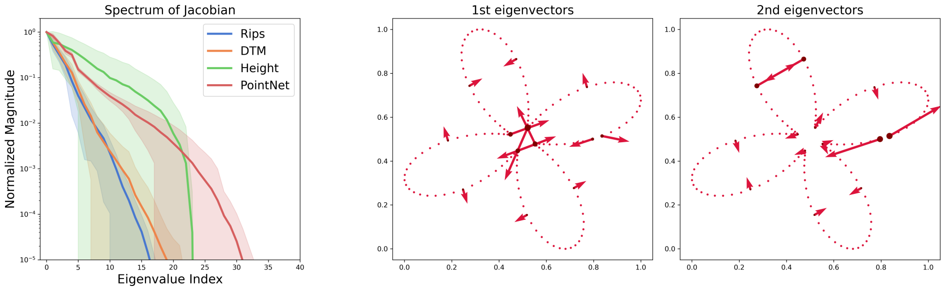


Figure B.5.: Left: The normalized spectrum of the Jacobian for different encodings. Shown is the mean and standard error of the ordered normalized singular values over different input point clouds. Right: The top two eigenvectors of the Jacobian for the PH encoding constructed on the Rips filtration at a particular input point cloud.

The left plot in Figure B.5 reports the normalized spectrum of the Jacobian for different encodings, which is the sequence of ordered normalized singular values

$$1 \geq \frac{\lambda_2}{\lambda_1} \geq \cdots \geq \frac{\lambda_{\mathrm{rank}(J)}}{\lambda_1} > 0.$$

The first observation is that the rank of the Jacobian is much smaller than the dimension of the point cloud space and the PI space. Indeed, note that the dimension of $\mathcal{M}$ is $m = D \times N = 2 \times 150 = 300$, and the dimension of $\mathcal{N}$ is $n = P \times P = 20 \times 20 = 400$. On the other hand, the normalized singular values decay to 10^{-5} before index 40. We conclude that the four encodings under consideration capture only a small set of variations in the input data and discard many others. Secondly, while Rips, DTM, and Height all have a similar number of singular values larger than 10^{-5}, Rips and DTM exhibit a sharper initial decay than Height. This implies that Height has a larger effective rank[6], while Rips and DTM concentrate their attention more specifically on a few variations. This difference in decay rates may stem from the fact that the size of holes, which is captured by Rips and DTM, is influenced by fewer variations compared to the position of holes, which is information retained by Height.

The middle and right plots in Figure B.5, show the top two eigenvectors for Rips at an example point cloud X. An eigenvector corresponds to a list of vectors attached

[6] The effective rank is the number of singular values that have a similar order of magnitude as the top singular value.

to all individual points in the point cloud. These eigenvectors provide insight into the nature of the "important" data variations. More technically, these variations correspond to the most effective way to change the birth/death parameters of certain homology classes.[7] For instance, in the middle plot of Figure B.5, the vectors on the petals depict variations that narrow/broaden the petals, which in turn change the death parameter of the corresponding homology classes. The eigenvectors can also be used to obtain point saliency maps, which we discuss in Appendix B.7.4. However, we observe that the eigenvectors do not necessarily have an obvious intuitive description. Hence, interpretations are needed to bridge the gap between abstract eigenvectors and human-understandable concepts.

B.4.2 *Alignment between eigenvectors and perturbation tangent vectors*

To interpret the eigenvectors of the encoding, we consider their alignment with different perturbation vector fields.

PERTURBATIONS ON THE DATA MANIFOLD We consider eight types of perturbations applied to the input data, illustrated in Figure B.6. The first two, *rotation* and *translation*, are Euclidean motions, i.e., transformations that preserve the Euclidean distances between the points in a point cloud. They are used to test the fact that pointwise distance-based encodings, namely Rips and DTM, should remain invariant under such variations. The *dilation*, *strecth_x*, and *shearing* variations serve to test the sensitivity of the encoding to invertible linear transformations of the point clouds. The next two variations are used to test the robustness of PH encoding against noise: the *noising* variation adds coordinate-wise Gaussian noise at each point in the point cloud; the *wiggly* variation adds a sine-type noise at every point in the point cloud along the normal direction. Lastly, the *convex* variation transforms the point cloud towards the boundary of its convex hull through a linear interpolation. We present a visualization of the effects of *shearing* and *convex* on the PH associated with Rips filtration and Height filtration in Appendix B.7.5.3.

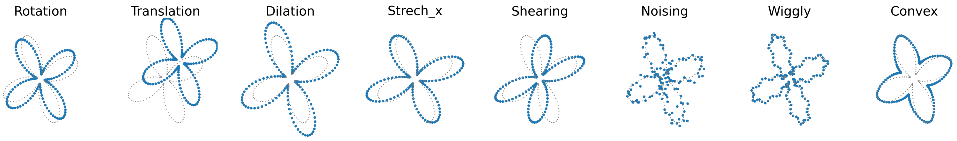


Figure B.6.: Eight types of perturbations on a RFP pointcloud.

We examine the angles between the top eigenvectors of the different encodings and different perturbation tangent vectors. In Figure B.7, we record the average inner

[7] This bears some resemblance to adversarial perturbations considered in neural networks.

product between the perturbation tangent vectors and the top four eigenvectors of each encoding method:

$$\frac{1}{|\mathcal{D}_{\text{RFP}}|}\sum_{X\in\mathcal{D}_{\text{RFP}}}\left|\left\langle \frac{V_\pi(X)}{\|V_\pi(X)\|}, q_i^f \right\rangle\right|, \quad i = 1,2,3,4.$$

The eigenvectors q_i^f depend on X. Since each encoding map f induces a different orthonormal basis $\{q_1^f, q_2^f, \dots, q_m^f\}$ on the tangent space of the data manifold, a fixed tangent vector will have different coordinates with respect to the different encodings. Figure B.7 serves as a sort of dictionary, showing how each perturbation (rotation, translation, dilation, etc.) is expressed in the language of each encoding.

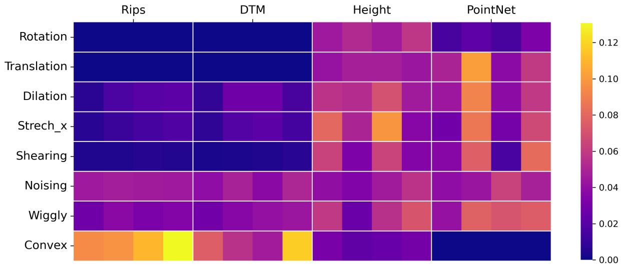


Figure B.7.: Absolute inner product between perturbation vectors and top four eigenvectors of different encodings. A higher value implies greater alignment, i.e., greater sensitivity to a perturbation.

For the PH encoding constructed on Rips and DTM filtration, we find that the top eigenvectors exhibit a relatively strong alignment with *convex*. The corresponding average inner product is around 0.1. This indicates that the most "important" data variation for Rips and DTM are closely related to convexity. This is consistent with the fact that these encodings capture geometric properties, such as birth values of holes in the filtration (that increase under the *convex* perturbation, see Figure B.6). At the same time, the top eigenvectors of Rips and DTM are orthogonal to *rotation* and *translation*. This indicates that Euclidean motion is not as relevant in the Rips and DTM, as is to be expected from the definitions of these encodings.

For the PH encoding constructed on Height filtration, the top eigenvectors have a significant alignment with Euclidean motions and *stretch_x*. This makes sense, since Height is designed to collect information on the position of holes. On the other hand, we do not observe a strong alignment between top eigenvectors of Height and *convex*. This might seem to be in contradiction with [338], who demonstrated that PH on height filtration can be used for detecting convexity. Note, however, that they use 0-dimensional PH on cubical complexes (or analogously, on Rips complexes on geodesic distances), which recognizes concave shapes by their multiple connected components (0-dimensional cycles) for at least some height filtration directions. 0-dimensional PH with respect to such a filtration would see two connected components (two petals) for a while, that would then merge into one at some point.

This would happen earlier under *convex* perturbation (i.e., one of the connected components would die sooner), so that the alignment can be expected to be more significant in that case.

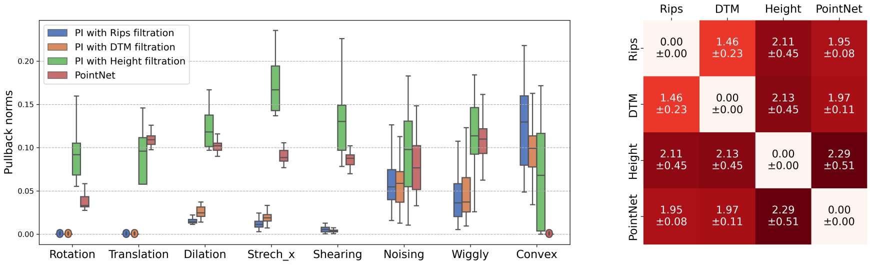
For PointNet encoding, the top eigenvector has a relatively strong alignment with *translation*. This is consistent with a previous observation by [338] where the PointNet did not perform well in classification tasks when the test data was corrupted by translations. Note akin to the approach adopted by [338], we do not use data augmentation techniques during the training for PointNet. This might lead to the sensitivity of the trained PointNet to *translation*. On the other side, PointNet is robust under *convex* perturbations. This can be attributed to the nature of RPF data set. Recall the RPF data set comprises point clouds defined by two independent parameters, a and w, which characterize the size and the number of the petals, respectively. While the PointNet is trained to identify the number of petals, it can easily learn from the data set to ignore the size of petals. Notice increasing the petal size bears strong resemblance to *convex* perturbation (see examples of RPF point clouds in Appendix B.7.5.3). Therefore, one can loosely infer that the RPF data set is "inherently" augmented by *convex* perturbation, and consequently the trained PointNet might learn from the data to ignore *convex* information.

B.4.3 *Pull-back norm of perturbation vector fields*

In some scenarios, one is interested in the sensitivity of an encoding to certain types of perturbations [292]. In Figure B.8 (left) we evaluate the average pull-back norm of different perturbation vector fields with respect to different encodings. The pull-back norm takes into account not only the alignment with the encoding eigenvectors but also the magnitude of the corresponding singular values.

For DTM and Rips, we find that *noising*, *wiggly*, and *convex*, have a significantly larger pull-back norm than the other data variations. This is consistent with the Jacobian spectrum in Figure B.5 (which indicates DTM and Rips have faster-decaying spectrum and therefore capture only few data variations), and the alignment information in Figure B.7 (which indicates alignment of the top eigenvectors with these particular variations). The PH encoding constructed on Height filtration has a relatively large pull-back norm for many of the considered data variations, including dilation, stretch, and shearing. Also the pull-back norm associated with *convex* perturbations with respect to Height exhibits a moderate average value and a large variance. This implies Height is sensitive to *convex* perturbations in certain point clouds, while being less sensitive in others. This aligns with the alignment information in Figure B.7, which indicates the most "important" data variations seen by Height, on average, are not closely related with *convex*. Similar to Height, PointNet also leads to relatively large pull-back norms, but with a different profile and with exception of *convex*, which has a small pull-back norm under PointNet. Rips and DTM have a faster-decaying

Jacobian spectrum than Height and PointNet, indicating that they are sensitive to fewer data variations.



	Rips	DTM	Height	PointNet
Rips	0.00 ±0.00	1.46 ±0.23	2.11 ±0.45	1.95 ±0.08
DTM	1.46 ±0.23	0.00 ±0.00	2.13 ±0.45	1.97 ±0.11
Height	2.11 ±0.45	2.13 ±0.45	0.00 ±0.00	2.29 ±0.51
PointNet	1.95 ±0.08	1.97 ±0.11	2.29 ±0.51	0.00 ±0.00

Figure B.8.: Left: Average pull-back norm of different perturbation vector fields with respect different encodings. Right: Bures-Wasserstein distance between Gram matrices $J^T J$ of different encodings.

B.4.4 *Distance between Gram matrices*

Along this thread, we can also investigate the relationship between encodings. The right panel of Figure B.8 shows the average Bures-Wasserstein distance between the Gram matrices of different encodings.

The average distance matrix shown in the right part of Figure B.8 indicates that all encodings are different, whereby some are more similar and some are more dissimilar (see also alignment pattern in Figure B.7). Rips and DTM are closest each other, while PointNet and Height both differ significantly from Rips and DTM. This indicates that Rips and DTM capture similar information which is different from the information that is captured by Height and PointNet. This makes sense, since the DTM filtration function is the average distance to neighbors, which approximates the distance function that underlies the Rips filtration; moreover, the data we consider does not contain outliers. We also find that although Height and PointNet give relatively similar pull-back norms for rotation, translation, and dilation, overall these two encodings are very different.

B.5 **Selecting hyperparameters**

In this section, we shift our focus to the problem: how do we select the hyperparameters of the encoding in order to detect a data feature of interest. Recall that there are three major hyperparameters to choose when constructig PIs: 1) the resolution P, 2) the kernel function $g_{(b,l)}(x,y)$ and its associated parameters and 3) the weighting function $\alpha(b,l)$. We first focus on the first two PI parameters, namely the resolution and the variance for the Gaussian kernels. We examine their impact on the rank and spectrum of Jacobian (Section B.5.1.1) and on the pull-back norm of gradient vector fields of data features of interest (Section B.5.1.2). We demonstrate there is a strong correlation between the pull-back norms of gradient vector fields and the downstream task performance, where the task objective is to predict that data feature (Section B.5.1.3). We then investigate the impact of weighting functions on the pull-back geometry. We introduce the *beta weighting function*, which allows highlighting persistence intervals with different length (persistence time). Then we examine the effects of the mean parameter of beta weighting function on the pull-back geometry (Section B.5.2.1). Finally, again we show a significant correlation between the pull-back norm of gradient vector fields and the downstream task performance (Section B.5.2.2).

REAL-WORLD DATA In this section we utilize the brain artery tree data [29]. This data set comprises 96 artery trees in $\mathbb{R}^3$ (see Figure B.9, left). These artery trees are obtained by applying a tube-tracking algorithm to Magnetic Resonance Angiography (MRA) images from 96 human subjects. We randomly subsample three point clouds from the vertices of each artery tree, with each point cloud containing $N = 500$ points. Then we normalize the sampled point clouds to the unit cube $[0,1]^3$.

FEATURE Each point cloud is labeled with a binary *sex* feature, based on the corresponding human subjects' medical information.

B.5.1 *Resolution and variance of Gaussian kernel*

PH ENCODING We focus on the 1-dimensional PIs on the Vietoris-Rips filtration. We investigate two hyperparameters involved in the construction of PIs: 1) the resolution P, and 2) the variance γ^2 of the Gaussian kernel (see Figure B.9, right). We set the baseline PI parameters as $P = 20$, $\gamma^2 = 10^{-4}$, and consider a linear weighting function $\alpha(b,l) = \frac{l}{\max\{l\}}$ [2].

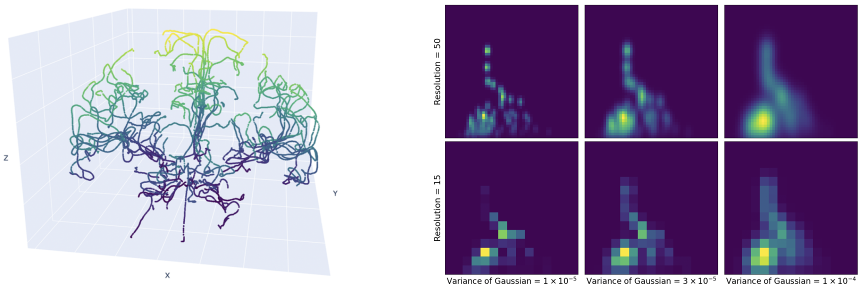


Figure B.9.: Left: point cloud sampled from a brain artery tree, where the color represents the z-coordinate. Right: the corresponding PI representation with different parameter settings.

B.5.1.1 *Spectrum of the Jacobian*

We begin by analyzing the effects of the resolution P and variance γ^2 of the Gaussian kernel on the spectrum of the Jacobian. In each plot of Figure B.10, we varied one parameter while keeping the other fixed at the baseline setting, and present the normalized spectrum. We again observe a low-rank phenomenon, since the average rank is always below 160, while in the baseline setting the dimension of the point cloud and PI spaces are respectively $m = D \times N = 3 \times 500 = 1500$ and $n = P \times P = 20 \times 20 = 400$.

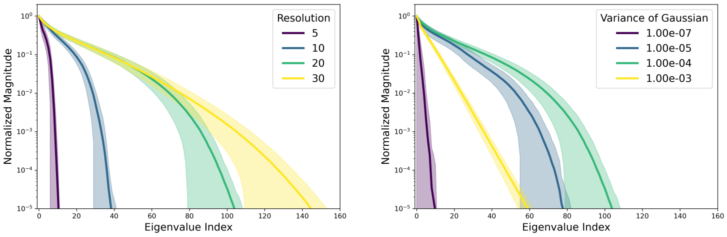


Figure B.10.: Spectrum of the Jacobian matrix resulting from the PH encoding on the brain artery tree data set, across different values of the PI parameters: resolution P (left plot), and Gaussian kernel variance γ^2 (right plot).

In the left part of Figure B.10, we see that the spectrum decays slower as the resolution increases, indicating an increase in rank. This implies that, as one would expect, higher resolution allows the PI to capture more information.

Interestingly, we observe that, as the variance γ^2 increases, the rank of the Jacobian initially increases and then decreases. For fixed resolution, very small variance results in sparse PIs (see the first column in the right panel of Figure B.9), where multiple PD points (b, l) may fall into one pixel and can only highlight that pixel; very large variance leads to blurred PIs (see the third column in the right panel of Figure B.9), where it also becomes difficult to distinguish between PD points.

B.5.1.2 *Pull-back norm of feature gradient vector fields*

We now explore the effects of the resolution and variance on the pull-back norm of gradient vector fields of the following data feature. Our goal is to locate the optimal PI parameters of the PH encoding to effectively detect the *sex* feature in the brain artery point clouds. We consider the domain $(P, \gamma^2) \in [15, 30] \times [10^{-5}, 10^{-4}]$. Here the ranges for P and γ^2 are selected based on the values where the spectra in Figure B.10 exhibit the slowest decay.

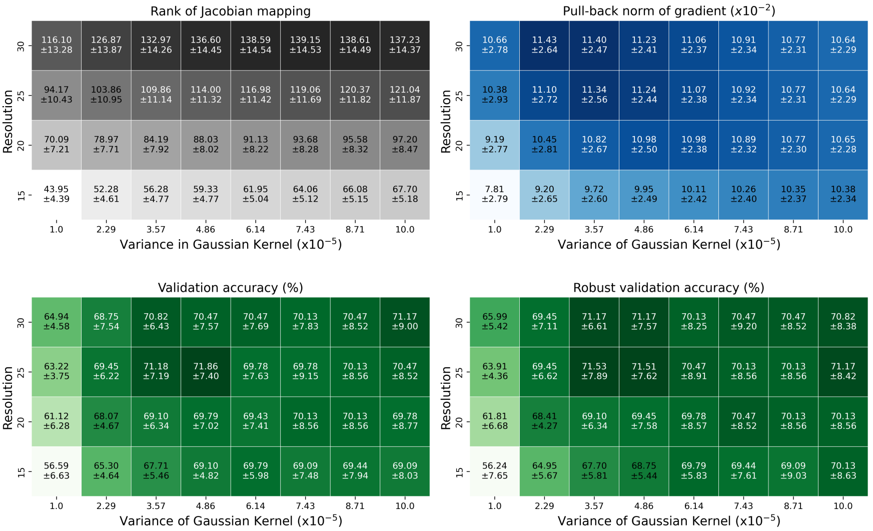


Figure B.11.: For the brain artery data set, shown is the effect of the resolution (vertical axis) and variance of the Gaussian kernel (horizontal axis) of the PI on the average rank of the Jacobian (upper left), average pull-back norm of the gradient vector field of the *sex* feature (upper right), as well as the test accuracy (lower left) and robust test accuracy (lower right) of the logistic regression model predicting *sex* based on the PI.

In the upper right plot of Figure B.11, we present the average pull-back norm of the gradient field of the *sex* feature under different PI parameter choices. The gradient fields are estimated via numerical methods detailed in Appendix B.7.5.4. We observe that the pull-back norm generally increases as the resolution P increases, whereas the pull-back norm is not monotonic with γ^2. Moreover, the optimal value for γ^2 varies depending on the choice of resolution. Also, comparison with the upper left part of Figure B.11 reveals that the maximum pull-back norm is not necessarily attained for parameters where the rank of the Jacobian is maximal, i.e., when PIs capture the most information about the point cloud.

B.5.1.3 *Correlation with downstream task performance*

We investigate the hypothesis that a high pull-back norm correlates with the performance of a predictor trained on the encoding. To this end we feed PIs generated with different choices of the parameters into logistic regression models and train these to predict the *sex* feature. Here we use logistic regression as the downstream model because of its simplicity, with the intention to minimize the impact of model complexity and training techniques on the task performance. We provide results for convolutional neural networks (CNN) in Appendix B.7.5.4.

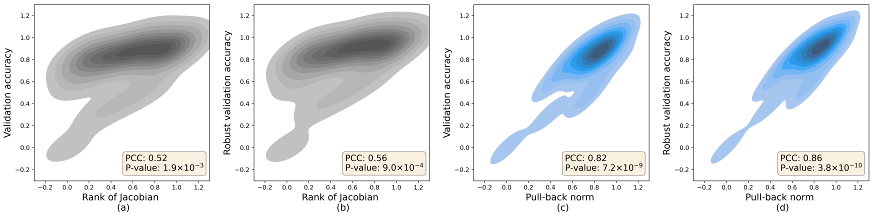


Figure B.12.: Gaussian kernel density estimation of the joint distribution of the rank of the Jacobian or pull-back norm vs. logistic regression (robust) validation accuracy.

We evaluate the performance in terms of validation accuracy and robust validation accuracy[8] using cross-validation, which are presented in the lower left and lower right plots in Figure B.11. The validation accuracy and robust validation accuracy exhibit a similar pattern to the pull-back norm. Notably, all three quantities reach their maximum at around $P = 30$ and $\gamma^2 = 3.57 \times 10^{-5}$, and their minimum at the lower-left corner.

For a more quantitative comparison, Figure B.12 shows a kernel density estimate of the joint distribution of four pairs of variables: Jacobian rank vs. validation accuracy, Jacobian rank vs. robust validation accuracy, pull-back norm vs. validation accuracy, and pull-back norm vs. robust validation accuracy. The plots clearly indicate a strong correlation between the pull-back norm and the performance on the downstream task. The Pearson's correlation coefficient (PCC) between the four pairs of variables, along with the p-value for a two-sided test, are presented in the lower right corner of each plot in Figure B.12.

We conclude that for the considered task, there is a significant correlation between the pull-back norm and the downstream task performance. It is also interesting to interpret these results together with the rank of the Jacobian (upper left in Figure B.11). The results demonstrate that the improvement in downstream performance is only somewhat correlated with including more information, but it is strongly correlated with including the most relevant information, which is precisely quantified by the pull-back norm. Therefore, we suggest that the proposed framework can be used to select appropriate PH encodings in practice. Note that the procedure is independent of the downstream model architectures and training techniques.

B.5.2 *Weighting function*

PH ENCODING We maintain our focus on the 1-dimensional PIs with respect to the Vietoris-Rips filtration. We set the baseline PI parameters as $P = 20$ and

[8] Robust validation accuracy evaluates the accuracy on a test data set subject to additive Gaussian noise on the inputs.

$\gamma^2 = 3 \times 10^{-5}$. For the weighting function, we consider the *beta weighting function* induced by the probability density function of a beta distribution:

$$\alpha(b,l) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}(\kappa l)^{\alpha-1}(1-\kappa l)^{\beta-1}$$

where $\Gamma(\cdot)$ is the Gamma function and κ is a scaling factor. We consider the mean-variance parameterization for the beta weighting function: $\alpha = k\left(\frac{k(1-k)}{s^2} - 1\right)$ and $\beta = (1-k)\left(\frac{k(1-k)}{s^2} - 1\right)$. Here k is the *mean parameter*, which controls the concentration of the weighting function, and s^2 is the variance parameter, which controls the "degree" of concentration. We set s^2 as 0.065 and κ as 1.

We consider beta weighting function for several reasons: 1) beta weighting function is compactly supported, which is more suitable for PI; 2) it assigns zero weight to the horizontal axis, which aligns with the stability criteria proposed by [2]; 3) by tuning the mean parameter, one can highlight persistence intervals with different length (see Figure B.13 for an illustration). This allows to investigate questions such as "are short persistence intervals more crucial to this application than long persistence intervals?"

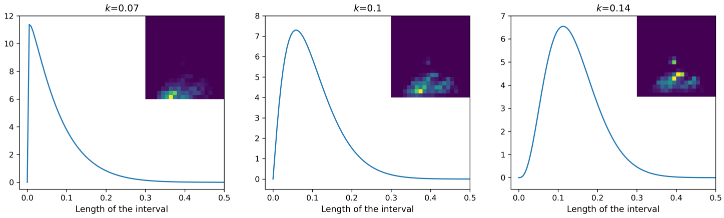


Figure B.13.: As a weighting function for PIs for the brain artery data, we use the beta weighting function with different values of mean parameter k. Larger k assigns more importance to longer persistence intervals. The top right corner of each panel shows the 1-dimensional PI derived from the Rips filtration on one point cloud, illustrating the impact of the weighting function depicted in the main plot.

B.5.2.1 *Pull-back norm of feature gradient vector fields*

We investigate the effects of the mean parameter k on the rank of Jacobian and pull-back norm of the gradient vector field of the *sex* feature (see the left and middle panel in Figure B.14). In the left panel of Figure B.14, we observe that as the weighting function assigns more importance to longer persistence intervals, the rank of the Jacobian monotonically decreases. This aligns with the fact that the number of longer persistence intervals is generally smaller than the number of shorter persistence intervals. However, the pull-back norm peaks when k is set to 0.1 and then decreases as k increases further.

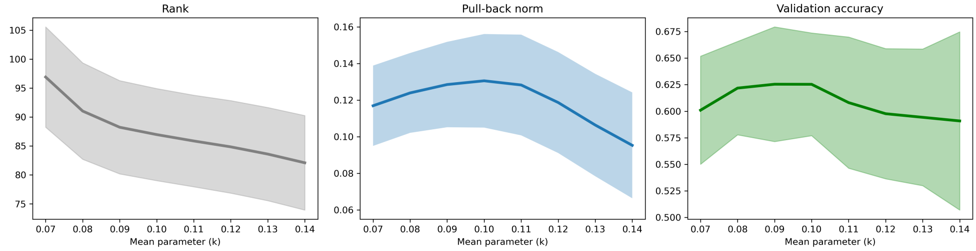


Figure B.14.: The impact of the mean parameter for the beta weighting function on the rank of Jacobian (left), pull-back norm of gradient field of feature *sex* (middle), and the 7-folded validation classification accuracy (right). The pull-back norms and validation accuracy are strongly correlated, and they both indicate persistence intervals with medium length are vital to classify the *sex* feature.

B.5.2.2 *Correlation with downstream task performance*

We again examine the correlation between the pull-back norm and the performance of the logistic regression models trained on PIs. We present the validation accuracy in the right panel of Figure B.14. Notably, we observe that the validation accuracy demonstrates a similar pattern to the pull-back norm. Quantitatively, the Pearson's correlation coefficient (PCC) between pull-back norm and validation accuracy is 0.839, with a two-sided test p-value of 0.009. In contrast, the PCC between rank and validation accuracy is 0.338, with a p-value as 0.413, which indicates once again that including more information in the data representation does not necessarily improve the downstream performance. These findings reinforce our conclusion that the pull-back norm is highly predictive for the downstream task performance in this task.

Interestingly, we observe that both the pull-back norm and the validation accuracy reach their maximal at an intermediate value for the mean parameter. This implies that persistence intervals of medium length are most crucial for classifying the *sex* feature, which is consistent with the observation in the original paper [29]. We note that this is an example of application for which medium length intervals in the barcode contain the most information for the problem at hand. In Appendix B.7.6, we complement this discussion by considering a real-world data set of point clouds sampled from human body meshes, on which we compare long and short persistence intervals from a different perspective: which *part* of the point clouds is the focus of long intervals and which is the focus of short ones.

B.6 Conclusions

The methods and observations presented in this work contribute to addressing some of the main bottlenecks in the practical application of PH, namely how to identify which data variations are captured by PH encodings, how to quantify the effectiveness of these encodings in detecting particular data features, and how to select the parameters of PH encodings in order to obtain data representations that are suitable for solving a particular task.

We presented ways to analyze the most relevant features on the data manifold that are captured by persistence images with different choices of the filtration and compared the results with neural-network-based encodings. For example, in the RFP dataset we found that while a pretrained PointNet had a relatively high alignment with translation and dilation, the 1-dimensional persistence image encoding with Height filtration had a high alignment with stretch, and the 1-dimensional persistence image encoding with Rips filtration had a high alignment with a data variation that makes the point clouds more convex. At the same time we observe that the response of the encodings to these perturbations is less than 10% as strong as for other more abstract data variations captured by the singular vectors of the Jacobian; for instance, the maximal value taken by inner products between unit tangent vectors representing perturbations and unit singular vectors is less than 0.1.

We demonstrated on the real-world brain artery tree data set that feature alignment as measured by the Jacobian permits PH parameter tuning without the need to train a classifier on top of the data representation in order to select the parameters based on the test accuracy. Rather, one can select the parameters based on the pull-back norm of the features of interest, and perform training using the data representation with the highest pull-back norm. Meanwhile, we found that the persistence intervals of medium length are crucial for classifying the *sex* feature on this data set. This goes against the popular belief that long intervals are the most important, at the same time confirming the findings from the original paper that employs PH on this data set.

limitations and future work Our analysis is based on the structure of the Jacobian of the data encoding, which by nature focuses only on local variations of the input data. In future it will be interesting to further advance these methods in regard to non-local data variations, where synthetic notions of derivatives such as our empirical evaluation of the vector fields, and ideas such as the application of the iterated closest point method could serve as a point of departure. The analysis of non-linear transformations via Gram matrices has seen a number of recent advances in the context of artificial neural networks. It will be interesting to explore possible synergies between those investigations and PH data encodings.

Another limitation of the proposed methodology is the assumption about the differentiability of the encoding, and the need for a Riemannian manifold structure for the representation space. For this reason, our methodology cannot be directly applied to analyze PH representations such as the most common persistence diagrams, since these cannot be endowed with a smooth structure [206]. We note, however, that a Riemannian framework for *approximated* PDs has been introduced in [13].

The pull-back geometry approach also faces some computational challenges, since calculating the average pull-back norm for either perturbation vector fields or gradient vector fields requires the computation of the Jacobian matrix J_X, which is of size $(P \times P, D \times N)$, for each data point X, where P, D, N are respectively the resolution for PI, the dimension of points in the point cloud and the number of points in the point cloud. On the positive side, this enables insights that are more intrinsic to the problem rather than being dependent on the choice of a classifier. Performance-based methods can also involve computational challenges, due to the need to choose the downstream models and tune their hyperparameters. We regard the proposed methods not as a substitute but as complementary to performance-based methods.

B.7 Supplementary material

The appendix is organized into the following sections.

Appendix B.7.1	Riemannian manifold structure of the space of point clouds
Appendix B.7.2	Differentiability of the mapping from point clouds to PIs
Appendix B.7.3	Visualizing the Jacobian of the encoding over the data manifold
Apeendix B.7.4	Point saliency maps for PH encodings
Appendix B.7.5	Details on the experiments
Appendix B.7.6	Investigating which part of the data is highlighted by PH encodings

B.7.1 *Riemannian manifold structure of the space of point clouds*

Let $\mathcal{M}$ denote the collection of all point clouds in $\mathbb{R}^D$ that contain exactly N points,

$$\mathcal{M} = \{X \subset \mathbb{R}^D \colon |X| = N\}.$$

Recall that the 2-Wasserstein distance d_W between two point clouds of the same size in $\mathbb{R}^D$ is defined as

$$d_W(X, Y) = \min_{\omega \in \Omega(X,Y)} \left(\sum_{x \in X} d_E^2(x, \omega(x)) \right)^{\frac{1}{2}}$$

where ω is an bijection between X and Y, $\Omega(X, Y)$ contains all bijections, and d_E denotes the Euclidean metric on $\mathbb{R}^D$. The 2-Wasserstein distance defined above induces a metric topology on $\mathcal{M}$.

Compared to other distances in the space of point clouds, for instance the Gromov-Hausdorff distance which is commonly used in the study of stability theory of persistent homology [see, e.g., 37], the 2-Wasserstein distance endows the space of point clouds with a favorable manifold structure. This manifold structure ensures that every small neighborhood is isometric to an Euclidean open set.

We discuss the topological manifold structure (Appendix B.7.1.1) and Riemannian manifold structure (Appendix B.7.1.2) on the space of point clouds. Then we introduce the Riemannian style definition for perturbation vector fields and gradient vector fields (Appendix B.7.1.3).

B.7.1.1 *Manifold structure*

We proceed to establish a manifold structure on $\mathcal{M}$.

Proposition B.4. *Let $\mathcal{M}$ be the set containing all point clouds in $\mathbb{R}^D$ with N distinct points, and d_W be the 2-Wasserstein distance on $\mathcal{M}$. For any point cloud $X \in \mathcal{M}$, there exists a Wasserstein ball $B_W(X, \varepsilon_X)$ and an injective mapping $\xi_X : B_W(X, \varepsilon_X) \to \mathbb{R}^{D\times N}$ such that*

$$d_W(Y, Z) = d_E(\xi_X(Y), \xi_X(Z)), \quad \forall Y, Z \in B_W(X, \varepsilon_X).$$

Proof. Consider a point cloud $X = \{x_i\}_{i=1}^N$ in $\mathcal{M}$. For arbitrary $\varepsilon > 0$, we construct an injective mapping ξ_X from $B_W(X, \varepsilon)$ to $\mathbb{R}^{D\times N}$, then choose a radius ξ_X such that the above equation holds. Denote $[N] = \{1, 2, \ldots, N\}$. The map $\xi_X(X)$ can be characterized by a total order in X, $\tau \colon [N] \to X$, where

$$\xi_X(X) = [\tau(1), \tau(2), \ldots, \tau(N)] \in \mathbb{R}^{D\times N}.$$

τ reorders X by assigning $[N] = \{1, 2, \ldots, N\}$ to $\{x_i\}_{i=1}^N$. For any other point cloud $Y \in B_W(X, \varepsilon)$, there exists an optimal transport plan between X and Y, denoted by $\omega_{XY} : X \to Y$, satisfying

$$\omega_{XY} = \underset{\omega \in \Omega(X,Y)}{\arg\min} \left(\sum_{x \in X} d_E^2(x, \omega(x)) \right)^{\frac{1}{2}}.$$

This maps assigns each element in X to a distinct element in Y. We define an embedding ξ_X from Y to $\mathbb{R}^{D\times N}$ as follows:

$$\xi_X(Y) = [\omega_{XY} \circ \tau(1), \omega_{XY} \circ \tau(2), \ldots, \omega_{XY} \circ \tau(N)] \in \mathbb{R}^{D\times N}.$$

We proceed to show that ξ_X is an injective embedding. For any $Y, Z \in B_W(X, \varepsilon)$ with $\xi_X(Y) = \xi_X(Z)$, consider optimal transport plans ω_{XY} between X and Y, and

ω_{XZ} between X and Z. Since $\xi_X(Y) = \xi_X(Z)$, we have $\omega_{XY}(x) = \omega_{XZ}(x), \forall x \in X$. Hence, for any $y \in Y$,

$$y = \omega_{XY} \circ (\omega_{XY})^{-1}(y) = \omega_{XZ} \circ (\omega_{XY})^{-1}(y) \in Z.$$

Notice $\omega_{XZ} \circ (\omega_{XY})^{-1}$ is a bijection between Y and Z. Therefore, $Y = Z$ and ξ_X is injective.

Next we calculate the radius ξ_X that preserves the distance between any two point clouds. The goal is to find a radius ε such that for any $Y, Z \in B_W(X, \varepsilon)$, the Wasserstein distance between the point clouds Y and Z,

$$d_W(Y, Z) = \left(\sum_{y \in Y} d_E^2(y, \omega_{YZ}(y)) \right)^{\frac{1}{2}},$$

is equal to the Euclidean distance between the embedding $\xi_X(Y)$ and $\xi_X(Z)$,

$$\begin{aligned} d_E(\xi_X(Y), \xi_X(Z)) &= \left(\sum_{i=1}^{N} d_E^2(\omega_{XY} \circ \tau(i), \omega_{XZ} \circ \tau(i)) \right)^{\frac{1}{2}} \\ &= \left(\sum_{x \in X} d_E^2(\omega_{XY}(x), \omega_{XZ}(x) \right)^{\frac{1}{2}} \\ &= \left(\sum_{y \in Y} d_E^2(y, \omega_{XZ} \circ (\omega_{XY})^{-1}(y) \right)^{\frac{1}{2}}. \end{aligned}$$

Notice it suffices to find a radius ε such that for any $Y, Z \in B_W(X, \varepsilon)$, $\omega_{YZ} = \omega_{XZ} \circ (\omega_{XY})^{-1}$. Equivalently, the optimal bijection between Y and Z is given by the composition $\omega_{XZ} \circ (\omega_{XY})^{-1} : Y \to X \to Z$. The key idea is that if Y and Z are both sufficiently close to X in the sense of the Wasserstein distance, then the distance between $y \in Y$ and $(\omega_{XY})^{-1}(y)$ and the distance between $z \in Z$ and $(\omega_{XZ})^{-1}(z)$ will be small. Hence, each point $y \in Y$ will be close to $\omega_{XZ} \circ (\omega_{XY})^{-1}(y)$ and thus we will have $\omega_{YZ} = \omega_{XZ} \circ (\omega_{XY})^{-1}$. The situation is illustrated in Figure B.15.

Denote $\omega_{XZ} \circ (\omega_{XY})^{-1}$ as $\widetilde{\omega_{YZ}}$. For any point cloud $Y \in B_W(X, \varepsilon)$, we have

$$d_W(X, Y) = \left(\sum_{x \in X} d_E^2(x, \omega_{XY}(x)) \right)^{\frac{1}{2}} < \varepsilon.$$

Hence,

$$\max_{x \in X} d_E(x, \omega_{XY}(x)) < \varepsilon.$$

This suggests for any two point clouds $Y, Z \in B_W(X, \varepsilon)$,

$$\max_{x \in X} d_E(\omega_{XY}(x), \omega_{XZ}(x)) < 2\varepsilon.$$

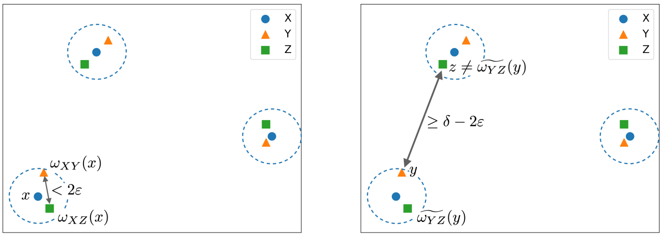

X
Y
Z
$\omega_{XY}(x)$
$< 2\varepsilon$
x
$\omega_{XZ}(x)$
X
Y
Z
$z \neq \widetilde{\omega_{YZ}}(y)$
$\geq \delta - 2\varepsilon$
y
$\widetilde{\omega_{YZ}}(y)$

Figure B.15.: Point clouds Y (orange triangles) and Z (green squares) in a Wasserstein neighborhood of point cloud X (blue circles). Right: both $\omega_{XY}(x) \in Y$ and $\omega_{XZ}(x) \in Z$ are close to $x \in X$, and hence they are close to each other. Left: the distance between y and $z \neq \widetilde{\omega_{YZ}}(y) \in Z$ is lower bounded.

Equivalently,

$$\max_{y \in Y} d_E(y, \widetilde{\omega_{YZ}}(y)) < 2\varepsilon.$$

Let δ denote the minimal pairwise distance of points in X:

$$\delta = \min_{x_1, x_2 \in X} d_E(x_1, x_2).$$

Note δ is strictly greater than zero since points in X are mutually different. For a fixed point $y \in Y$, any point z in Z other than $\widetilde{\omega_{YZ}}(\mathrm{y})$ has a lower-bounded distance from y:

$$\min_{z \in Z \setminus \{\widetilde{\omega_{YZ}}(y)\}} d_E(y, z) \geq \delta - 2\varepsilon, \quad \forall y \in Y.$$

Now consider $\varepsilon_X = \frac{\delta}{8}$. We have

$$d_E(y, \widetilde{\omega_{YZ}}(y)) < 2\varepsilon_X = \frac{1}{4}\delta < \frac{3}{4}\delta = \delta - 2\varepsilon_X \leq \min_{z \in Z \setminus \{\widetilde{\omega_{YZ}}(y)\}} d_E(y, z), \quad \forall y \in Y.$$

Equivalently,

$$\widetilde{\omega_{YZ}}(y) = \arg\min_{z \in Z} d_E(y, z), \quad \forall y \in Y.$$

This means $\omega_{YZ} = \widetilde{\omega_{YZ}}$, which completes the proof. □

Let $\mathcal{M}$ be the set containing all point clouds in $\mathbb{R}^D$ with N distinct points. $\mathcal{M}$, together with the metric topology induced by 2-Wasserstein distance, forms a manifold of dimension $D \times N$.

Proof. By Proposition B.4, for every $X \in \mathcal{M}$ one can find a neighborhood $B_W(X, \varepsilon_X)$ and an injective mapping $\xi_X : B_W(X, \varepsilon_X) \to \mathbb{R}^{D \times N}$ satisfying

$$d_W(Y, Z) = d_E(\xi_X(Y), \xi_X(Z)), \quad \forall Y, Z \in B_W(X, \varepsilon_X).$$

Notice ξ_X is a bijective isometry between $(B_W(X,\varepsilon_X),d_W)$ and $(\xi_X(B_W(X,\varepsilon_X)),d_E)$. Hence, ξ_X is open and continuous. Consequently, ξ_X is a homeomorphism, and $\{\xi_X : B_W(X,\varepsilon_X) \to \mathbb{R}^{D\times N}\}_{X\in\mathcal{M}}$, serving as an atlas, endows $\mathcal{M}$ with the manifold structure. □

B.7.1.2 *Riemannian metric structure*

Next we introduce the Riemannian metric structure for the manifold of point clouds, and show the distance induced by the Riemannian metric coincides with the Wasserstein distance. To this end, consider an alternative definition for the space of point clouds:

$$\mathcal{M}' = \{X\colon [N] \to \mathbb{R}^D \mid X \text{ is injective}\}/\sim_{S_N}.$$

The equivalence relation is defined by:

$$X_1 \sim_{S_N} X_2 \quad \Leftrightarrow \quad \exists \nu \in S_N\colon X_1 \circ \nu = X_2 \quad \Leftrightarrow \quad \mathrm{Im}(X_1) = \mathrm{Im}(X_2).$$

Here S_N denotes the N-symmetric group and $\mathrm{Im}(X)$ denotes the image of mapping X. Two mappings are deemed equivalent when their images are identical. Note the image of each mapping $X : [N] \to \mathbb{R}^D$ is a point cloud in $\mathbb{R}^D$ as we defined earlier, i.e. $\mathrm{Im}(X) \in \mathcal{M}$. In fact, the mapping $\mathcal{M}' \to \mathcal{M}\colon [X] \mapsto \mathrm{Im}(X)$ gives the identification between the original and new definitions for the space of point clouds. For simplicity, we use one representative X to denote the equivalence class $[X]$ and also refer to X's as point clouds.

This definition allows defining smooth curves in the point clouds space. Specifically, a smooth curves in $\mathcal{M}'$ is an element of the following set:

$$\mathcal{C} = \{\gamma\colon [N] \times I \to \mathbb{R}^D \mid \gamma(\cdot,t) \text{ is injective}, \forall t \in I; \gamma(k,\cdot) \in C^\infty(I), \forall k \in [N]\}/\sim_{S_N}.$$

Here I denote the closed unit interval $[0,1]$ and the equivalence relation is defined by

$$\gamma_1 \sim_{S_N} \gamma_2 \quad \Leftrightarrow \quad \exists \nu \in S_N\colon \gamma_1(\nu(\cdot),\cdot)) = \gamma_2(\cdot,\cdot)) \quad \Leftrightarrow \quad \mathrm{Im}(\gamma_1(\cdot,t)) = \mathrm{Im}(\gamma_2(\cdot,t)), \forall t.$$

A curve in point cloud space $[\gamma] \in \mathcal{C}$ is essentially a collection of N curves in $\mathbb{R}^D$ such that at each time t the N points on those curves form a point cloud (see Figure B.16). For simplicity we use γ to denote the equivalence class $[\gamma]$.

The *tangent space* at point cloud X is defined as

$$T_X\mathcal{M}' = \{V\,|_{\mathrm{Im}(X)}\colon \mathrm{Im}(X) \to \mathbb{R}^D \mid V \text{ a vector field on } \mathbb{R}^D\}.$$

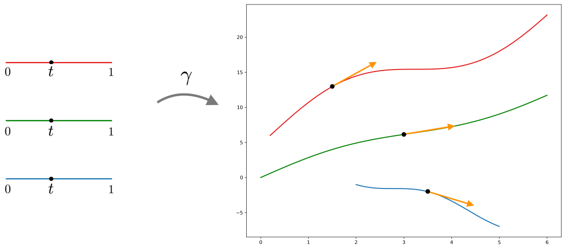


Figure B.16.: A curve in the point cloud space $[\gamma] \in \mathcal{C}$, which maps the set $\{1,2,3\} \times [0,1]$ (left) to the plane $\mathbb{R}^2$ (right). In the right plot, the collection of black dots represent the image of γ at a specific time $t \in [0,1]$, which forms a point cloud in $\mathbb{R}^2$; the collection of orange arrows represents the velocity tangent vector at γ_t along the curve γ.

A *Riemannian metric* is a manifold structure that smoothly assigns a positive-definite inner product $g_X(\cdot,\cdot)$ on the tangent space $T_X\mathcal{M}$ at each $X \in \mathcal{M}$. We introduce a Riemannian metric for $\mathcal{M}$ as follows: $T_X\mathcal{M}'$:

$$g_X(V,W) = \sum_{x \in \mathrm{Im}(X)} \langle V(x), W(x) \rangle.$$

Endowed with this Riemannian metric, $\mathcal{M}'$ is a Riemannian manifold, denoted by $(\mathcal{M}', g)$. We now discuss the distance induced by this Riemannian metric. For a curve in point cloud space $\gamma \in \mathcal{C}$, the *velocity tangent vector* at γ_t along curve γ is defined as (see an example in the right panel in Figure B.16):

$$\dot{\gamma}_t \colon \mathrm{Im}(\gamma_t) \to \mathbb{R}^D; \quad x \to \partial_t \gamma(k_x, t).$$

Here $k_x \in [N]$ is the preimage of point x under γ_t, i.e., $\gamma_t(k_x) = x$. The *length of curve* is given by the length functional L:

$$L \colon \mathcal{C} \to \mathbb{R}; \quad \gamma \mapsto \int_I \sqrt{g_{\gamma_t}(\dot{\gamma}_t, \dot{\gamma}_t)}dt.$$

The distance induced by the Riemannian metric is the minimal length of curves between two points on the manifold.

Definition B.5 (Riemannian distance)**.** Let $(\mathcal{M}, g)$ be a Riemannian manifold. The Riemannian distance between two points $X, Y \in \mathcal{M}$ is defined as

$$d_g(X,Y) = \inf\{L(\gamma) : \gamma \text{ a smooth curve in } \mathcal{M} \text{ connecting } X \text{ and } Y\}.$$

We point out the Riemannian distance coincides with the Wasserstein distance. To see this, we need the following result by [Lemma 2.3, Chap. 9, 103].

Lemma B.1 ([103]). *Let X_0, X_1 be two points in Riemannian manifold $\mathcal{M}$, and γ a curve joining X_0 to X_1. Then γ minimizes the length functional if and only if γ minimizes the energy functional defined as follows:*

$$E\colon \mathcal{C} \to \mathbb{R}; \quad \gamma \mapsto \int_I g_{\gamma_t}(\dot{\gamma}_t, \dot{\gamma}_t) dt.$$

Moreover, when γ is the minimizer, $L(\gamma) = \sqrt{E(\gamma)}$.

Now we proceed to introduce the main statement of this subsection.

Proposition B.6. *Let $(\mathcal{M}', g)$ be the Riemannian manifold of point clouds. Then the Riemannian distance is equivalent to Wasserstein distance, $d_g = d_W$.*

Proof. Consider two point clouds $X_0, X_1 \in \mathcal{M}'$. Let L be the length functional, and E be the energy functional of all curves connecting X_0 and X_1. For any curve γ joining X_0 to X_1, we have

$$\begin{aligned} E(\gamma) &= \int_I g_{\gamma_t}\left(\dot{\gamma}_t, \dot{\gamma}_t\right) dt \\ &= \int_I \sum_{x \in \operatorname{Im}(\gamma_t)} \langle \dot{\gamma}_t(x), \dot{\gamma}_t(x) \rangle dt \\ &= \int_I \sum_{x \in \operatorname{Im}(\gamma_t)} \|\partial_t \gamma(k_x, t)\|^2 dt \\ &= \int_I \sum_{k=1}^{N} \|\partial_t \gamma(k, t)\|^2 dt \\ &= \sum_{k=1}^{N} \int_I \|\partial_t \gamma(k, t)\|^2 dt \\ &= \sum_{k=1}^{N} E[\gamma(k, \cdot)]. \end{aligned}$$

The above equations along with Lemma B.1 indicate the minimizer of the length functional L coincides with the minimizer of $\sum_{k=1}^{N} E(\gamma(k, \cdot))$. Recall for each k, $\gamma(k, \cdot)$ is a curve in $\mathbb{R}^D$ joining $X_0(k)$ to $X_1(k)$. In Euclidean spaces, it is known that straight lines minimize the length functional, which implies, by Lemma B.1, they also minimize $E(\gamma(k, \cdot))$. Specifically for fixed end points $X_0(k)$ and $X_1(k)$, $E(\gamma(k, \cdot))$ has minimal value $d_E^2(X_0(k), X_1(k))$. Therefore, finding the minimizer of $\sum_{k=1}^{N} E(\gamma(k, \cdot))$ is equivalent to finding the bijection between $\operatorname{Im}(X_0)$ and $\operatorname{Im}(X_1)$ that produces the minimal value of the sum of squared distances between points paired by the bijection. This exactly coincides with the optimal transport problem. In conclusion, we have

$$\begin{aligned} d_g(X_0, X_1) &= \min_{\gamma} L(\gamma) \\ &= \min_{\gamma} \sqrt{E(\gamma)} \\ &= \min \left(\sum_{k=1}^{N} E[\gamma(k, \cdot)] \right)^{\frac{1}{2}} \\ &= \min_{\omega \in \Omega(\mathrm{Im}(X_0), \mathrm{Im}(X_1))} \left(\sum_{x \in \mathrm{Im}(X_0)} d^2(x, \omega(x)) \right)^{\frac{1}{2}} \\ &= d_W(\mathrm{Im}(X_0), \mathrm{Im}(X_1)) \end{aligned}$$

This is what was claimed. □

B.7.1.3 *Vector fields*

In this section, we provide definitions for perturbation vector fields and gradient vector fields for the manifold of point clouds.

Assume $\pi : \mathcal{M} \to \mathcal{M}$ is a perturbation mapping. As introduced in Section B.3.2, the perturbation vector fields V_π in the Euclidean space is defined as:

$$V_\pi(X) = \pi(X) - X, \quad \forall X \in \mathcal{M}.$$

The main idea behind this definition is that each tangent vector $V_\pi(X)$ specifies the direction of the straight line connecting X and $\pi(X)$. For general Riemannian manifolds, the notion of straight lines is generalized by minimizing geodesics. Formally, the curve between two points on manifold that minimizes the length functional is a *minimizing geodesic*. Then the question arises whether there exists a minimizing geodesic between any two points on the manifold. To address this, we introduce the concepts of geodesically completeness and the Hopf-Rinow theorem.

As introduced in Appendix B.7.1.2, the Riemannian metric induces a distance d_g on the manifold $\mathcal{M}$. A sequence $\{X_i\}_{i \in \mathbb{Z}^+}$ of points on $(\mathcal{M}, d_g)$ is a d_g-*Cauchy sequence* if for any positive number ε there exists a positive integer N such that $d_g(X_i, X_j) < \varepsilon, \forall i, j > N$.

Definition B.7 (Geodesically complete manifold)**.** The Riemannian manifold $(\mathcal{M}, g)$ is geodesically complete if any d_g-Cauchy sequence $\{X_i\}_{i\in\mathbb{Z}^+}$ converges in $\mathcal{M}$: $\exists Y \in \mathcal{M}$ such that $\lim_{i\to\infty} d_g(X_i, Y) = 0$.

Since the Euclidean space $\mathbb{R}^D$ is a complete metric space, automatically the Riemannian manifold of point clouds is geodesically complete. Also notice this manifold is connected, since there exists a path connecting any two point clouds. The Hopf-Rinow theorem ensures the existence of minimizing geodesics between any two point clouds.

Theorem B.8 (Hopf-Rinow theorem)**.** *Let $(\mathcal{M}, g)$ be a connected Riemannian manifold. If $(\mathcal{M}, d_g)$ is geodesically complete, there exists a minimizing geodesic between any two points on $\mathcal{M}$.*

Now we formally define the perturbation vector fields.

Definition B.9 (Perturbation vector field)**.** Let $(\mathcal{M}, g)$ be a geodesically complete manifold, and $\pi\colon \mathcal{M} \to \mathcal{M}$ be a perturbation mapping. The perturbation vector field V_π is defined as

$$V_\pi\colon \mathcal{M} \to \sqcup_X T_X\mathcal{M}; \quad X \mapsto V_\pi(X) = \dot{\gamma}_{X,\pi(X)}(0),$$

where $\gamma_{X,\pi(X)}$ is a minimizing geodesic $\gamma_{X,\pi(X)}\colon I \to \mathcal{M}$ with $\gamma_{X,\pi(X)}(0) = X$ and $\gamma_{X,\pi(X)}(1) = \pi(X)$.

We proceed to define the Riemannian gradient vector field. Assume ρ is a real-valued smooth function on $\mathcal{M}$. In the cases of Euclidean spaces, the gradient vector can be characterized by the following property:

$$\langle \nabla\rho(X), v\rangle = \frac{\partial}{\partial v}\rho, \quad \forall X \in \mathbb{R}^m, v \in T_X\mathbb{R}^m = \mathbb{R}^m.$$

For general Riemannian manifolds, the notion of directional derivatives is generalized by derivations.

Definition B.10 (Derivative)**.** Let $\mathcal{M}$ be a manifold, and $C^\infty(\mathcal{M})$ be the space of smooth functions on $\mathcal{M}$. A derivative at $X \in \mathcal{M}$ is a linear map $\partial\colon C^\infty(\mathcal{M}) \to \mathbb{R}$ satisfying the Leibniz identity:

$$\partial(fg) = \partial(f)\cdot g(X) + \partial(g)\cdot f(X).$$

For a fixed point $X \in \mathcal{M}$, it turns out that each tangent vector $v \in T_X\mathcal{M}$ can be uniquely associated with a derivative, denoted by ∂_v, in the sense that $\partial_v(\rho)$ measures the rate of change of the function value $\rho(X)$, moving through X with the velocity specified by v. Detailed discussion regarding the equivalence between tangent vectors and derivations can be found in [Chap. 8, 336]. Now we provide the definition of the Riemannian gradient.

Definition B.11 (Gradient vector field)**.** Let $(\mathcal{M}, g)$ be a Riemannian manifold, and $\rho\colon \mathcal{M} \to \mathbb{R}$ a smooth function. The gradient vector field of ρ, denoted by $\nabla\rho$, is defined as the vector field

$$\nabla\rho\colon \mathcal{M} \to \sqcup_X T_X\mathcal{M}; \quad X \mapsto \nabla\rho(X)$$

satisfying the property:

$$g_X(\nabla\rho(X), v) = \partial_v(\rho), \quad \forall X \in \mathcal{M}, v \in T_X\mathcal{M}.$$

Note that Definition B.9 and Definition B.11 are applicable to other types of data, provided that the data space can be equipped with a geodesically complete Riemannian manifold structure.

B.7.1.4 *Pull-back metric*

In this section, we provide definitions for pull-back metric for general encoding mappings between Riemannian manifolds.

Let $(\mathcal{M}, g_{\mathcal{M}})$ be the Riemannian manifold for input data, $(\mathcal{N}, g_{\mathcal{N}})$ be the Riemannian manifold for output data, and $f : \mathcal{M} \to \mathcal{N}$ be a differential encoding mapping between the input space and output space. Recall that for any X in $\mathcal{M}$ the Jacobian of f at X, denoted by J_X^f, is a linear mapping between $T_X\mathcal{M}$ and $T_{f(X)}\mathcal{N}$. The *pull-back metric* induced by f, denoted by g^f, is the structure that assigns the following inner-product on the tangent space $T_X\mathcal{M}$ for each $X \in \mathcal{M}$:

$$g^f(V, W) = g_{\mathcal{N}}(J_X^f(V), J_X^f(W)).$$

Then the *pull-back norm* for any tangent vector V at X is defined as

$$\|V\|_f = \sqrt{g^f(V, V)}.$$

Please note that when one considers $\mathcal{N}$ as a vector space and $g_{\mathcal{N}}$ as the Euclidean metric, the above definition for pull-back metric reduces to the one that we introduced in Section B.3.3. Meanwhile, we point out that, while the Riemannian metric on $\mathcal{M}$ does not affect the pull-back metric, it is still necessary in our approach since it's essential in defining the perturbation vector field and gradient vector field (see Definition B.9 and Definition B.11).

B.7.2 *Differentiability of the mapping from point clouds to PIs*

We provide details for the differentiability of the mapping from point cloud data to PIs and computation the Jacobian. We start with a brief review of the PI computation pipeline. Let X be a point cloud and $\mathrm{Cl}(X, R)$ be the clique complex of the R-neighborhood graph. The pipeline for computing k-dimensional PH and constructing PI involves the following steps (Chapter 1):

- Determine a filtration map: $\phi\colon \mathrm{Cl}(X, R) \to \mathbb{R}$. This induces a filtration.
- Pair simplices such that every non-trivial k-dimensional homology class in the filtration is associated to a pair of simplices (σ_b, σ_d). Roughly, one can think of σ_b as the k-simplex that creates the homology class and σ_b as the $(k+1)$-simplex that trivializes the homology class.
- Obtain the persistence diagram PD, as a multiset of birth-death values $(\phi(\sigma_b), \phi(\sigma_d))$ and transform PD into a multiset $\eta(\mathrm{PD})$ of birth-lifespan pairs $(\phi(\sigma_b), \phi(\sigma_d) - \phi(\sigma_b))$.
- Choose a smooth kernel $g_{(b,l)}$, e.g., Gaussian kernel, and a smooth weighting function $\alpha(b, l)$, e.g., linear function of l, and construct the persistence surface ψ:
$$\psi(x, y) = \sum_{(b,l)\in\eta(\mathrm{PD})} \alpha(b, l) g_{(b,l)}(x, y).$$
- Obtain the PI in the form of a $P \times P$ matrix whose (i, j)-th entry is:
$$\mathrm{PI}_{ij} = \int_{\mathrm{pixel}_{ij}} \psi(x, y).$$

  Here $\cup_{ij}\mathrm{pixel}_{ij}$ forms a grid subdivision of a subdomain of ψ. In our experiments, we consider a rectangle subdomain $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ and evenly-spaced rectangle pixels. Specifically, consider grid points $x_k = x_{\min} + k\Delta x$ for $k = 0, 1, \cdots, P$, where $\Delta x = \frac{x_{\max} - x_{\min}}{P}$, and $y_l = y_{\min} + l\Delta y$ for $l = 0, 1, \cdots, P$, where $\Delta y = \frac{y_{\max} - y_{\min}}{P}$. Set $\mathrm{pixel}_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ for $i = 0, 1, \cdots, P-1$ and $j = 0, 1, \cdots, P-1$. We then estimate the integral value by
$$\mathrm{PI}_{ij} = s \cdot \psi(x_i, y_i),$$
  where s is the area of pixel_{ij}, i.e., $s = \Delta x \Delta y$.

Notice ψ, as a sum of Gaussian kernels, is differentiable with respect to the coordinates of the kernel centers (b, l). Meanwhile, the filtration mapping is differentiable with respect to the coordinate of every point in X. For example, the filtration value in Vietoris-Rips filtration, $\phi(\sigma) = \mathrm{Diam}(\sigma) = \max_{x,y\in\sigma} d_E(x, y)$, is differentiable with respect to the coordinate of any point in X. Hence, the mapping from point cloud data to PI is differentiable.

More technically, as discussed in Appendix B.7.1.1, for every point cloud $X \in \mathcal{M}$, there exists a neighborhood U of X and a coordinate map $\xi\colon U \to \mathbb{R}^{D\times N}$. We claim that the mapping from point clouds to PIs $f\colon \mathcal{M} \to \mathbb{R}^{P\times P}$ is smooth in the sense that the following composition map is smooth:

$$f \circ \xi^{-1}\colon \xi(U) \to \mathbb{R}^{P\times P}.$$

Here we omit the manifold notation for the output space $\mathcal{N}$, given that $\mathcal{N}$ is a submanifold in $\mathbb{R}^{P\times P}$. Specifically, assume $\tau\colon [N] \to X$ is the total order that characterizes $\xi(X)$, i.e., $\xi(X) = [\tau(1), \tau(2), \ldots, \tau(N)]$. For any $(i, j) \in [P] \times [P]$, $k \in [N]$, the partial derivative $\frac{\partial \mathrm{PI}_{ij}}{\partial \tau(k)}$ can be formulated as follows:

$$\begin{aligned}\frac{\partial \mathrm{PI}_{ij}}{\partial \tau(k)} &= \sum_{\sigma} \frac{\partial \mathrm{PI}_{ij}}{\partial \phi(\sigma)} \frac{\partial \phi(\sigma)}{\partial \tau(k)} \\ &= \sum_{\sigma} \frac{\partial s \cdot \psi(x_{ij}, y_{ij})}{\partial \phi(\sigma)} \frac{\partial \phi(\sigma)}{\partial \tau(k)}.\end{aligned}$$

For computing the Jacobian in our experiments, we use the Gudhi library [331] version 3.8.0 and Tensorflow version 2.12.0. Specifically, we use Gudhi library to collect PDs, especially `Gudhi.tensorflow.RipsLayer` class to collect PDs with respect to the Rips filtration. We then manually compute PIs as described above using Tensorflow. Finally, we collect the Jacobian for the whole pipeline with `tensorflow.GradientTape.jacobian` function.

B.7.3 *Visualizing the Jacobian of the encoding over the data manifold*

In this section we visualize the singular value decomposition (SVD) of Jacobian on a toy data set, aiming at providing further intuition for the eigenvectors of Jacobian mappings. We consider point clouds that are uniformly sampled from axis-aligned ellipses of width w and height h. These are illustrated in the left panel of Figure B.17. To visualize the Jacobian at an input point cloud X, we plot the pull-back unit ball around X in the data manifold,

$$B^*(X, 1) = \{v \in T_X\mathcal{M} : \|v\|_f = 1\}.$$

This corresponds to the preimage of a unit ball in $T_{f(X)}\mathcal{N}$. Notably, the equation $1 = \|v\|_f = \sum \lambda_i \langle v, q_i\rangle^2$ indicates that the pull-back unit ball forms an m-dimensional ellipsoid with semi-axes $q_1, q_2, \ldots, q_m$, and the lengths of these semi-axes are given by $\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, \ldots, \frac{1}{\sqrt{\lambda_m}}$.

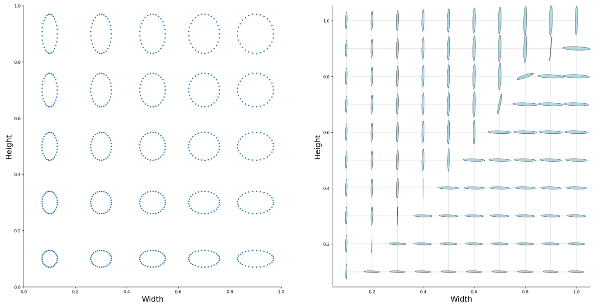


Figure B.17.: Pull-back unit balls (right) on a toy data set of point clouds sampled from ellipses of width w and height h (left). Shorter semi-axes of the pull-back unit ball correspond to larger eigenvalues of the Jacobian.

In the right panel of Figure B.17 we plot the pull-back unit balls for the encoding f given by the 1-dimensional PI with respect to the Vietoris-Rips filtration. The plot reveals that the eigenvectors of the encoding associated with the larger eigenvalue consistently align with the direction of increasing $\min\{w, h\}$. This alignment is in accordance with what we would expect the Vietoris-Rips filtration to capture on this specific data set, since the death value depends on the radius of the inner circumcircle of a hole. Consequently, variations in the length of the major axis have minimal impact on PI. Conversely, altering the length of the minor axis directly affects the death parameter and changes the PI.

B.7.4 *Point saliency maps for PH encodings*

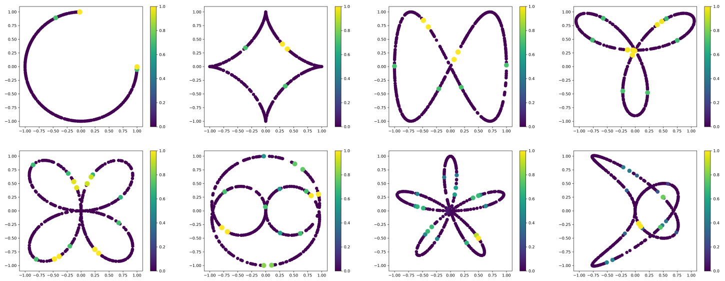


Figure B.18.: Point saliency score with respect to 1-dimensional PIs on Vietoris-Rips filtration for synthetic 2D point clouds.

In the context of interpretability, point saliency maps are commonly used tools to explain the decisions made by trained models [239, 380]. These maps assign importance scores to each point in an input point cloud (or to each pixel in an input image), indicating their significance in relation to the model's prediction. In our study, we employ a similar strategy to visualize the importance of each point in an input point cloud with respect to the PH encoding.

Given a point cloud X and an encoding map f, we define the encoding saliency score for each point $x_i \in X$ as

$$s_f(x_i) = \left\| \frac{\partial f}{\partial x_i} \right\|_F ,$$

where $\| \cdot \|_F$ denotes the Frobenius norm, which is the generalized Euclidean norm for matrices. Here, $\frac{\partial f}{\partial x}$ represents the Jacobian matrix of the encoding, which is a matrix of format $P^2 \times D$ obtained by flattening the PI into a vector in $\mathbb{R}^{P^2}$ and taking the partial derivative of each output pixel with respect to each coordinate of each point in the input point cloud. Similarly, $\frac{\partial f}{\partial x_i}$ is the vector of partial derivatives of the encoding with respect to the coordinates of the i-th point x_i in the point cloud. This score quantifies the sensitivity of the representation to variations on the coordinates of x.

In Figure B.18, we plot the point saliency score with respect to the Rips used in Section B.5 for point clouds sampled from eight synthetic curves in $\mathbb{R}^2$. For each point cloud we highlight individual points according to their saliency scores. We observe that endpoints and points related to the inner circumcycle are often highlighted. These points correspond to the simplices that create/destroy certain homology classes in the filtration. For instance, in the curved rhombus in the first-row, second-column panel, the endpoints (yellow) on the upper right side are highlighted. Note that the edge between these two endpoints will connect the gap and create a homology class. Therefore, variations on these two points can significantly change the birth parameter of the corresponding homology class. Meanwhile, the two points (green) on the upper left and lower right sides are highlighted. They are related to the inner circumcycle of the rhombus, and are vertices of the triangle that destroys the homology class. Therefore variations on these two points can change the death parameter of the homology class significantly.

Before concluding this section, it's important to note that the saliency scores shown in Figure B.18 are heavily influenced by the sampling process. For example, many of the highlighted endpoints shown in Figure B.18 are a result of sparse sampling. Consequently, the resulting saliency scores might be more indicative of the sampling rather than the underlying shape itself. To better comprehend the shape, one potential approach is to compute the average saliency score across multiple different samplings.

B.7.5 *Details on the experiments*

B.7.5.1 *Reproducibility*

The data and code developed for this research are made available at `https://anonymous.4open.science/r/persistent-homology-0915`.

B.7.5.2 *Jacobian normalization*

Throughout our work, we use pull-back norm to quantify the sensitivity of the encoding method to data variations. One needs to be careful when comparing the pull-back norms induced by different encodings, since encodings may have different scaling levels. For example, consider Euclidean data space $\mathcal{M} = \mathbb{R}^2$, and two encoding mappings $f_1 = x + 0.1y$, $f_2 = x + 10y$. The pull-back norms of a tangent vector $v = [1, 0]^T$ with respect to f_1 and f_2 are:

$$\|v\|_{f_1} = \|J^{f_1} v\| = \left\| \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix} \cdot [1, 0]^T \right\| = 1,$$
$$\|v\|_{f_2} = \|J^{f_2} v\| = \left\| \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix} \cdot [1, 0]^T \right\| = 1.$$

In terms of "absolute sensitivity", f_1 and f_2 has the same level of sensitivity to variation v. More specifically, when variation v is applied to a data point X, both $f_1(X)$ and $f_2(X)$ would change with distance approximately 1 in the representation space. However, in terms of "relative sensitivity", f_1 is more sensitive to v. The reason is that for f_1, the vector v is the eigenvector of the Jacobian with the largest eigenvalue; while for f_2, v is the eigenvector with the smallest eigenvalue. Equivalently, for f_1, v has the largest pull-back norm among all tangent vectors with the same norm as v, whereas for f_2, v has the smallest pull-back norm.

In Section B.4.3, our goal is to study and compare the *focus* of different encodings. In Section B.5.1.2 we search for the encoding whose primary *focus* is on the data variations of interest. Hence, in both sections we remove the scaling factor by considering the normalized Jacobian. Specifically, we divide the Jacobian matrix by its largest singular value, $\tilde{J}^f = J^f / \lambda_1^f$. For vector fields V we consider the normalized average pull-back norm:

$$\frac{1}{|\mathcal{D}|} \sum_{X \in \mathcal{D}} \|\tilde{J}^f \cdot V(X)\|.$$

Returning to the previous example, the normalized pull-back norm for vector v with respect to f_1 and f_2 are $\frac{\|J^{f_1}v\|}{\lambda_1^{f_1}} = \frac{1}{1} = 1$ and $\frac{\|J^{f_2}v\|}{\lambda_1^{f_2}} = \frac{1}{10} = 0.1$, respectively.

B.7.5.3 *Identifying what is recognized*

We provide details for the experiments in Section B.4.

RADIAL FREQUENCY PATTERN DATA SET Figure B.19 shows some examples in the Radial Frequency Patterns (RFP) data set $\mathcal{D}_{\text{RFP}}$.

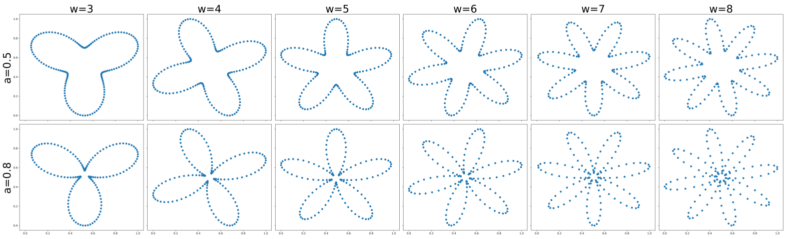


Figure B.19.: Radial Frequency Pattern data.

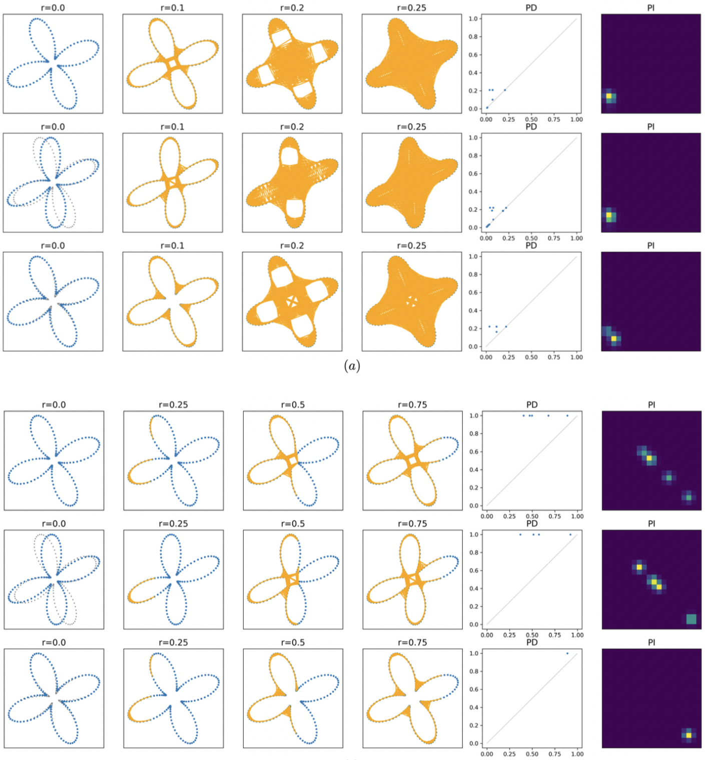
PH PARAMETERS For the Vietoris-Rips filtration, we set *maximal_edge_length* as 1. For the DTM filtration, we set *maximal_edge_length* as 0.5 and parameter m as 0.02. For the Height filtration, we set *maximal_edge_length* as 0.1. We note that we need to set the maximum edge length to a small value to be able to capture topological features of interest (for instance, one wants to avoid connecting different outer regions of petals); alternatively, one could use geodesic distances or cubical complexes, see also [338, Fig.20] for a detailed discussion. Notice there's no need to set a small maximum edge length for Rips and DTM filtrations since the filtration value of an edge in Rips and DTM filtrations takes the distance between its two vertices into account. In contrast, in Height filtration the filtration value of an edge is defined by the maximal filtration value of its vertices, which implies any two points will be immediately connected by an edge between them, if exists, after they appear in the filtration. For the construction of PI, we set the resolution P as 20, variance γ^2 of the Gaussian kernel as 10^{-4}, and the range of the image as $[0,1] \times [0,1]$. The weighting function is set as $\alpha(b,l) = l$. The implementation utilized Tensorflow version 2.12.0 and Gudhi [331] version 3.8.0.

VISUALIZATION OF THE EFFECTS OF PERTURBATION ON PH We visualizes the effects of *shearing* perturbation and *convex* perturbation on the persistent homology with respect to Rips filtration (see the upper panel in Figure B.20) and Height filtration (see the lower panel in Figure B.20). We omit plots associated with DTM filtration as they closely resemble those associated with Rips filtration. Please note that while these visualizations provide insights into the effects of perturbations on PH, they are highly dependent on the specific point clouds under consideration. In fact, the effects of perturbations on PH can vary significantly across different point clouds.

As shown in the upper penal in Figure B.20, the *shearing* changes the sparsity of points in the point clouds and hence changes the birth values (x coordinates) of certain points in PD with respect to Rips filtration. Meanwhile, *shearing* changes the size of petals and hence changes the death values (y coordinates) of certain PD points with respect to Rips filtration. On the other hand, *convex* has the effect of "opening up" the central region of the point clouds. Consequently, some loops appear later in the filtration, notably the five loops that already exist in the second column of the first row but do not show up in the second column of the third row. Consequently, *convex* also induces changes in the birth values of PD points associated with these loops. We note that, in comparison to *shearing*, the *convex* perturbation has a more significant impact on PI. This is consistent with the findings illustrated in Figure B.8, where Rips is more sensitive to *convex* compared to *shearing*.

In the lower panel of Figure B.20, we can observe distinct effects of these two perturbations on the PH associated with the Height filtration. The *shearing* perturbation directly changes the x coordinates of the points in the original points clouds, which correspond to their filtration values, and consequently changes the PI noticeably. On the other hand, the *convex* perturbation changes the PH in a more significant way. Specifically, certain small loops that initially appeared in the filtration (as seen in the first row of the lower panel in Figure B.20) no longer appear in the *entire* filtration after perturbation (as seen in the third row of the lower panel in Figure B.20). This observation arises from the small value we assigned to the *maximal_edge_length* parameter for Height filtration. In fact, for some point clouds, the central points originally have a distance smaller than *maximal_edge_length*. Then *convex* perturbation can pull these central points apart, causing their distance to exceed the *maximal_edge_length* threshold. Consequently, edges connecting these central points vanish from the filtration, resulting in the disappearance of certain small loops. In such cases, as shown in Figure B.20, the *convex* can significantly change the PI associated with Height filtration. However, for other point clouds where the central points initially have a distance greater than *maximal_edge_length*, the corresponding PI may remain relatively unchanged after the *convex* perturbation. This is again consistent with the results shown in Figure B.8, where the pull-back norms of the *convex* perturbation associated with Height filtration exhibit a significant range of variation.

POINTNET ARCHITECTURE AND TRAINING DETAILS In our experiments, we labeled each point cloud data $X \in \mathcal{D}_{\mathrm{RFP}}$ with the number of petals, i.e. the parameter w of the curve $\mathrm{RFP}_{a,w}$ from which X is sampled. This produces 8 classes in total and



(a)

(b)

Figure B.20.: Figure (a) visualizes the effects of different perturbations on persistent homology with respect to Rips filtration. The first four columns illustrate the simplicial complex K_r in the Rips filtration with parameter r ranging from 0.0 to 0.25. The fifth and sixth columns show the corresponding PDs and PIs, respectively. The first row is associated with the original point cloud; while the second and third rows are associated with the point cloud perturbed by *shearing* and *convex*, respectively. Figure (b) visualizes the effects of perturbations on persistent homology with respect to Height filtration, following the same row-column arrangement as in Figure (a).

we train the PointNet model to classify each point cloud in the data set. The PointNet model consists of a 1-dimensional convolutional layer with 64 filters and kernel size 1, followed by batch normalization and rectified linear unit (ReLU) activation. Then global max pooling is applied to obtain a permutation-invariant representation. This is followed by two fully connected layers with 128 and 64 hidden units, respectively, with ReLU activation. The final output layer uses softmax activation to produce theprobability distribution over the output classes. We trained the model with a batch size of 32 for 100 epochs, using a learning rate of 0.001. The optimization algorithm used was Adam, and the model was trained using the cross-entropy loss function. We note data augmentation techniques are used for training PointNet in literatures [see, e.g., 282]. However, we do not augment the data during training, in order to ensure a fair comparison with other encodings. The implementation utilized Tensorflow version 2.12.0.

PERTURBATION VECTOR FIELD ESTIMATION Let $\mathcal{D} = \{X_i\}_{i \in \mathcal{I}}$ be a finite data set of point clouds, and $\pi : \mathcal{M} \to \mathcal{M}$ a perturbation mapping defined on the data manifold. In the case where the data lies in Euclidean space, i.e. $\mathcal{M} = \mathbb{R}^m$, one can compute the perturbation vectors as following:

$$V_\pi(X) = \pi(X) - X, \quad \forall X \in \mathcal{D}.$$

However, in the case of general Riemannian data manifold, the subtraction between any two points on the manifold may not be well-defined. To address this, we control the perturbation mapping such that for every $X \in \mathcal{D}$ the perturbed point cloud $\pi(X)$ lies in a small neighborhood of X and calculate the perturbation vector via the local coordinate system. Specifically, as shown in Appendix B.7.1.1, for each point cloud $X \in \mathcal{D}$, one can find a neighborhood $X \in U_X \subset \mathcal{M}$ and an injective isometry $\xi_X : U_X \to \mathbb{R}^m$. We control the perturbation mapping sends every point X to U_X, i.e.

$$\pi(X) \in U_X, \quad \forall X \in \mathcal{D}.$$

Then the perturbation vectors can computed with the subtraction on the Euclidean domain $\xi_X(U_X)$ as follows:

$$V_\pi(X) = \xi_X(\pi(X)) - \xi_X(X), \quad \forall X \in \mathcal{D}.$$

UNNORMALIZED PULL-BACK NORMS As discussed in Appendix B.7.5.2, we consider the normalized average pull-back norms for perturbation tangent vector fields in Section B.4.3. We present the results of unnormalized average pull-back norms in Figure B.21. Note that one could conclude from Figure B.21 that, on average, Height is more sensitive to *convex* than Rips in terms of "absolute sensitivity". However, we reach the opposite conclusion from Figure B.8 in terms of "relative sensitivity".

B.7.5.4 *Selecting hyperparameters*

We provide further details on Section B.5.

PH PARAMETERS In this section, we focus on PH encoding constructed on Vietoris-Rips filtration. We set the parameter *maximal_edge_legnth* as 0.25, and the range of PI as $[0.0, 0.25] \times [0.0, 0.25]$. The implementation utilized Gudhi [331] version 3.8.0 and Tensorflow version 2.12.0.

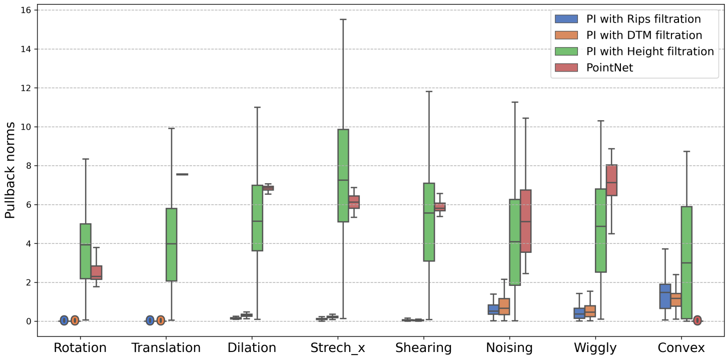


Figure B.21.: Unnormalized average pull-back norm of different perturbation vector fields with respect different encodings.

GRADIENT VECTOR FIELD ESTIMATION bLet $\mathcal{D} = \{X_i\}_{i \in \mathcal{I}}$ be a finite data set of point clouds, and $\rho(X_i)$ be the corresponding feature values of point clouds in $\mathcal{D}$. In the case when the data lies in Euclidean space, we can estimate the gradient vectors with the finite difference method (FDM) as follows:

$$\overline{\nabla \rho(X)} = X' - X, \quad X' = \operatorname*{argmax}_{Y \in \mathcal{D}} \frac{\mid \rho(Y) - \rho(X) \mid}{d_E(X, X')}, \quad \forall X \in \mathcal{D}.$$

For binary categorical feature, i.e. $\rho(X) \in \{0, 1\}$, the formula can be modified as

$$\overline{\nabla \rho(X)} = X' - X, \quad X' = \operatorname*{argmin}_{Y \in \{Z \in \mathcal{D} : \rho(Z) \neq \rho(X)\}} d_E(X, Y), \quad \forall X \in \mathcal{D}.$$

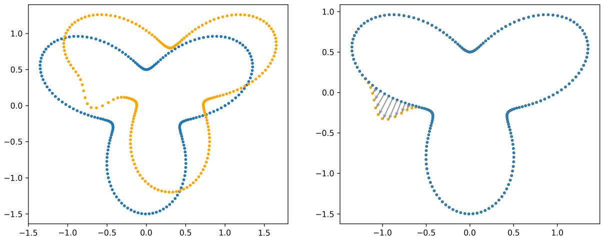


Figure B.22.: The grey vectors (right) represent the difference vector between the blue point cloud and the transformed orange point cloud (left).

However, in the case of general Riemannian data manifold, the subtraction between any two points on $\mathcal{M}$ may not be well-defined. To address this, when estimating gradient vector located at X, we send other point clouds in the data set to a neighborhood of X via transformation that preserves the feature value. In our experiment, we use Euclidean transformation since the *sex* feature is irrelevant to the position or orientation of the brain artery trees. Then FDM can be applied to the transformed data set, via the coordinate system on that neighborhood.

Specifically, we utilize the iterated closest point (ICP) method [32, 74]. Let $E(\mathbb{R}^D)$ be the collection of all Euclidean transformations in $\mathbb{R}^D$

$$E(\mathbb{R}^D) = \{\iota : \mathbb{R}^D \to \mathbb{R}^D : d_E(x,y) = d_E(\iota(x), \iota(x)), \forall x, y \in \mathbb{R}^D\}.$$

Note each ι can be naturally extended to a transformation on point clouds: $\iota(X) \triangleq \{\iota(x) : x \in X\}$. Let ζ be a error function in the sense that $\zeta(X,Y)$ measures the "difference" between X and Y. Given two point clouds X and Y, the ICP algorithm searches the Euclidean transformation that gives the minimal error value:

$$\iota_{X,Y} = \underset{\iota \in E(\mathbb{R}^D)}{\operatorname{argmin}} \zeta(X, \iota(Y)).$$

Here $\iota_{X,Y}$ is Euclidean transformation found by ICP algorithm for point clouds X and Y. Define the ICP discrepancy between point clouds X and Y (not necessarily a distance) as

$$d_{\text{ICP}}(X,Y) = d_W(X, \iota_{X,Y}(Y)),$$

where d_W is Wasserstein distance between point clouds. Let U_X be a neighborhood of X and ξ_X a coordinate map $\xi_X : U_X \to \mathbb{R}^m$. Estimate the gradient vector field for binary categorical feature ρ as follows (see an illustration in Figure B.22):

$$\overline{\nabla\rho(X)} = \xi_X(\iota_{X,X'}(X')) - \xi_X(X), \quad X' = \underset{Y \in \{Z \in \mathcal{D} : \rho(Z) \neq \rho(X)\}}{\operatorname{argmin}} d_{\text{ICP}}(X,Y), \quad \forall X \in \mathcal{D},$$

The implementation utilized python library Open3D [381] version 0.17.0 and POT [124] version 0.9.0.

DOWNSTREAM TASKS AND PERFORMANCE In Section B.5, we fed PIs produced by different parameter settings to logistic regression models to predict the *sex* feature. Specifically, we normalize the PIs such that the pixel values range within $[0,1]$. We implemented logistic regression models using Scikit-learn [267] version 1.2.2 with default hyperparameters. When evaluating the model, we use a 7-fold cross validation. And for robust evaluation in Section B.5.1.3, we add identically and independent distributed Gaussian noise with variance 10^{-2} to each coordinate of each point in input point clouds.

Here, we also investigate the effects of resolution and variance of Gaussian kernel on the downstream performance of convolutional neural network (CNN). The CNN model takes PIs as inputs, then begins with a convolutional layer with 32 filters and a 3 × 3 kernel, followed by a ReLU activation function. Then max pooling with a pool size of 2 × 2 is applied. Subsequently, another convolutional layer with 64 filters and a 3 × 3 kernel is added, also followed by a ReLU activation and max pooling. The resulting outputs are then flattened and passed through a fully connected layer

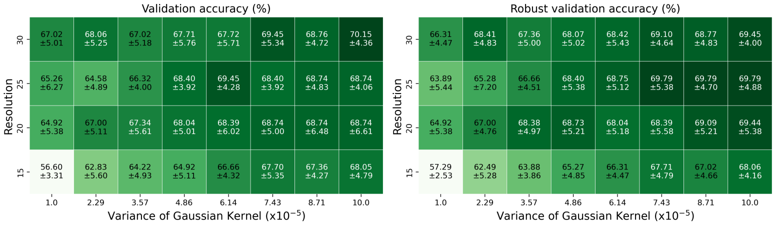


Figure B.23.: Standard (left) and robust (right) test accuracy of the convolutional neural network (CNN) trained on PIs, across different produced by different parameter.

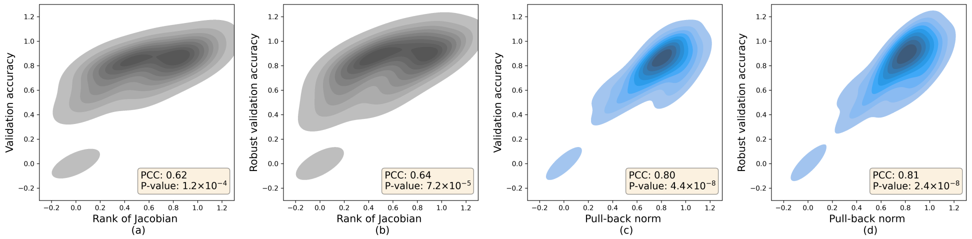


Figure B.24.: Gaussian kernel density estimation of the joint distribution of the rank of the Jacobian or pull-back norm vs. CNN (robust) validation accuracy.

with 64 neurons and ReLU activation. Finally, a single neuron with a sigmoid activation function is used for binary classification. To train the model, we employ Adam optimizer with the cross-entropy loss function. The implementation utilized Tensorflow version 2.12.0.

We represent the downstream performance of the CNN model in Figure B.23. Shown in Figure B.24 is the kernel density estimation of the joint distribution of four pairs of variables: Jacobian rank vs. validation accuracy, Jacobian rank vs. robust validation accuracy, pull-back norm vs. validation accuracy, and pull-back norm vs. robust validation accuracy. Additionally, the Pearson correlation coefficient and p-value of a two-sided test are presented at the lower right corner of each point in Figure B.24. We observe that the correlation between pull-back norms and downstream performances remains significant.

We conjecture that complex models, such as CNNs, are able to obtain good downstream performance even if the average pull-back norm is low, so long as it is not zero. Intuitively, when the feature information is indeed contained in the encoded representation but is not significantly pronounced, a simple model may not be able to extract this information, but a complex model, along with appropriate training techniques, can still learn to extract and utilize this information.

B.7.6 *Investigating which part of the data is highlighted by PH encodings*

In this section, we demonstrate how our method facilitates investigating which *part* of the data is the focus of PH encodings. To this end, we will introduce noising perturbation on different *parts* of the point clouds and examine the average pull-back norm of these perturbation vector fields. Moreover, we also consider the beta weighting function for PIs and investigate the effects of the beta mean parameter on the pull-back geometry, which allows comparing the focus of long and short persistence intervals.

HUMAN BODY DATA We utilize the benchmark mesh segmentation data [73]. This data set consists of meshes representing 19 different types of 3-dimensional shapes, each annotated with manually added segmentation labels. For our analysis, we focus on the subset of meshes representing the human body, which encompasses various gestures such as standing, walking, and sitting. We randomly subsample three point clouds from the vertices of each human body mesh, with each point cloud containing $N = 500$ points.

PH ENCODING We focus on the PH encoding that sends each point cloud to its 2-dimensional PI with respect to the Vietoris-Rips filtration. We choose 2-dimensional PH because we note that the underlying geometric objects of human body meshes are $2-$dimensional surfaces whose 1-dimensional homology is typically zero, and whose reduced 0-dimensional homology is zero as well. For the construction of PIs, we set the PI hyperparameters as $P = 20, \gamma = 1 \times 10^{-4}$, *maximal_edge_legnth*$= 0.3$, and the range of PI as $[0.0, 0.3] \times [0.0, 0.3]$. For the weighting function, we again employ the beta weighting function that we introduced in Section B.5.2. We set the variance parameter for the beta weighting function s^2 as 0.04 and consider the mean parameter k ranging from 0.1 to 0.5. We present a visualization of the impact of the mean parameter on the PIs in Figure B.25.

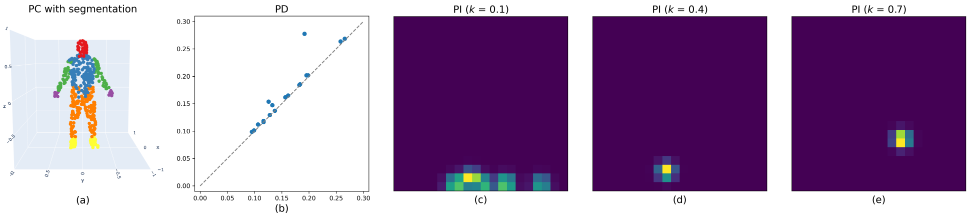


Figure B.25.: An example point cloud from the human body data set, with segmentation of body parts (a), its 2-dimensional persistence diagram with respect to the Vietoris-Rips filtration (b), and persistence images with beta weight function, with the beta mean parameter 0.1, 0.4 and 0.7 (c)-(e).

perturbation We merge the segmentation labels for the sampled human body point clouds into 6 categories: head, torso, arms, hands, legs and feet (see panel (a) in Figure B.25 for an illustration). Accordingly, we consider 6 types of perturbations, each adding independent Gaussian noise to points in one body part. We provide visualizations of some of the perturbations in Figure B.26.

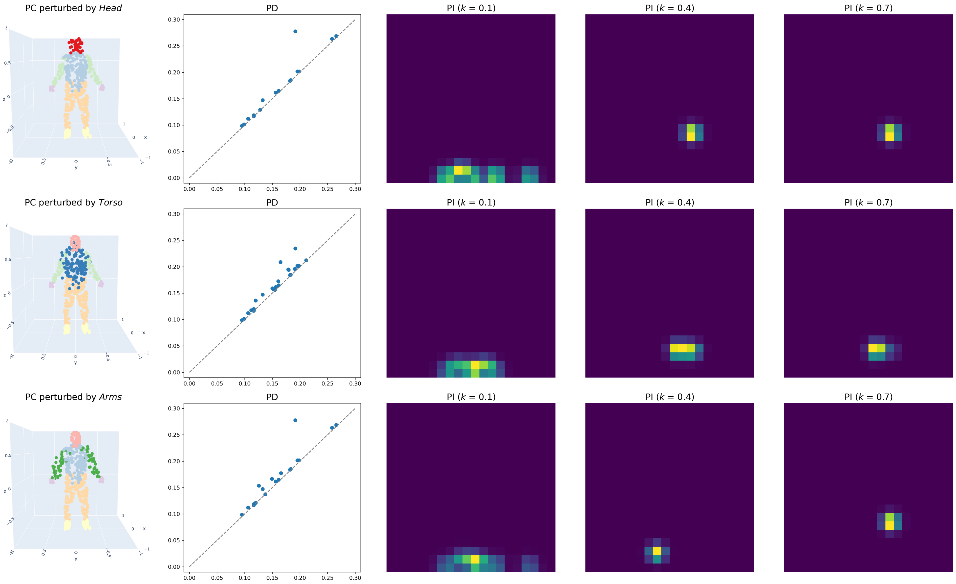


Figure B.26.: Sensitivity of the 2-dimensional PH encoding on the Vietoris-Rips filtration on an example point cloud from human body data set, to perturbations of the *Head*, *Torso*, and *Arms* on the point cloud, across different paramters of the PI beta weight function.

pull-back norm We evaluate the average pull-back norms of the noising perturbation on 6 body parts and present the results in Figure B.27.

It is noticeable in Figure B.27 that persistence intervals of varying lengths capture distinct aspects of the data. Specifically, when the mean parameter is set to 0.1, the encoding exhibits significant sensitivity to perturbations on *legs*, when $k = 0.2$, the focus of the encoding switches to *headF*. For larger mean parameters, the encoding becomes most sensitive to perturbations on *torso*. This could be explained by observing that shorter persistence intervals in 2-dimensional PDs on Rips filtration are more related to smaller hollow shapes in the data, such as arms and head, while longer intervals relate more to larger hollow shapes, such as the torso.

These findings can also guide the selection of hyperparameters for PH encodings. For instance, in a face recognition task, we know from above that shorter persistence intervals are sensitive to data variations on *head*. Hence, one should choose a value around 0.2 for the beta mean parameter in order to obtain a data presentation that is most suitable for this task. At the same time, we note once again that long persistence intervals do not always contain the most important information and that the optimal choice of the weighting function (and other hyperparameters) depend on the specific task at hand.

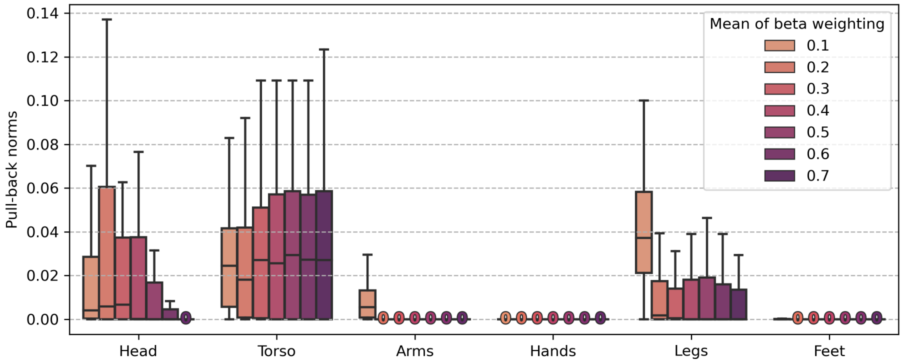


Figure B.27.: Average pull-back of the 2-dimensional PH encoding on the human body data set, across perturbations of different body parts, and different paramters of the PI beta weight function.

C

Topological data analysis of structure preservation via hyperdimensional computing: Additional experimental results

This appendix consists of the supplementary material that accompanies Chapter 4, i.e., the corresponding publication:

> Renata Turkeš, Steven Mortier, Jorg De Winne, Dick Botteldooren, Paul Devos, Steven Latré and Tim Verdonck, *Who is WithMe? EEG features for attention in a visual task, with auditory and rhythmic support*, under review.

In Chapter 4, the detailed experimental results are only shown for an example data set lying on the bumpy circle, and the appendix shows the results for the remaining data.

C.1 Distance preservation

C.1.1 *MDS embedding*

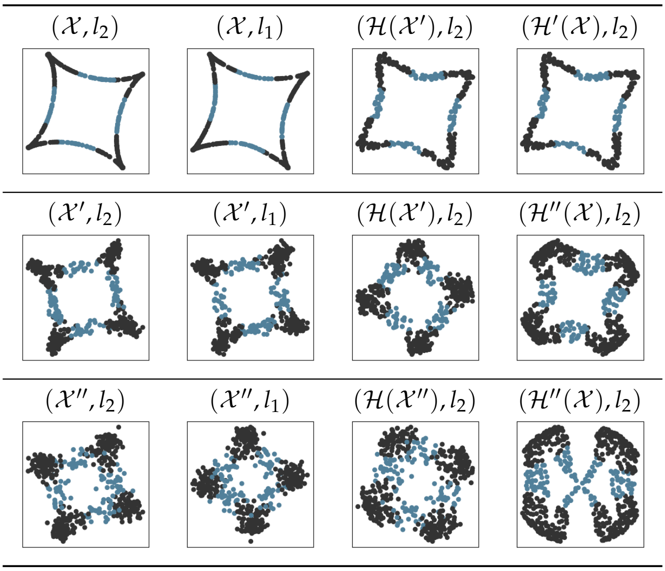


Figure C.1.: MDS embedding of astroid into $\mathbb{R}^2$.

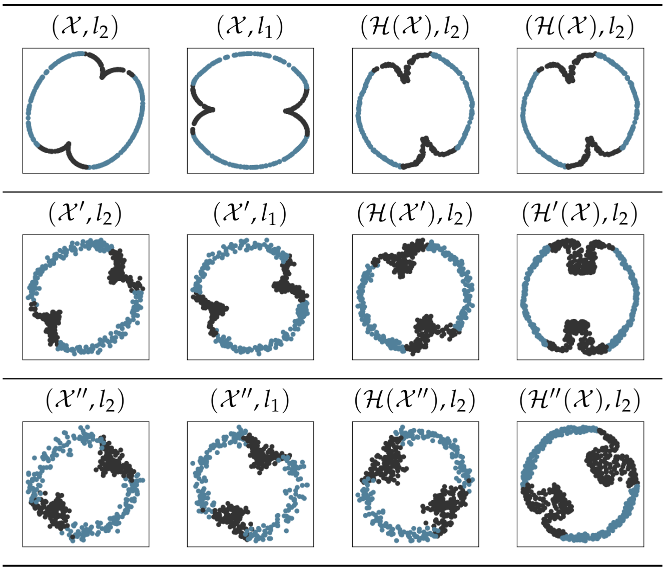


Figure C.2.: MDS embedding of apple into $\mathbb{R}^2$.

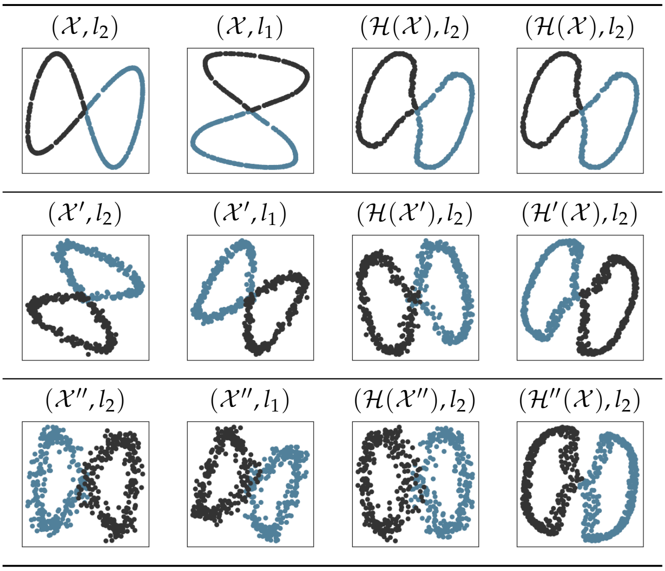


Figure C.3.: MDS embedding of lemniscate into $\mathbb{R}^2$.

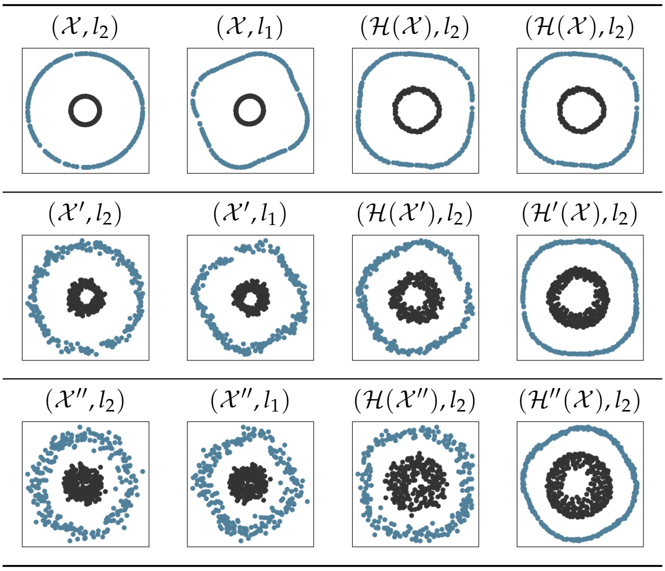


Figure C.4.: MDS embedding of concentric circles into $\mathbb{R}^2$.

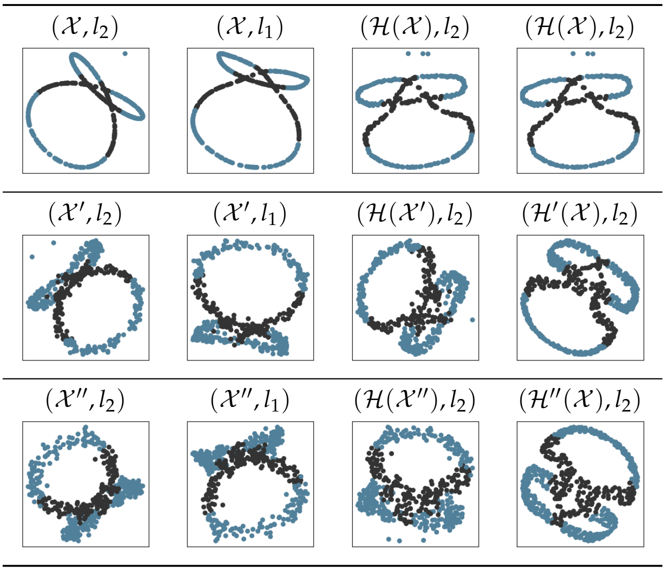


Figure C.5.: MDS embedding of fish into $\mathbb{R}^2$.

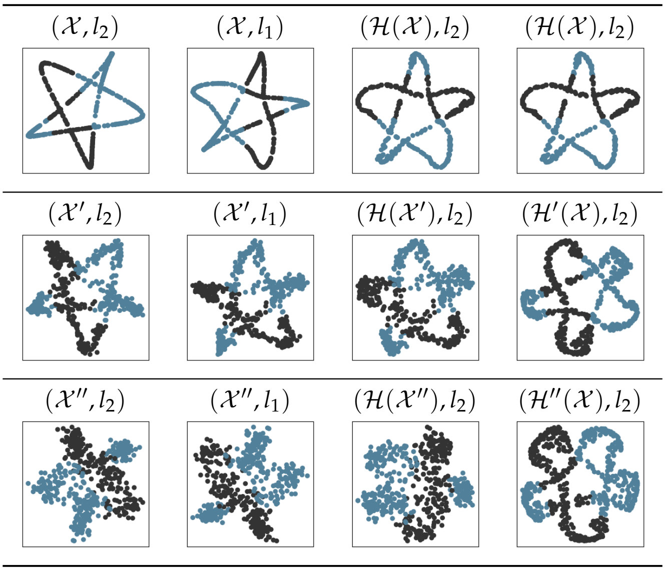


Figure C.6.: MDS embedding of star into $\mathbb{R}^2$.

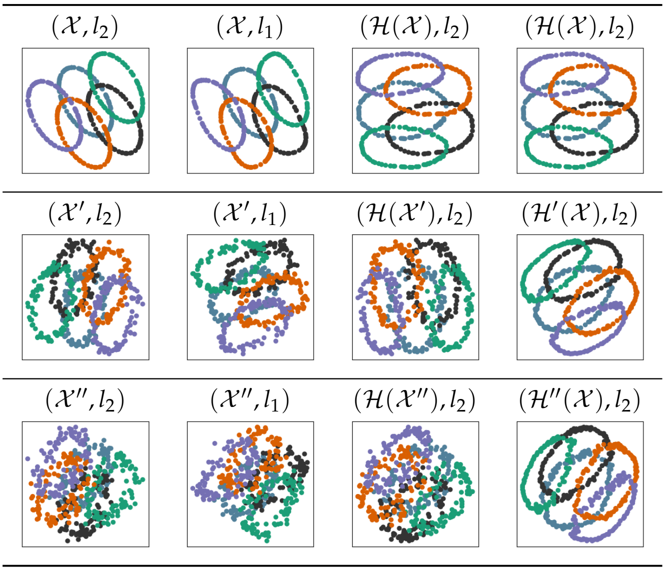


Figure C.7.: MDS embedding of olympics into $\mathbb{R}^2$.

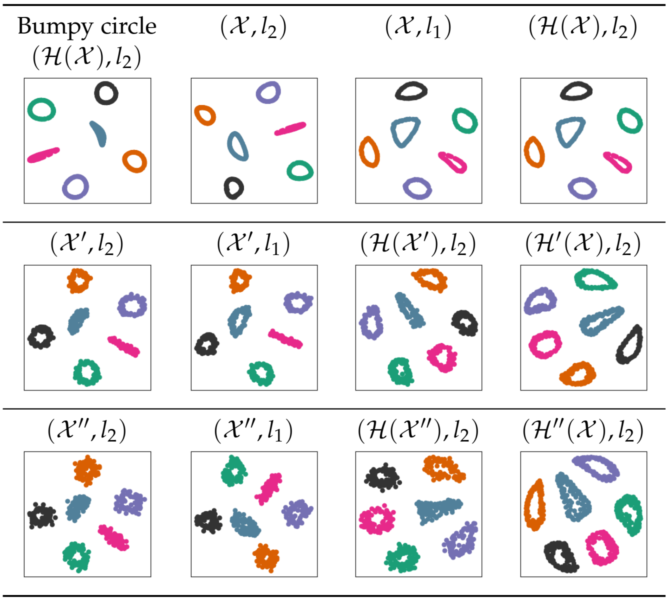


Figure C.8.: MDS embedding of orthogonal circles into $\mathbb{R}^2$.

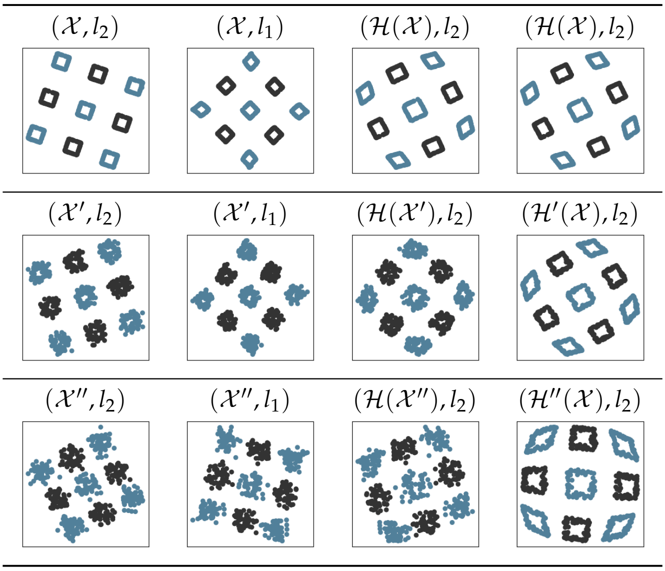


Figure C.9.: MDS embedding of nine squares into $\mathbb{R}^2$.

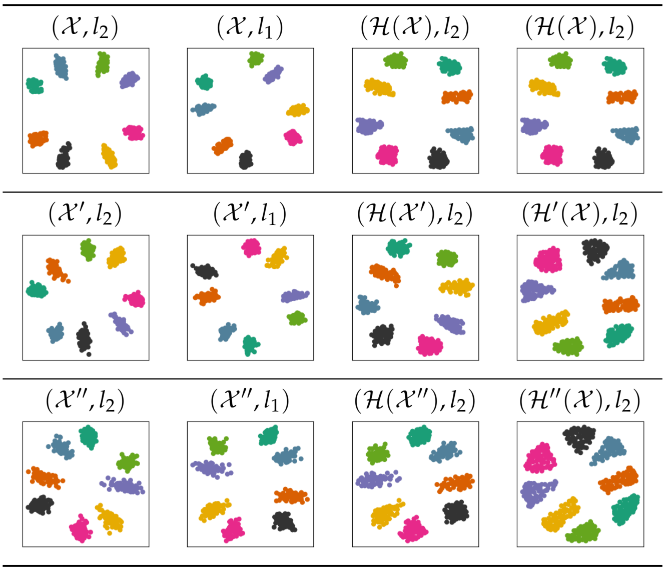


Figure C.10.: MDS embedding of cube vertices into $\mathbb{R}^2$.

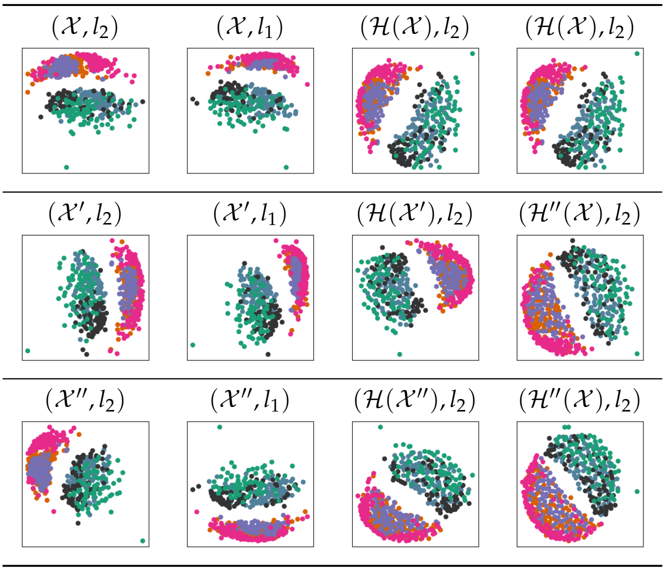


Figure C.11.: MDS embedding of UCI-HAR into $\mathbb{R}^2$.

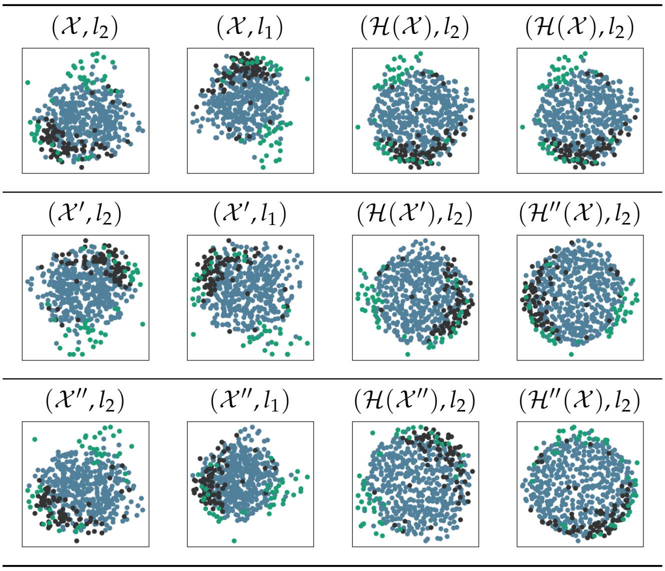


Figure C.12.: MDS embedding of CTG into $\mathbb{R}^2$.

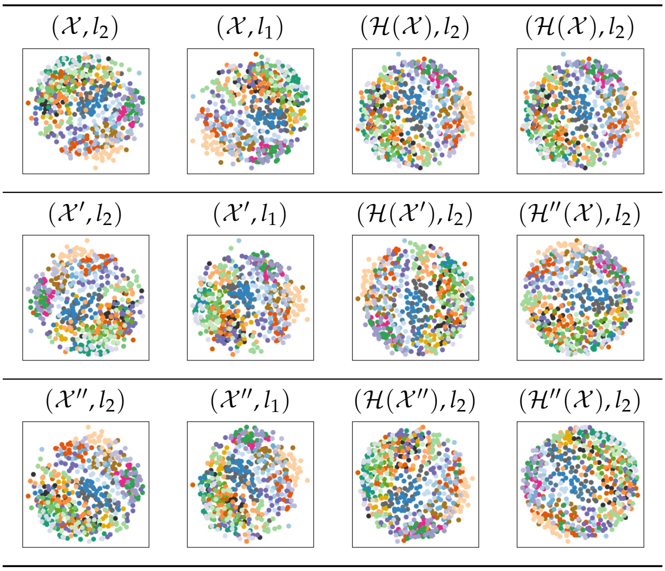
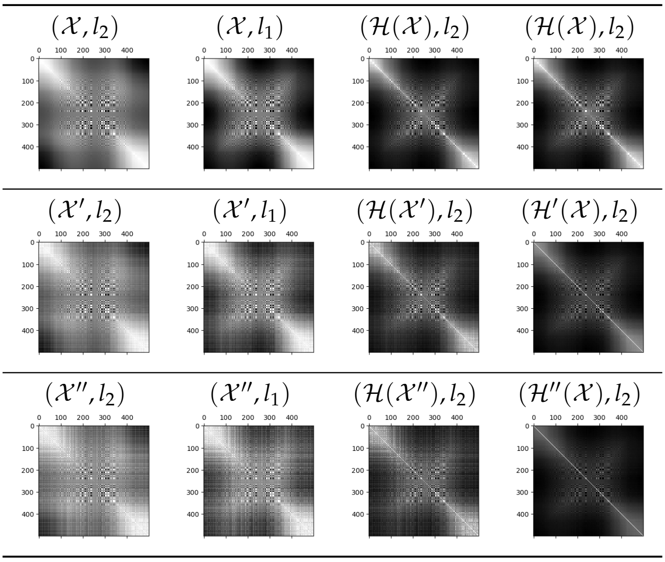


Figure C.13.: MDS embedding of ISOLET into $\mathbb{R}^2$.

c.1.2 *Distance matrix*

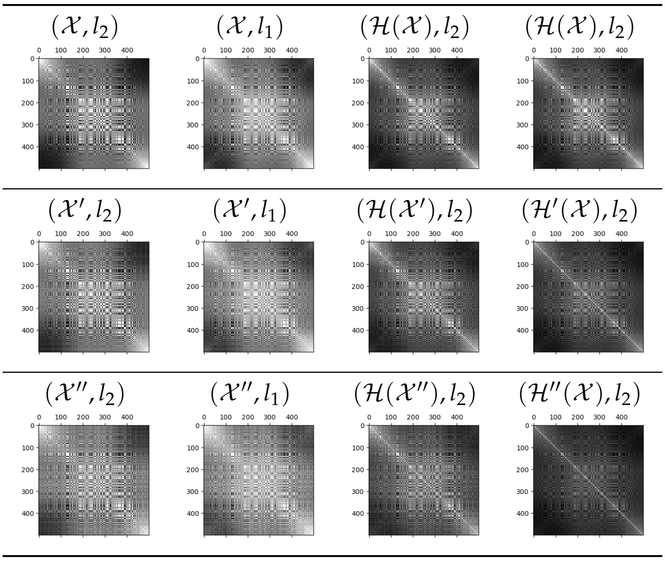


$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.14.: Distance matrices for the astroid.



$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.15.: Distance matrices for the apple.

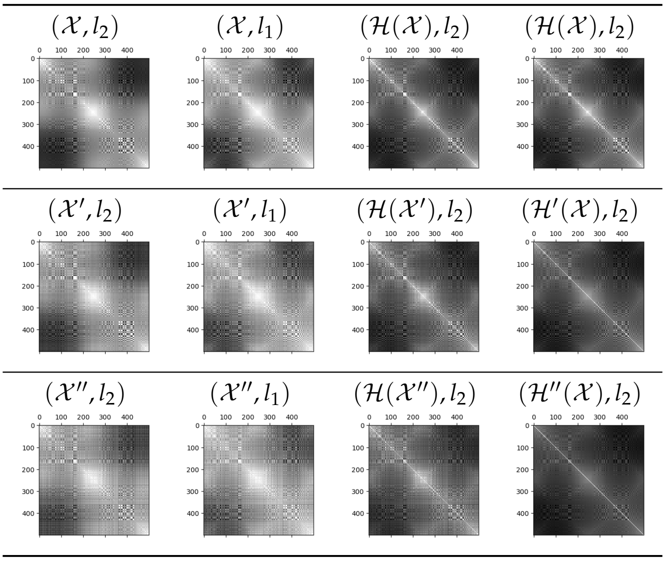


Figure C.16.: Distance matrices for the lemniscate.

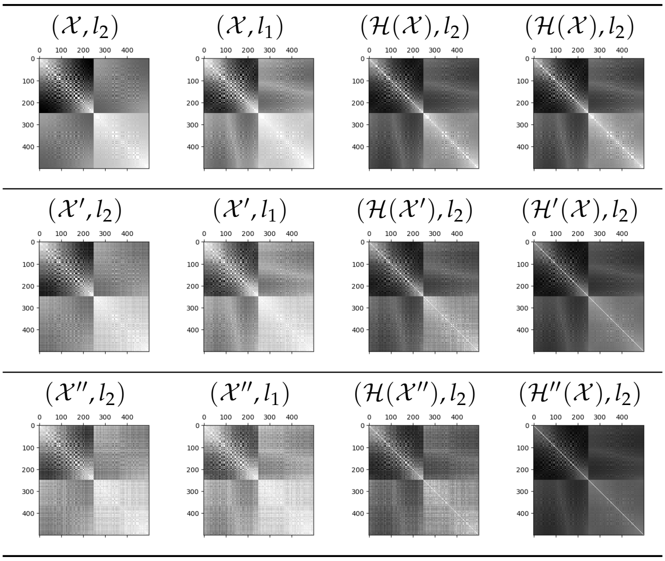


Figure C.17.: Distance matrices for the concentric circles.

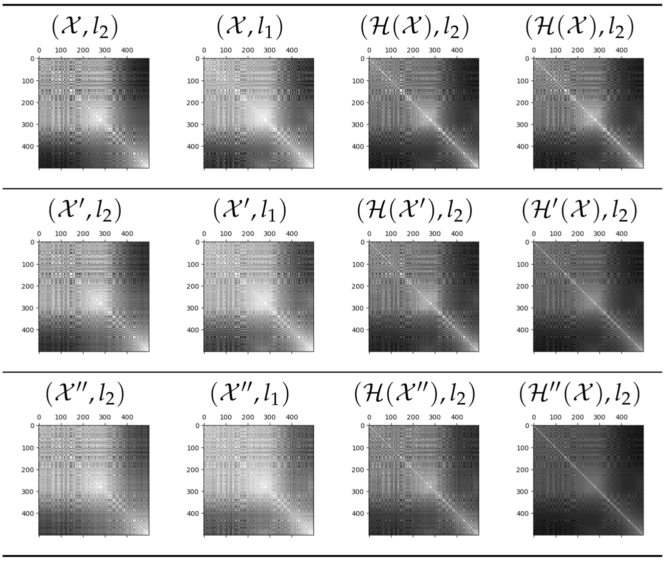


Figure C.18.: Distance matrices for the fish.

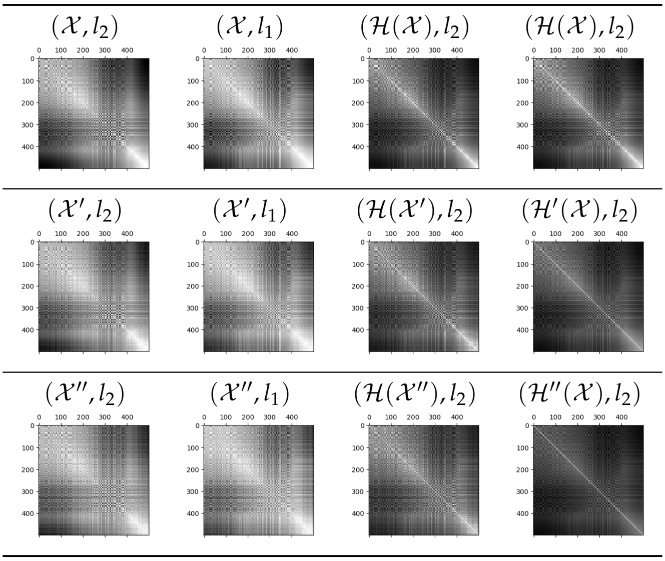


Figure C.19.: Distance matrices for the star.

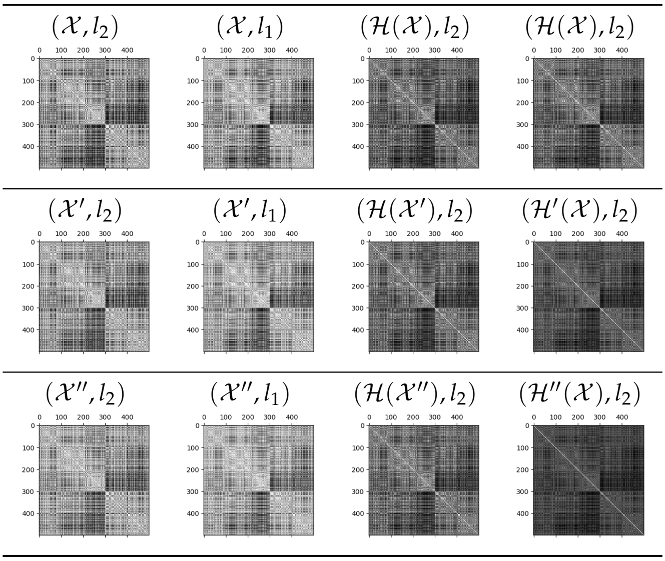


Figure C.20.: Distance matrices for the olympics.

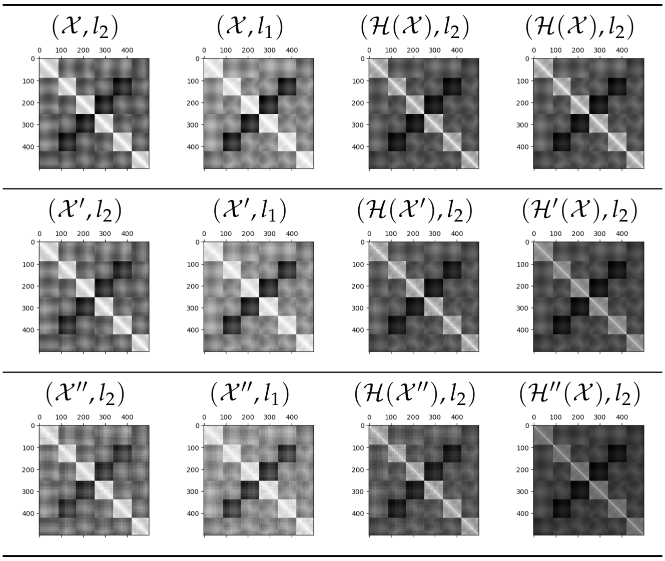


Figure C.21.: Distance matrices for the orthogonal circles.

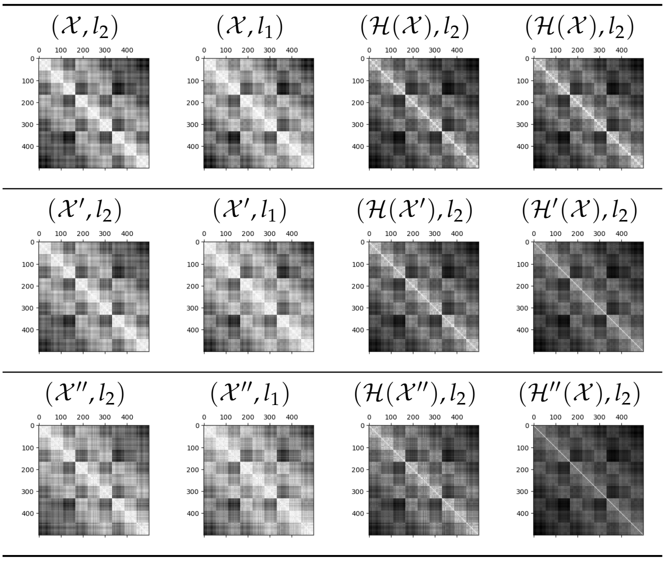


Figure C.22.: Distance matrices for the nine squares.

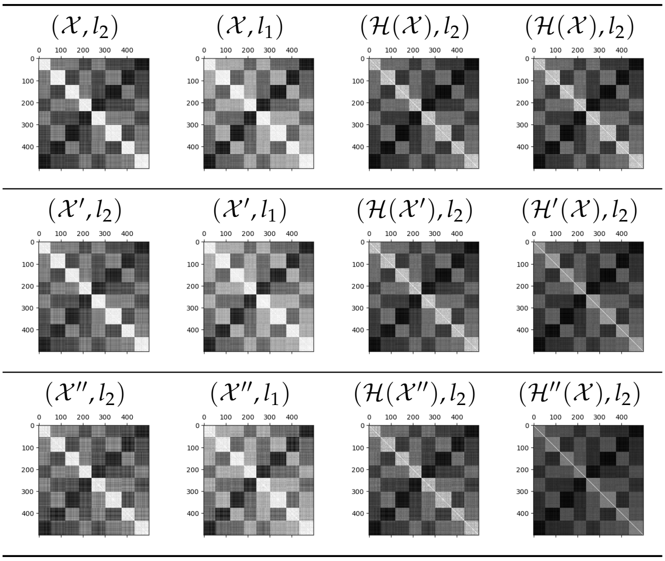


Figure C.23.: Distance matrices for the cube vertices.

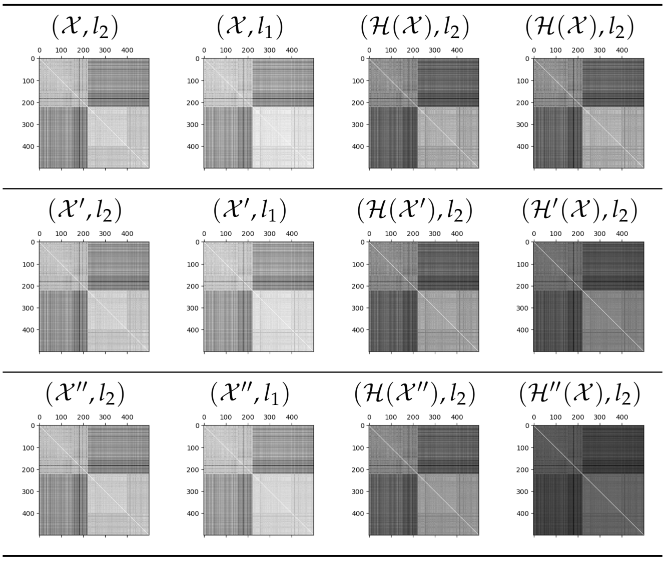


Figure C.24.: Distance matrices for UCI-HAR.

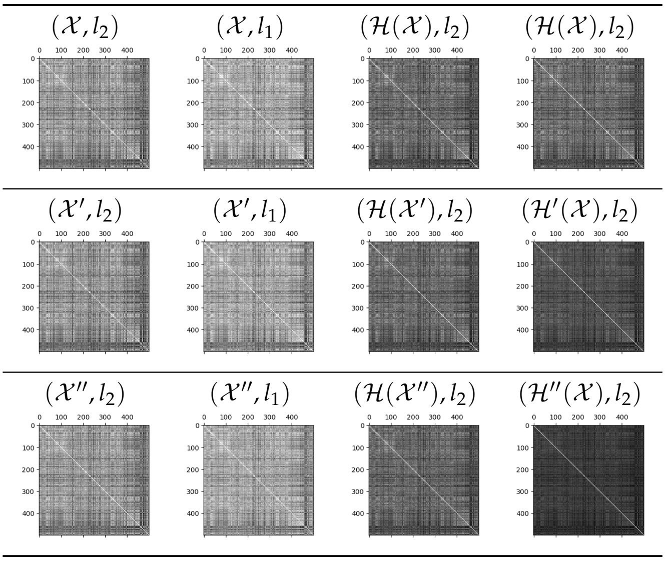


Figure C.25.: Distance matrices for CTG.

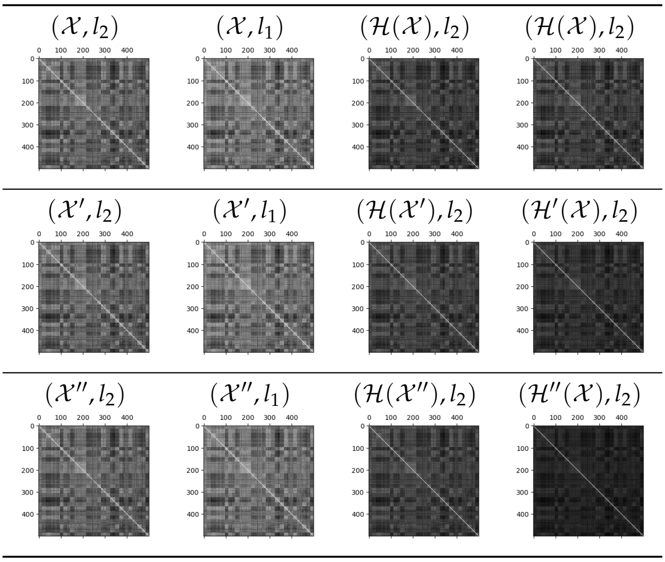


Figure C.26.: Distance matrices for ISOLET.

C.1.3 *Distribution of distances*

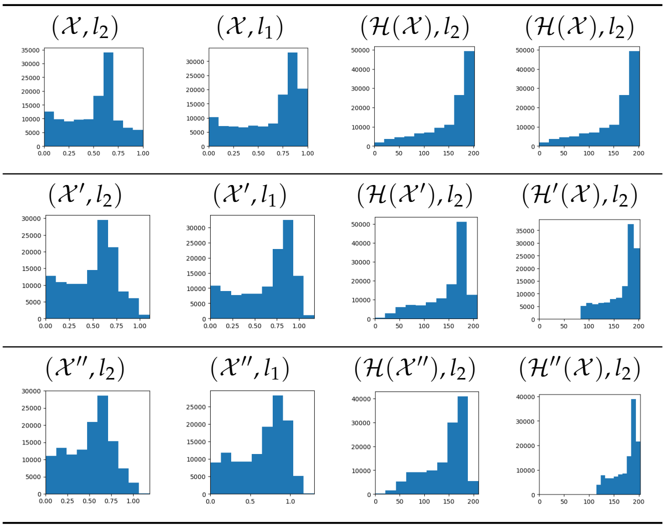


Figure C.27.: Distribution of distances for the astroid.

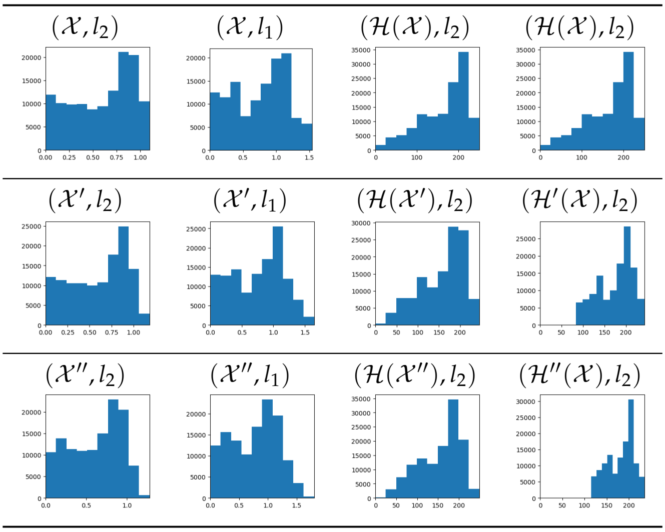


Figure C.28.: Distribution of distances for the apple.

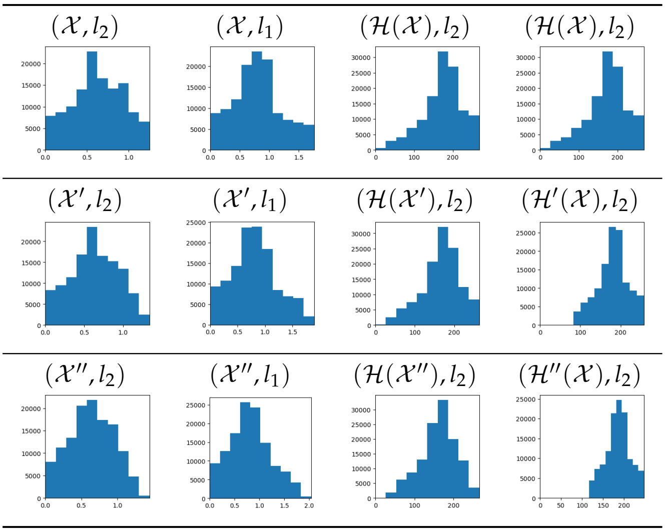


Figure C.29.: Distribution of distances for the lemniscaste.

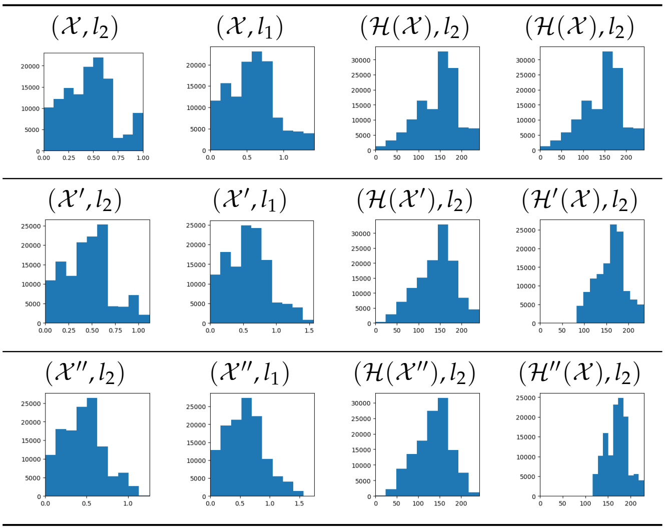


Figure C.30.: Distribution of distances for the concentric circles.

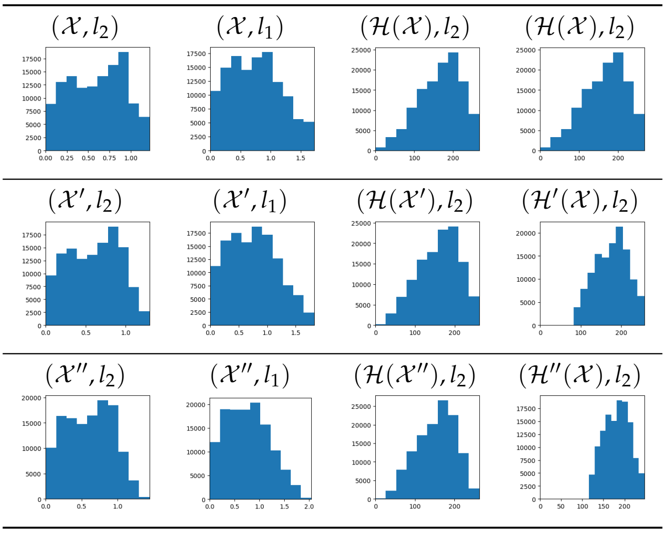


Figure C.31.: Distribution of distances for the fish.

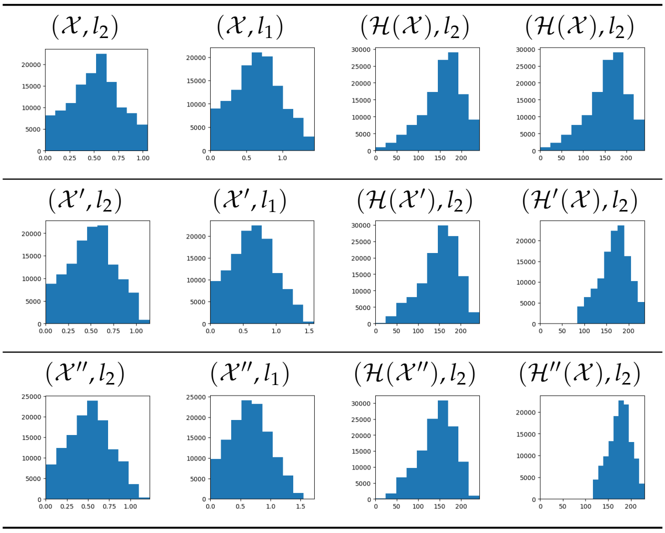


Figure C.32.: Distribution of distances for the star.

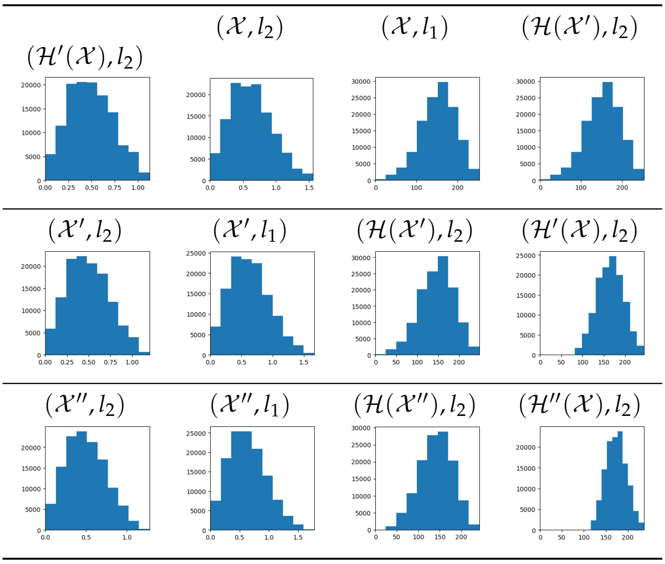


Figure C.33.: Distribution of distances for the olympics.

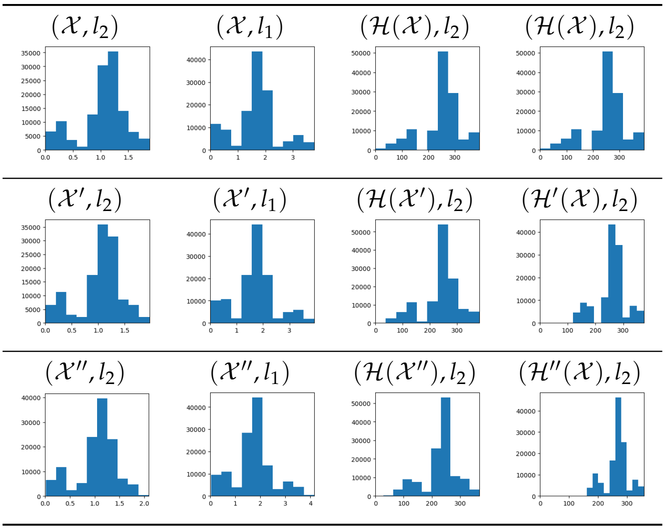


Figure C.34.: Distribution of distances for the orthogonal circles.

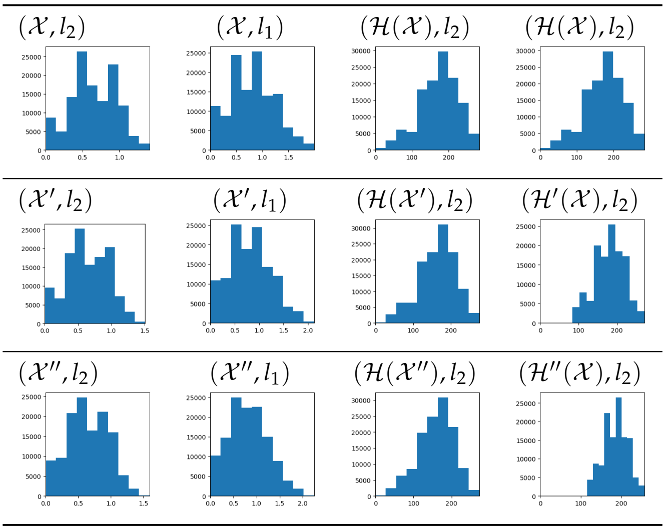


Figure C.35.: Distribution of distances for the nine squares.

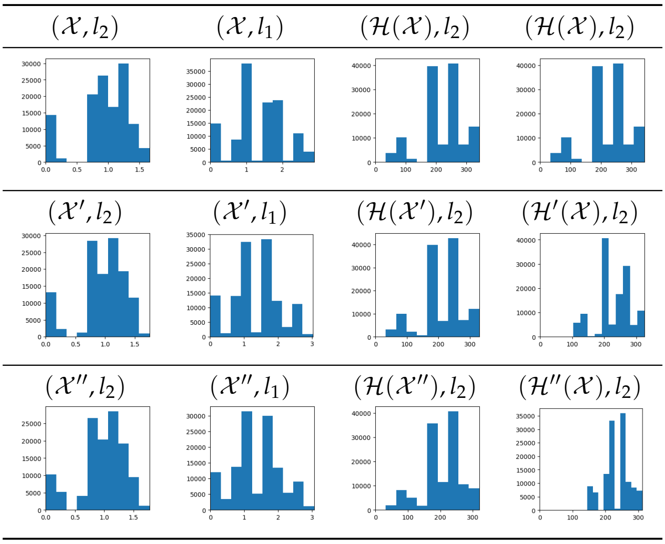


Figure C.36.: Distribution of distances for the cube vertices.

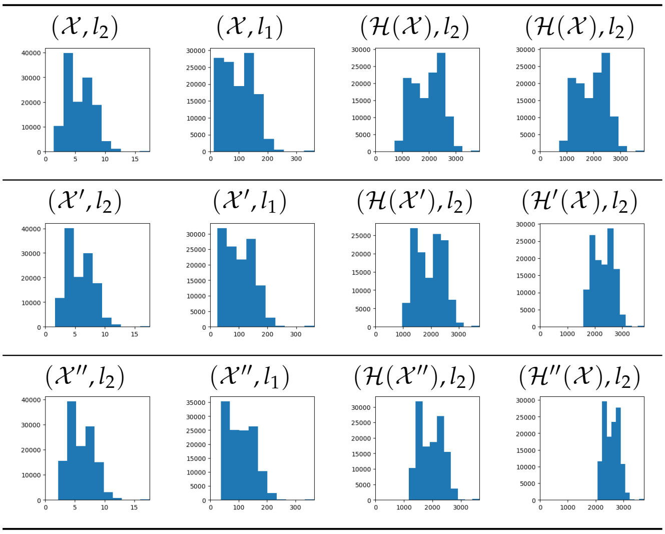


Figure C.37.: Distribution of distances for UCI-HAR.

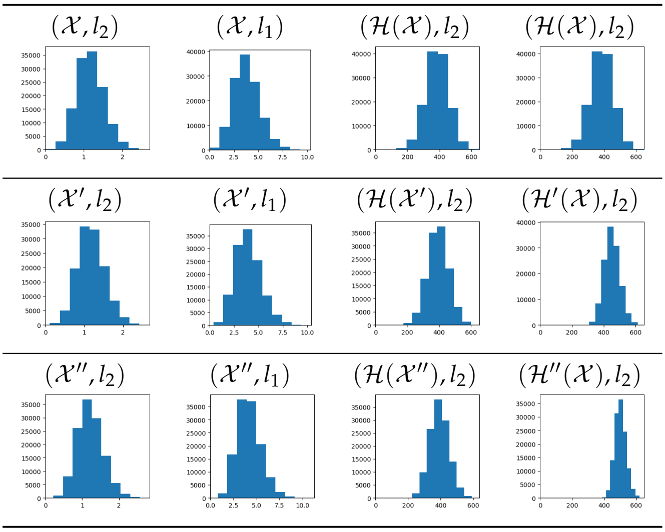


Figure C.38.: Distribution of distances for CTG.

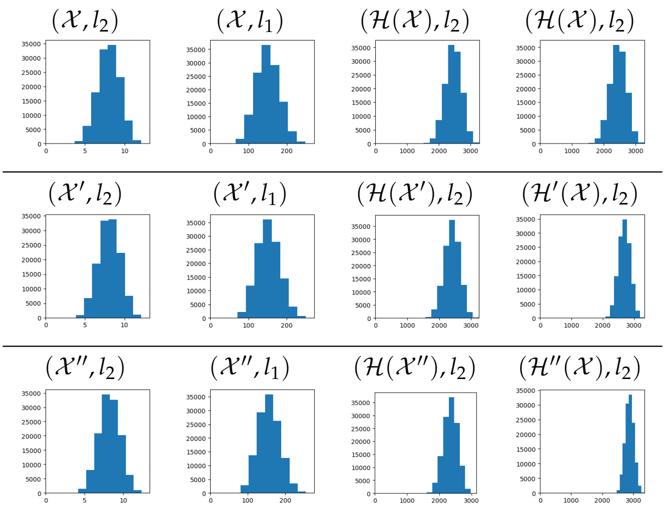


Figure C.39.: Distribution of distances for ISOLET.

C.1.4 *Norm distribution*

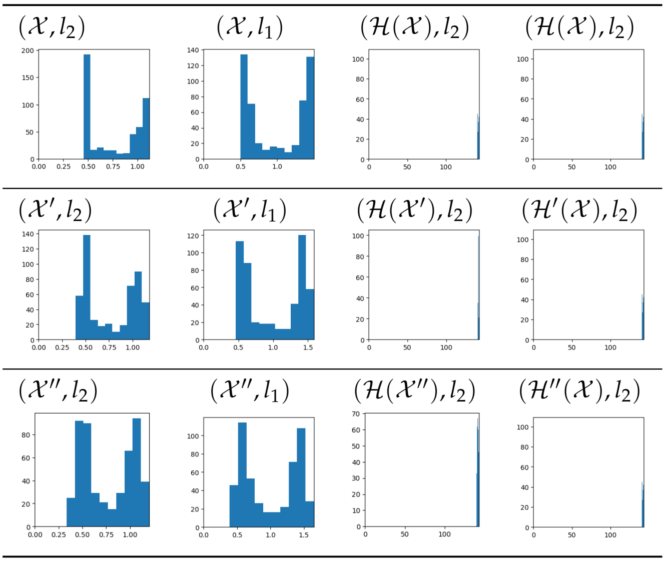


Figure C.40.: Norm distribution for the astroid.

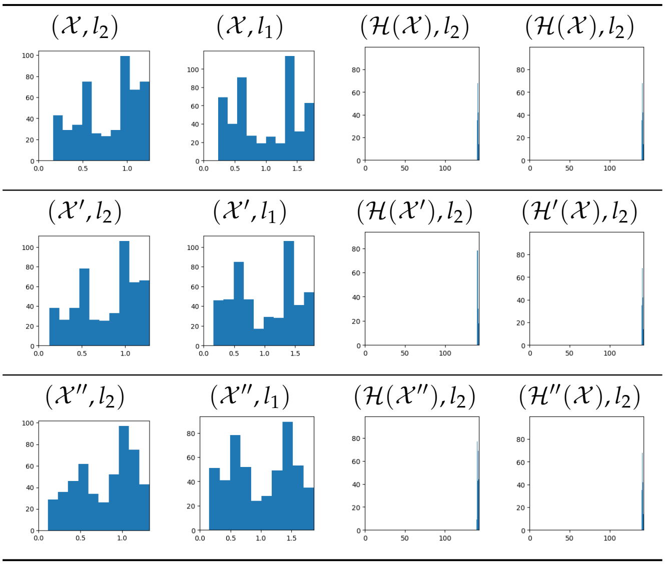


Figure C.41.: Norm distribution for the apple.

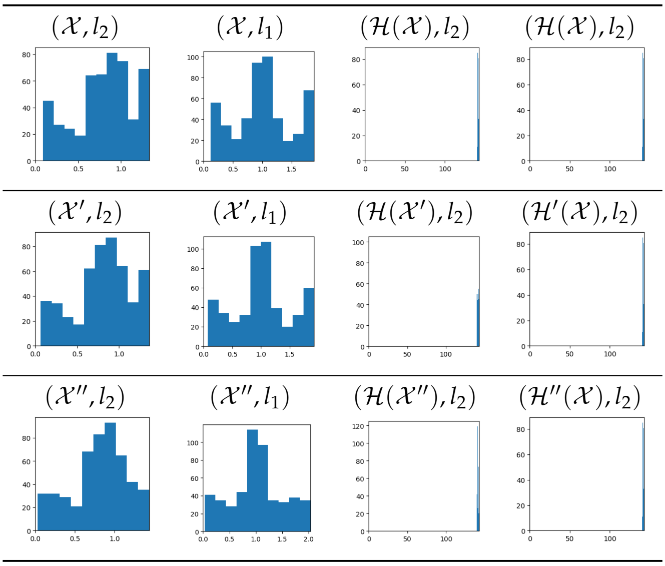


Figure C.42.: Norm distribution for the lemniscate.

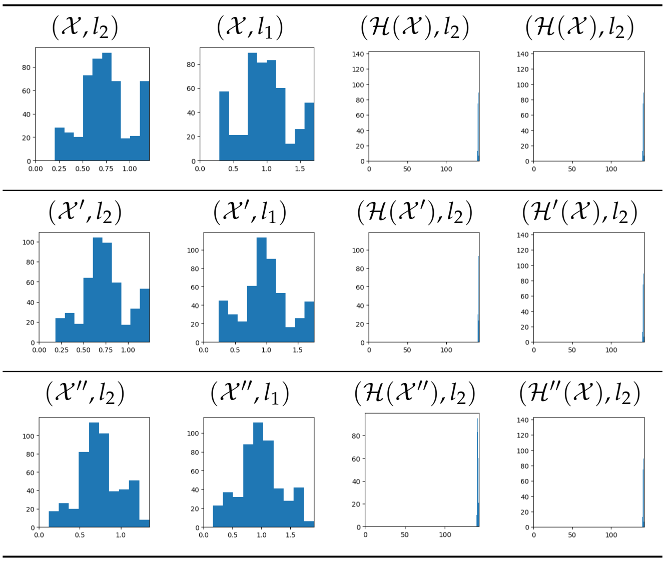


Figure C.43.: Norm distribution for the concentric circles.

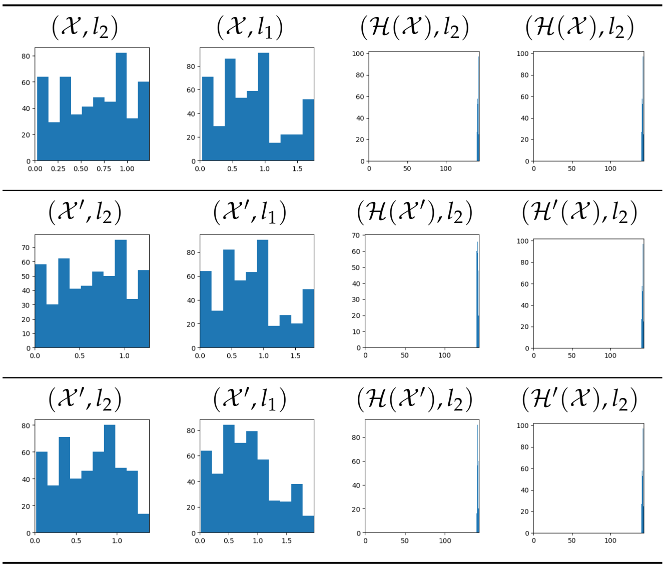


Figure C.44.: Norm distribution for the fish.

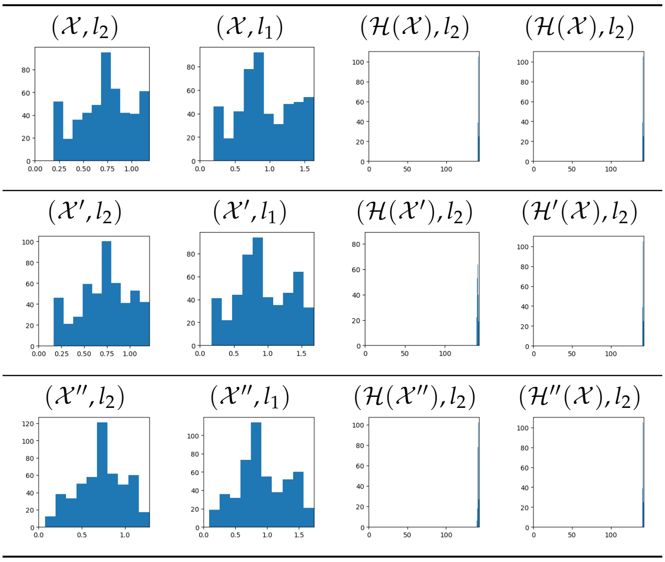


Figure C.45.: Norm distribution for the star.

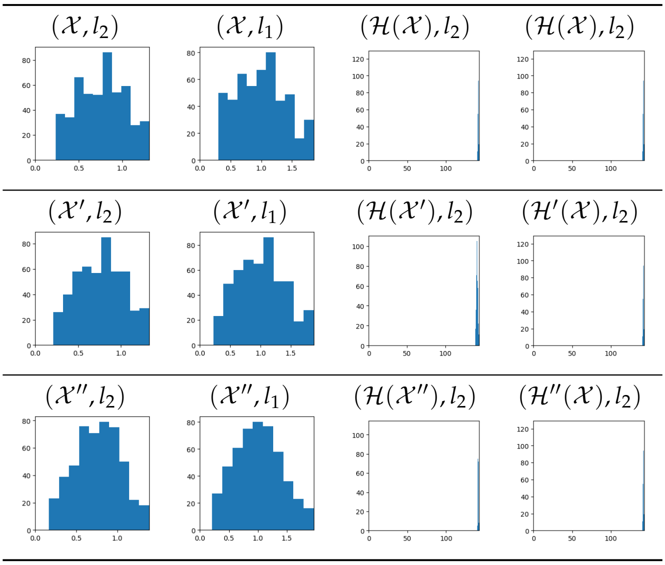


Figure C.46.: Norm distribution for the olympics.

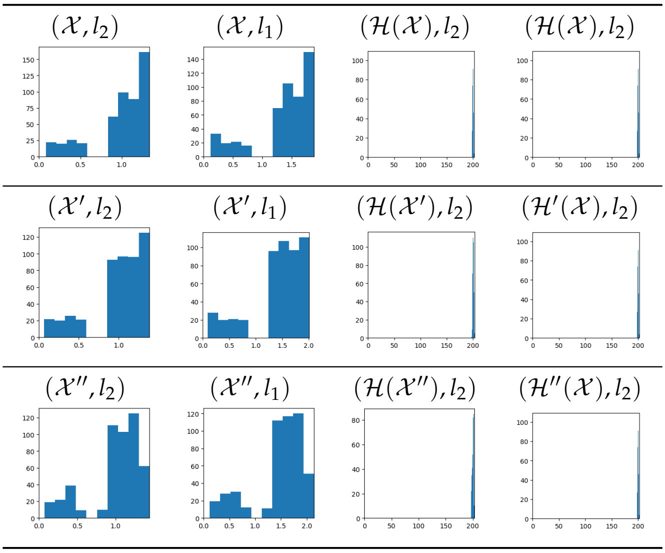


Figure C.47.: Norm distribution for the orthogonal circles.

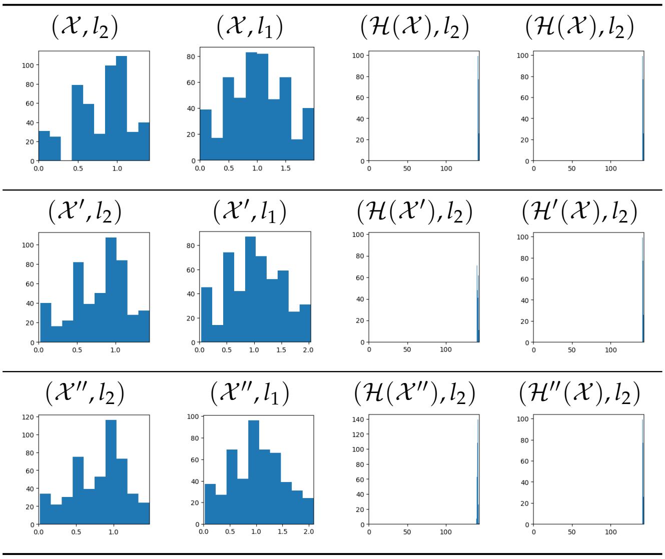


Figure C.48.: Norm distribution for the nine squares.

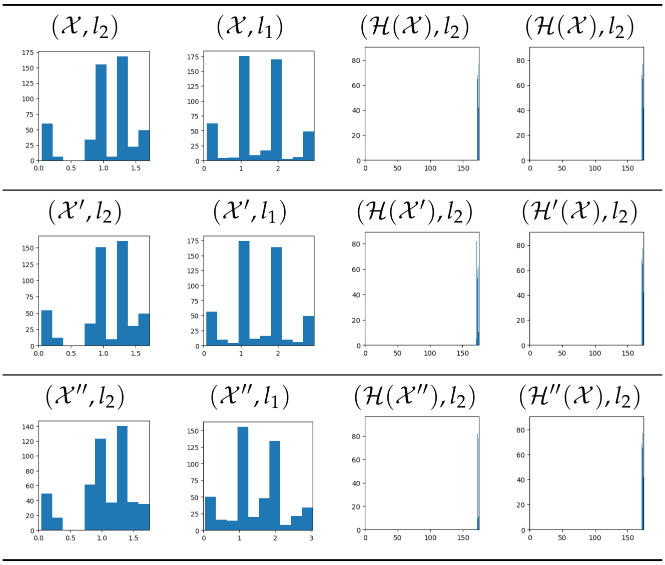


Figure C.49.: Norm distribution for the cube vertices.

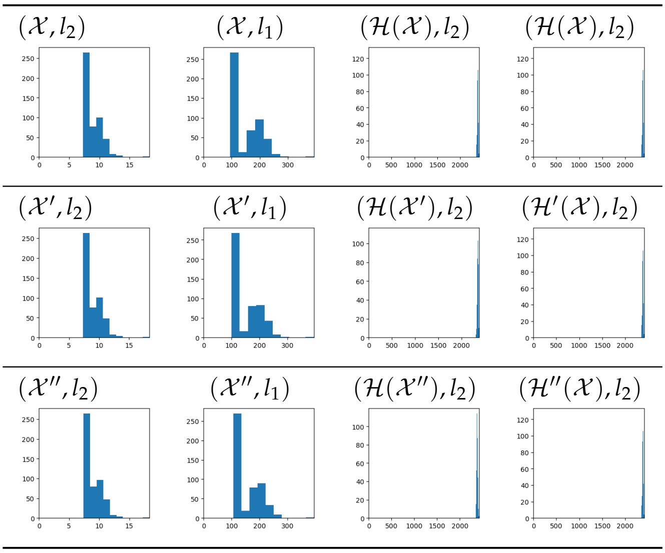


Figure C.50.: Norm distribution for UCI-HAR.

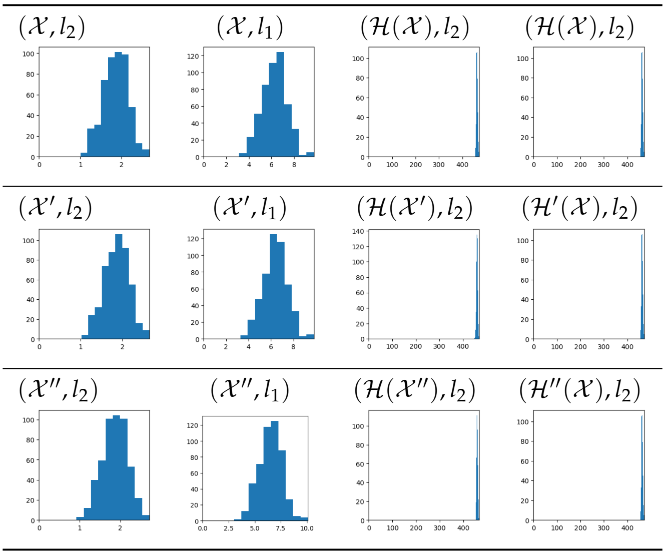


Figure C.51.: Norm distribution for CTG.

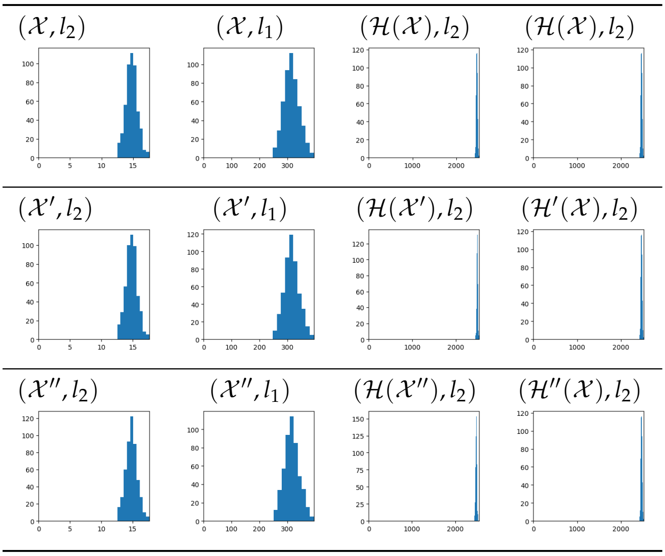


Figure C.52.: Norm distribution for ISOLET.

C.2 Structure preservation, or preservation of persistent homology

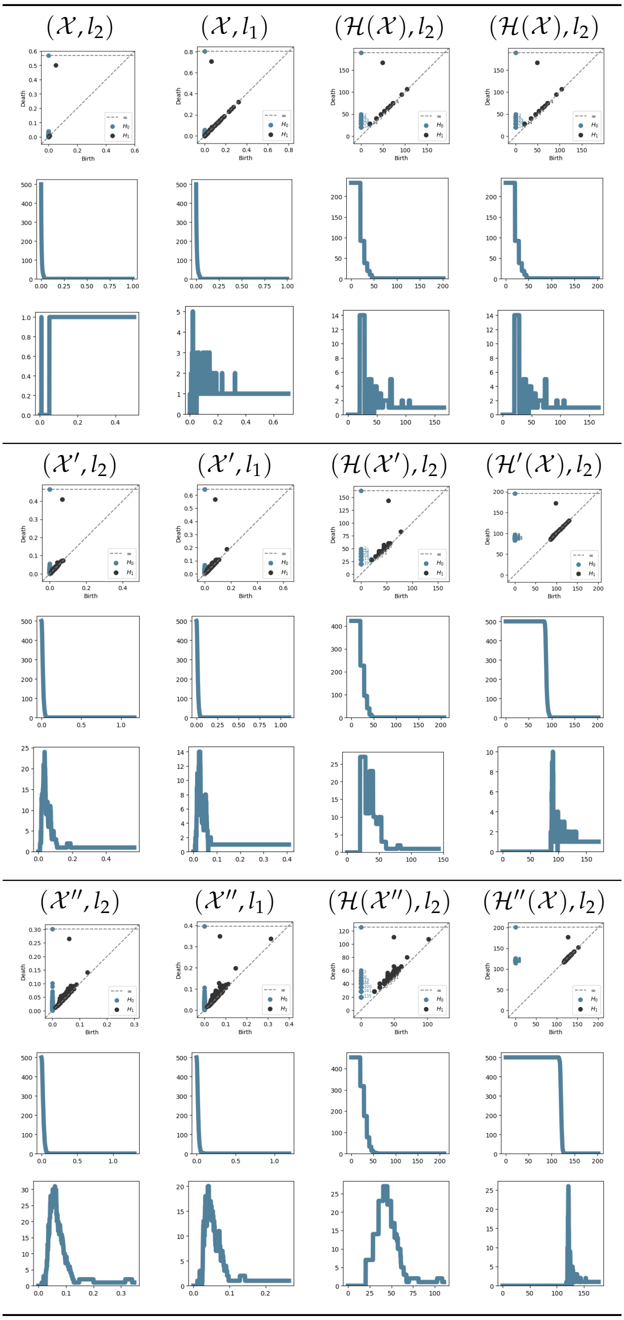


Figure C.53.: 0- and 1-dimensional persistent homology for the astroid.

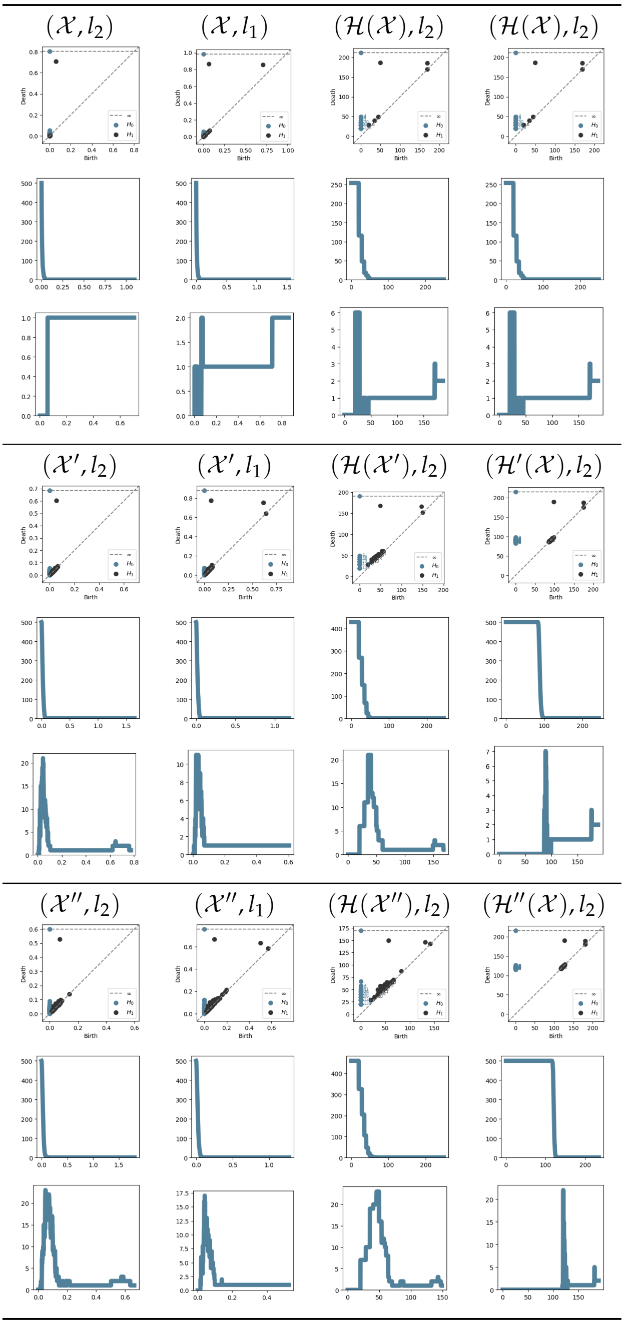
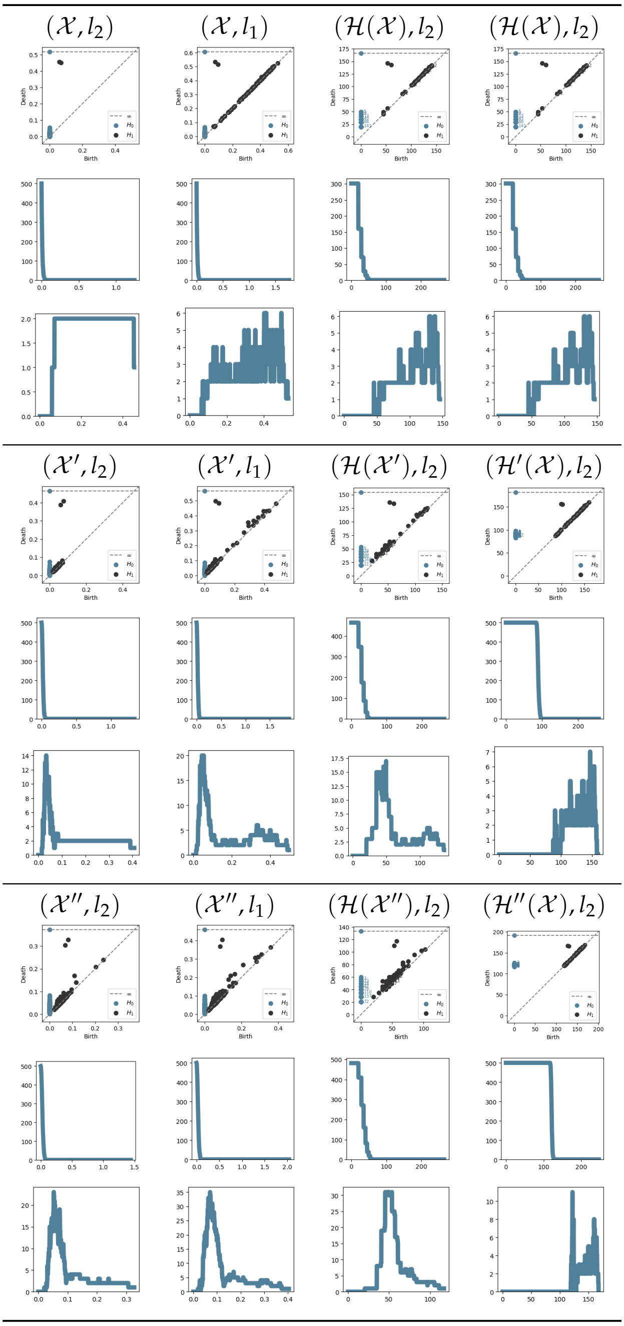


Figure C.54.: 0- and 1-dimensional persistent homology for the apple.

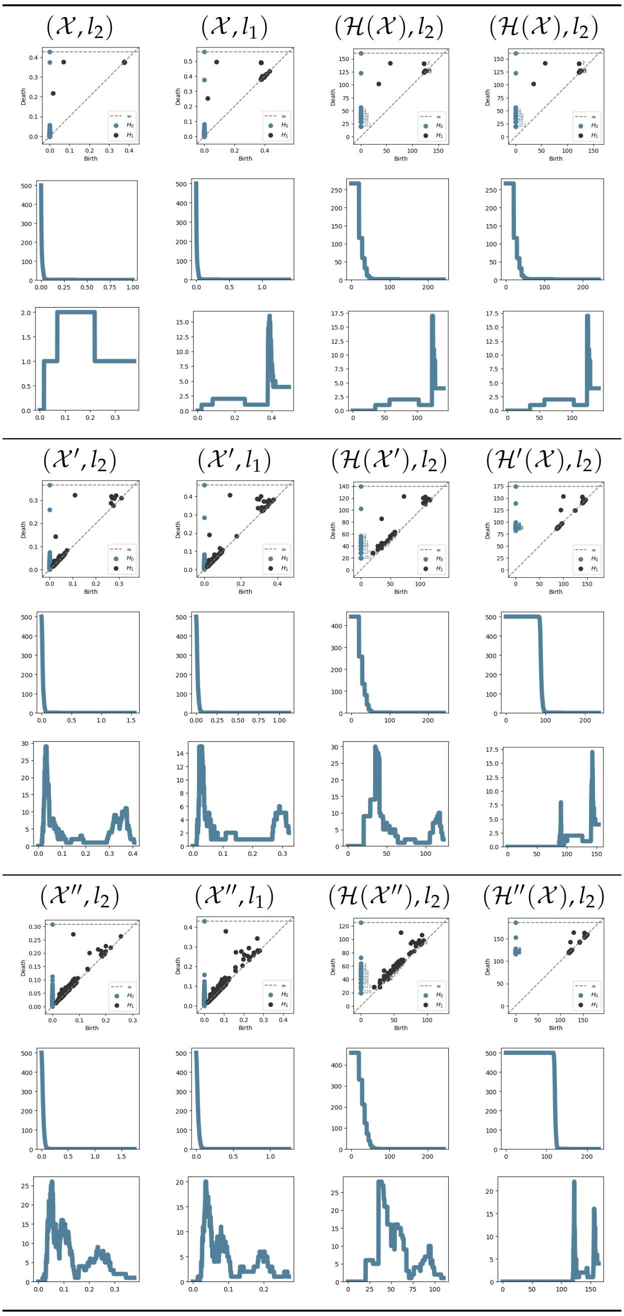


$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.55.: 0- and 1-dimensional persistent homology for the lemniscate.



$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.56.: 0- and 1-dimensional persistent homology for the concentric circles.

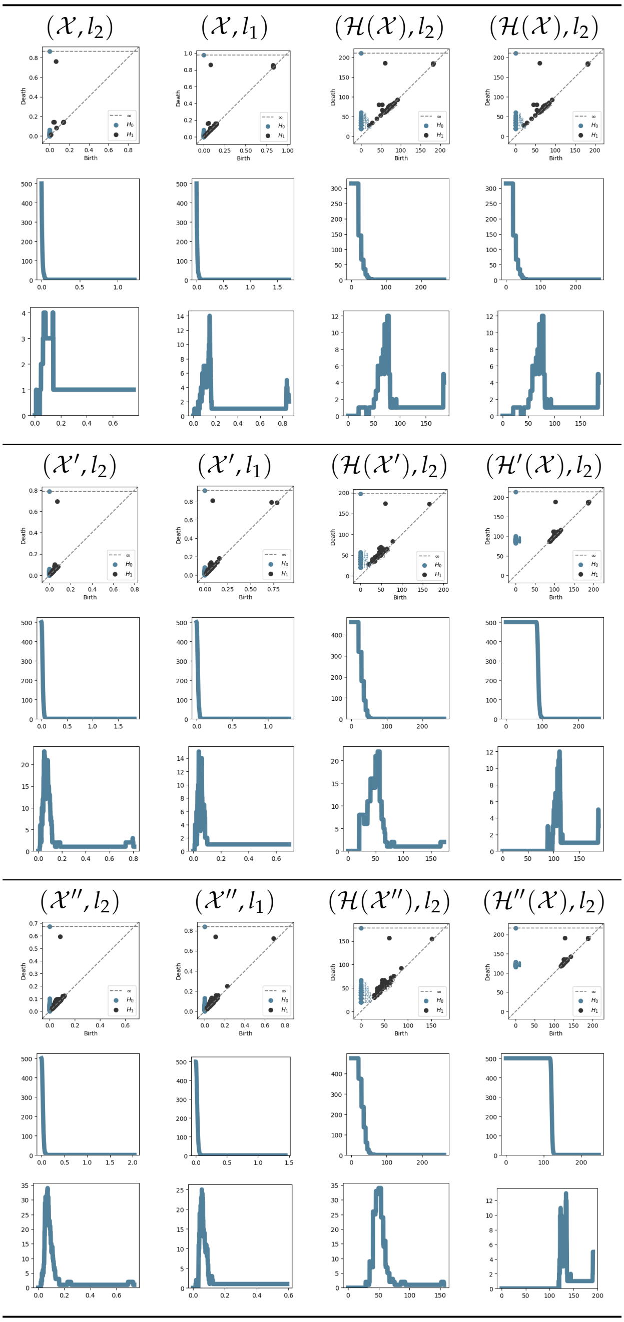


Figure C.57.: 0- and 1-dimensional persistent homology for the fish.

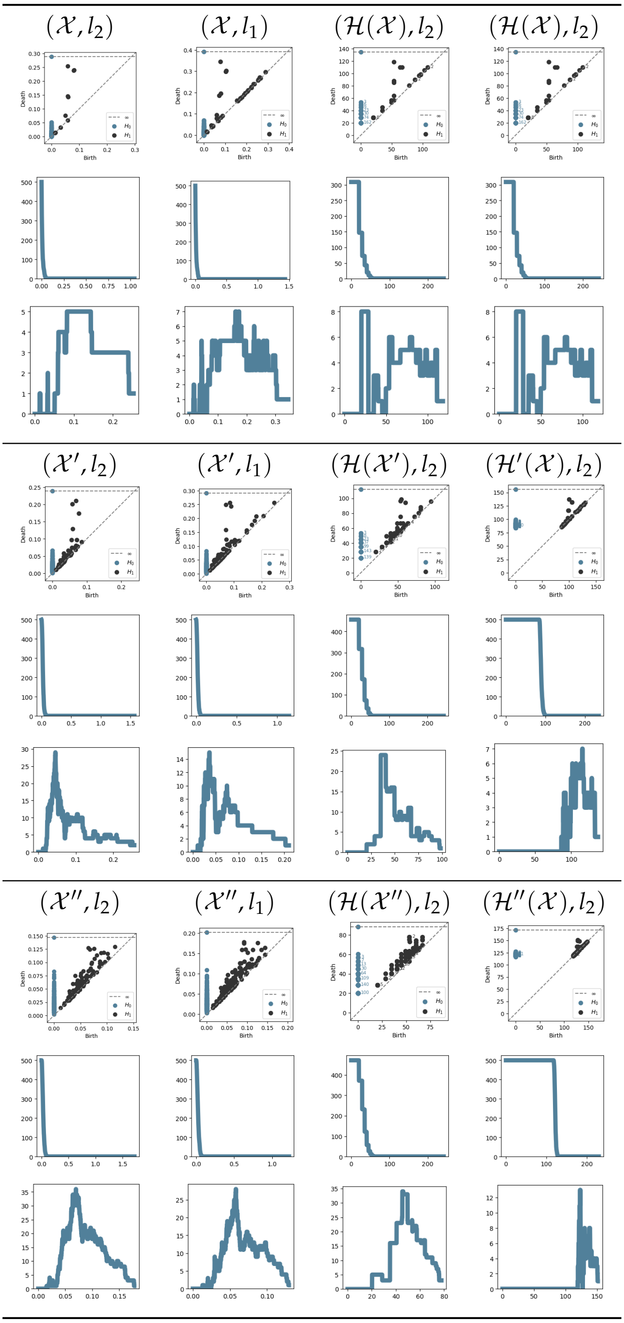
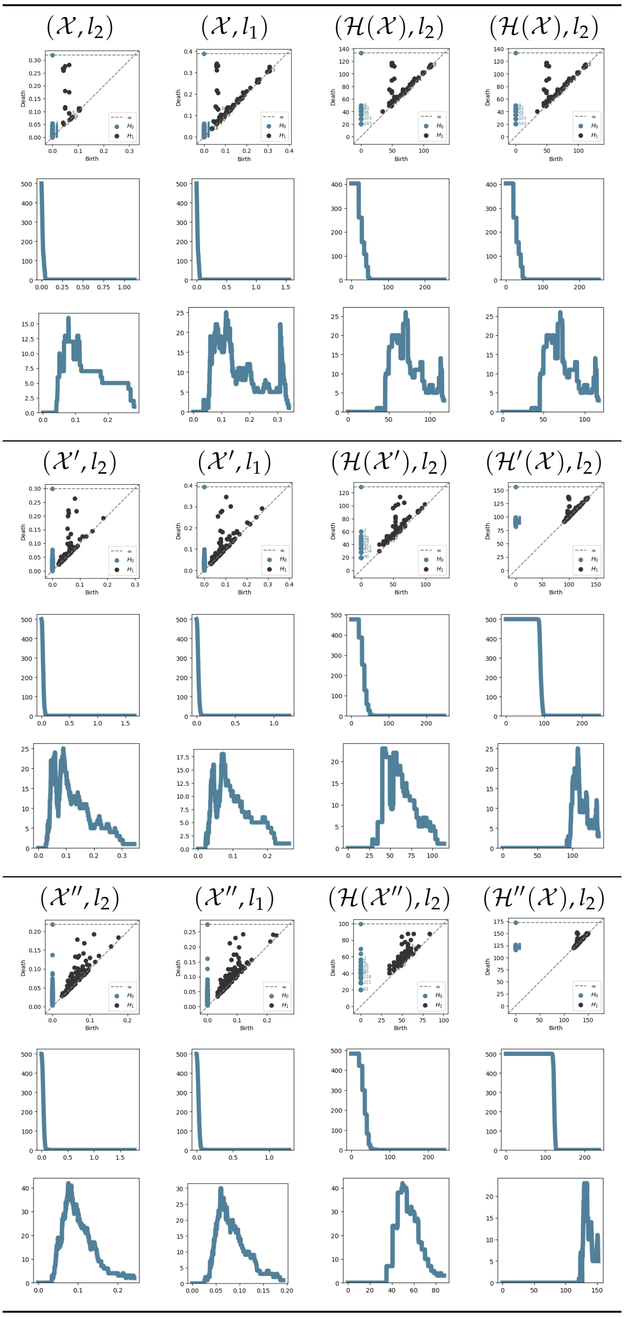


Figure C.58.: 0- and 1-dimensional persistent homology for the star.


$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.59.: 0- and 1-dimensional persistent homology for the olympics.

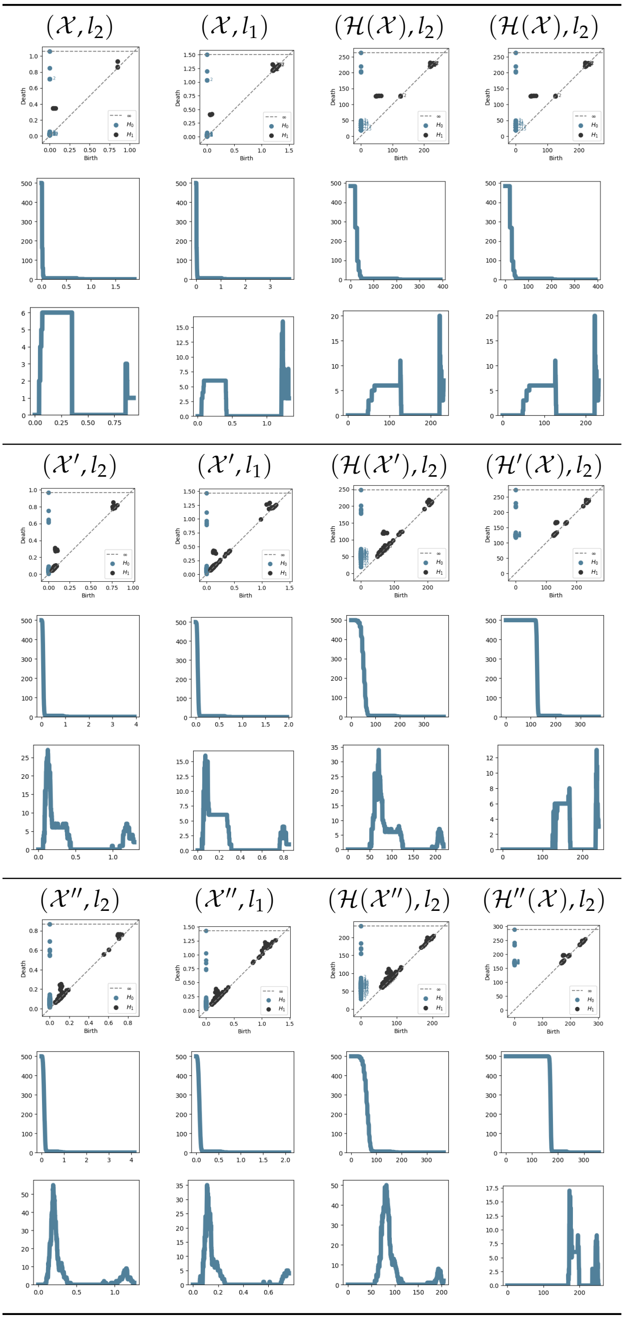
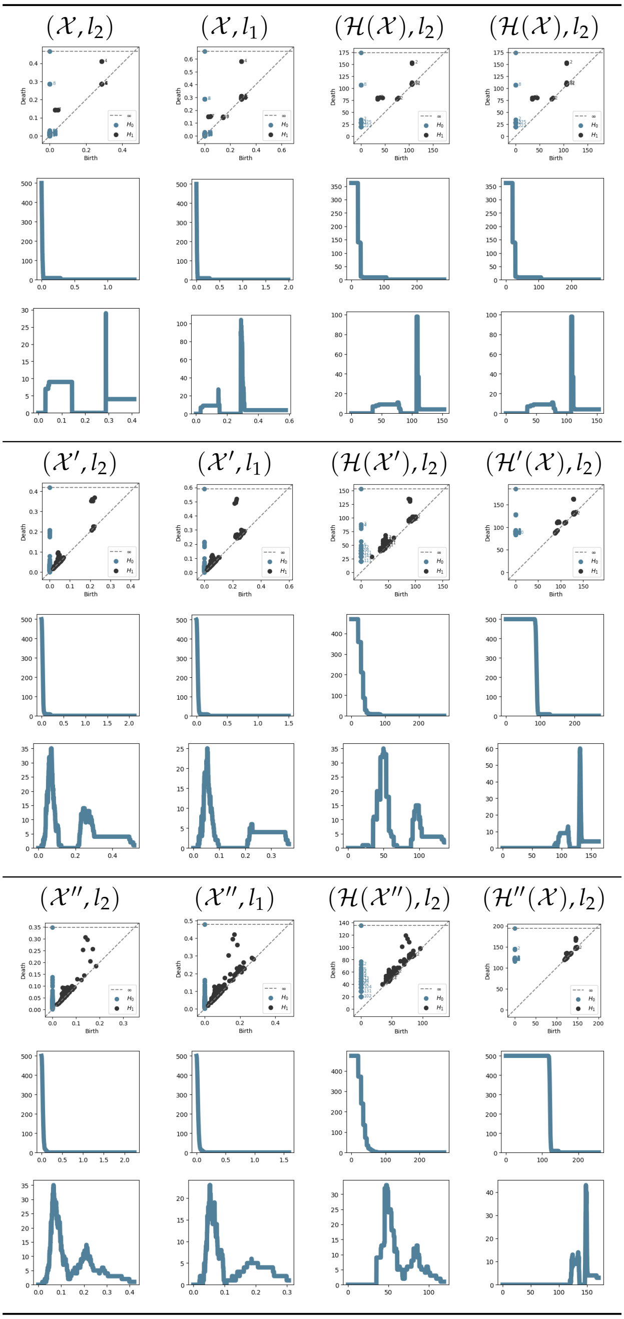


Figure C.60.: 0- and 1-dimensional persistent homology for the orthogonal circles.

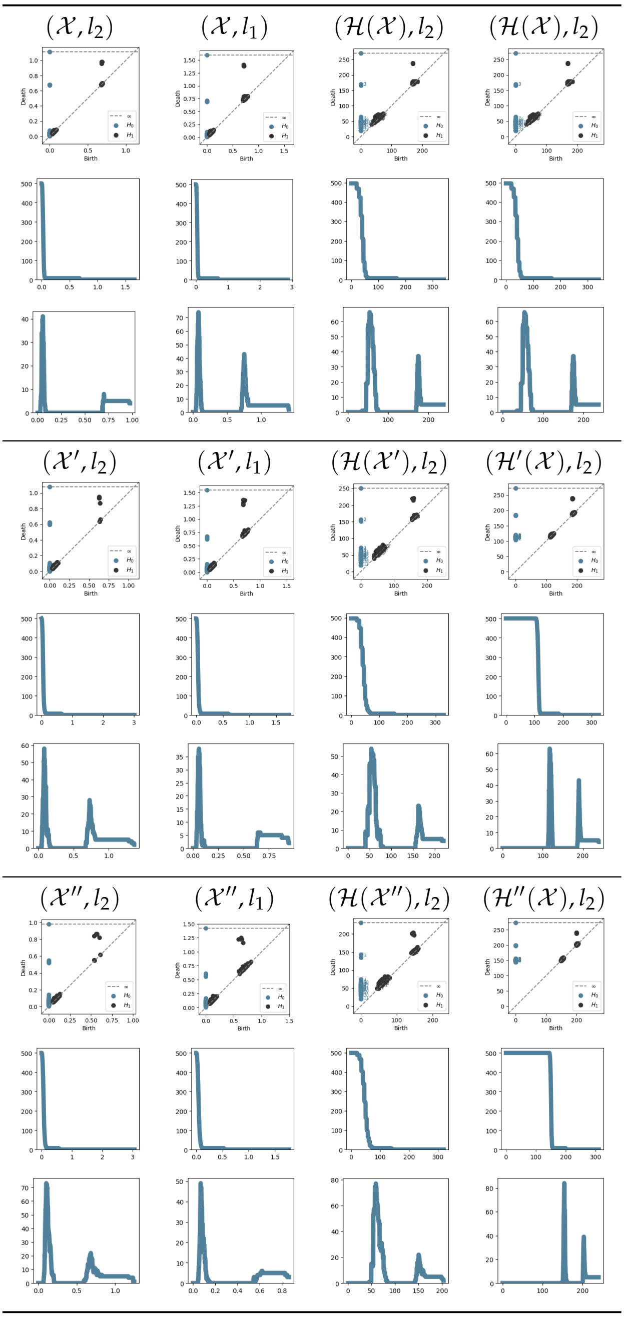


$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.61.: 0- and 1-dimensional persistent homology for the nine squares.



$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.62.: 0- and 1-dimensional persistent homology for the cube vertices.

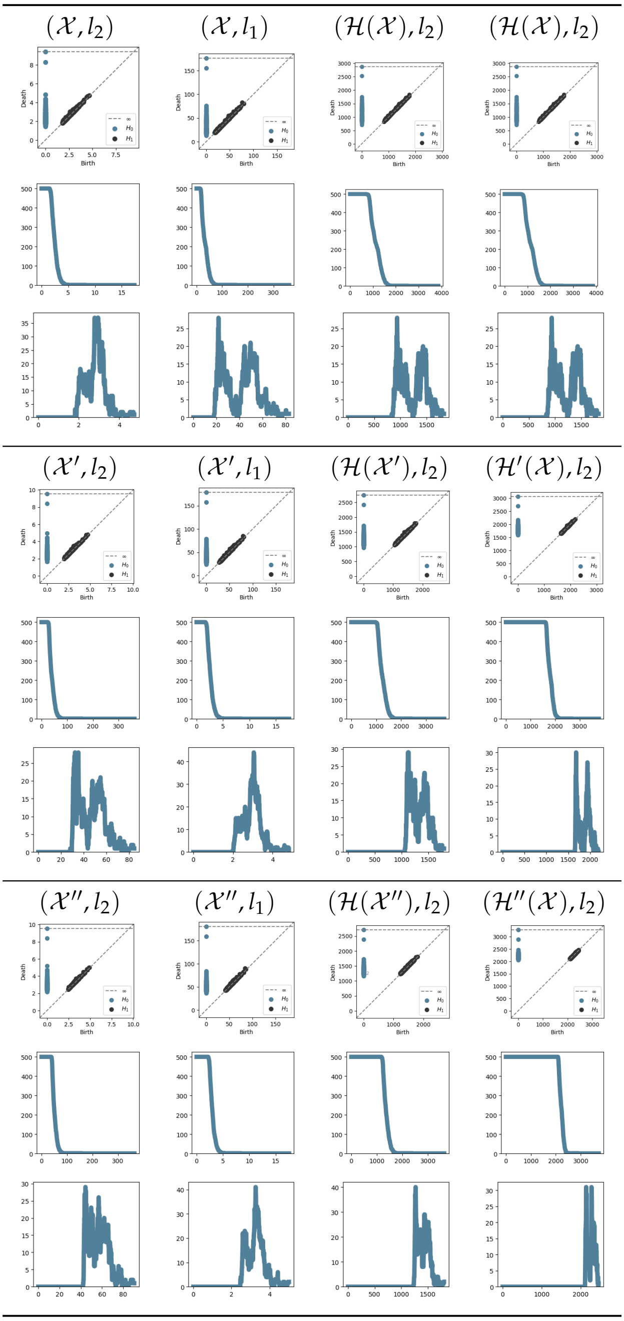
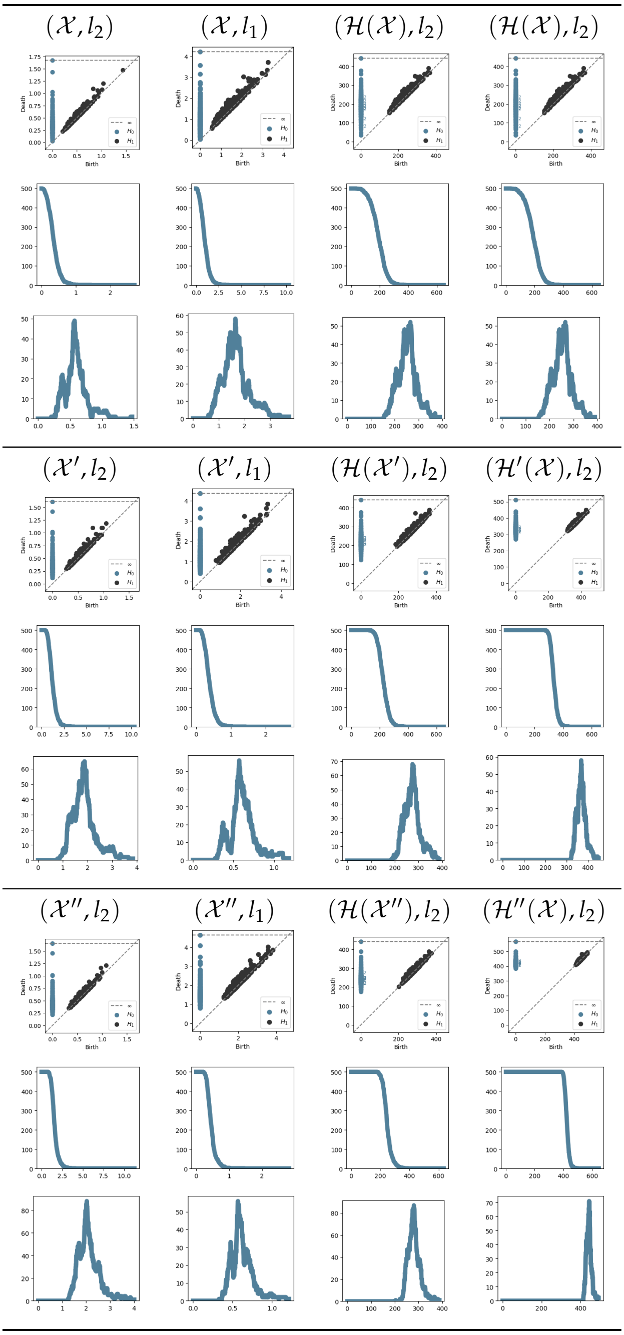


Figure C.63.: 0- and 1-dimensional persistent homology for UCI-HAR.



$(\mathcal{X}, l_2)$ $(\mathcal{X}, l_1)$ $(\mathcal{H}(\mathcal{X}), l_2)$ $(\mathcal{H}(\mathcal{X}), l_2)$

$(\mathcal{X}', l_2)$ $(\mathcal{X}', l_1)$ $(\mathcal{H}(\mathcal{X}'), l_2)$ $(\mathcal{H}'(\mathcal{X}), l_2)$

$(\mathcal{X}'', l_2)$ $(\mathcal{X}'', l_1)$ $(\mathcal{H}(\mathcal{X}''), l_2)$ $(\mathcal{H}''(\mathcal{X}), l_2)$

Figure C.64.: 0- and 1-dimensional persistent homology for CTG.

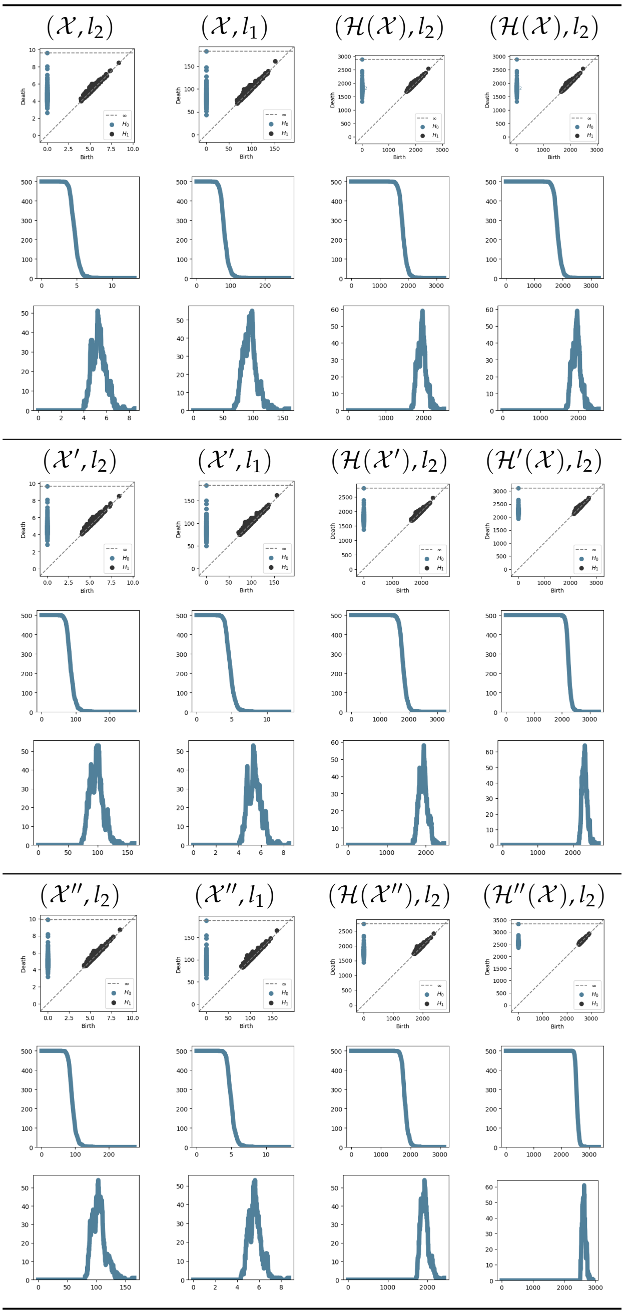


Figure C.65.: 0- and 1-dimensional persistent homology for ISOLET.

D

Classification of targets and distractors in an audiovisual attention task based on EEG

This appendix can be seen as a predecessor to Chapter 5, as it involves preliminary investigation of the same WithMe project that aims to improve interaction between humans and artificial intelligence. In the WithMe experiment, participants are shown sequences of Target and Distractor digits, and are instructed to recall the Targets. The goal here is to validate the experimental paradigm and data set: we show that the benchmark EEG models perform well in classifying between EEG signals for the Target and Distractor stimuli, and therefore, the WithMe paradigm is appropriate for eliciting attention and related P300 event related potentials (ERPs).

The appendix is based on the following publication:

> Steven Mortier, Renata Turkeš, Jorg De Winne, Wannes Van Ransbeeck, Dick Botteldooren, Paul Devos, Steven Latré, Marc Leman, and Tim Verdonck, *Classification of targets and distractors in an audiovisual attention task based on EEG*, Sensors (2023).

This research was funded by the Research Foundation-Flanders (FWO) under Grant No. G0A0220N.

D.1 Introduction

The interaction between humans and artificial intelligence (AI) still lacks the level of engagement and synchronization that symbolizes the interaction between humans. The primary goal of the WithMe project[1] is to thoroughly study the processes that occur in the human brain during joint activities with another individual, such as working towards shared objectives [93]. The brain signals collected in this study are primarily indicative of attention, but also of emotion and reward. The purpose of this research is to determine relevant EEG features indicative of attention using machine learning (ML).

To this end, a specific experiment was designed. Temporal audiovisual integration and support of visual attention by sound is well demonstrated in the pip-and-pop experiment [48]. The pip-and-pop experiment is based on a visual search, which does not lead to a strong visually evoked potential. Moreover, as we expected that rhythmic presentation of target stimuli also affects working memory, the task was replaced by a modified digit-span task where five target digits had to be remembered and reported in our experiment [93]. This task involves visual attention, working memory, and sequence recall. To investigate the role of attention, we measured the brain activation directly by means of EEG.

Specifically, event-related potentials (ERPs) have been shown to be excellent tools for studying attention [220, 361]. Risto Näätänen was a pioneer in this domain, as he studied the connection between ERPs and attention, which led to his discovery of the (auditory) mismatch negativity ERP [107, 247–249]. Additionally, research has shown that the amplitude of the P300 is directly related to the amount of attentional resources available for stimulus processing [107, 144, 170, 294]. The P300 ERP is observed to be elicited for deviant stimuli in a sequence of standard stimuli, where the deviant stimuli are in some way more relevant to the presented task [275, 309]. In our experiment, we thus expect that the targets elicit a P300 ERP. Research has shown that the P300 actually consists of two subcomponents: the P3a and P3b [277]. The P3a generally reaches its peak around 250ms to 280ms post-stimulus and is associated with attention-related brain activity [276]. On the other hand, the P3b peak can vary in latency, lying between 300ms and 500ms post-stimulus [277]. The P3b is elicited by improbable events, provided that the improbable event is somehow relevant to the task at hand [104]. In our experimental setting, we expect to elicit a P3a, as the target stimuli are not scarce (there are approximately 50% targets and 50% distractors) and our experiment is designed to evoke attention. We do not expect to elicit a P3a for distractors, as subjects should not pay attention to them.

The goal of this work is to accurately classify whether a target or distractor stimulus was presented to the subject, based on the subject's EEG data. For this purpose, we

[1] The WithMe project is a research project funded by the Research Foundation Flanders (FWO). More information can be found at `https://researchportal.be/en/project/withme-making-human-artificial-intelligence-interactions-more-entraining-and-engaging`.

applied different existing ML methods to classify EEG data and investigate which method performs best on our specific use case. As we expect to elicit attention when a target is shown (and not when a distractor is shown), the trained ML will effectively be an attention detector. We expect the attention to manifest itself in the form of a P3a ERP, and therefore expect that the model will base its predictions on the presence of a P3a peak. Detecting P3a signals, and, more broadly, P300 signals, has a wide range of applications [224, 253], particularly in P300-based brain computer interfaces (BCIs) [223], for example in spellers [150, 181, 256] and intelligent home control systems [16, 225]. These applications can be of great help for patients suffering from amyotrophic lateral sclerosis (ALS) or spinocerebellar ataxia, as it can enable them to communicate in a daily environment [147, 150, 160, 256]. In the literature, a wide array of techniques are used to classify and detect P300 [5]. Some techniques rely on a data transformation and subsequently use logistic regression to classify the transformed data, for example xDAWN+RG [21, 22, 85, 298]. Recently, deep learning approaches, primarily based on convolutional neural networks (CNNs), for example EEGNet [198, 271, 374], have also gained in popularity [192, 229, 313]. Finally, as EEG data are essentially heavily correlated multivariate time series, it is possible to apply standard time series classification techniques as well [95, 96, 108].

Building BCIs that are trained on multiple subjects and generalize well to previously unseen subjects holds significant value [139]. Indeed, BCIs often need to be re-trained or at least calibrated for the end-user [362], which is a costly and user unfriendly process [221, 363]. However, due to inter-subject variability of EEG data, training models that generalize to multiple subjects (cross-subject (CS) models) is a harder task than training models for one subject (individual subject (IS) models) [221, 242, 363]. For this reason, we also investigate the hypothesized drop in performance when transitioning from IS to CS models. Additionally, the ML models should be able to make predictions in real-time, as this is essential in real-world BCI applications.

Finally, we analyze which EEG channels and time points are used by our models to make its predictions, and check whether these align with the expected P3a attention signature. However, ML models such as CNNs are considered “black boxes”, as no clear explanation for the decisions made by these models exists [212]. The rapidly emerging and improving field of explainable AI (xAI) aims to tackle these issues by providing insights into ML models’ decision-making processes. Some xAI techniques that are often used to gain insights into EEG classification models are Local Interpretable Model-Agnostic Explanations (LIME) [162, 293], DeepLIFT [127, 198, 314] and saliency maps [15, 121, 317], among others.

In summary, we aim to enhance the interaction between humans and AI and have designed a novel experiment for this purpose. Specifically, this work considers building a ML model to recognize targets shown to a subject, which equates to creating an attention detector. These models should ideally generalize well to previously unseen subjects. The primary contributions of this work are:

- Training of state-of-the-art classification methods to accurately predict target and distractor stimuli based on EEG data.
- Analysis of the performance difference between IS and CS models.
- Investigation into which EEG channels and time points were important for the model predictions, using xAI.

Ultimately, the contributions of this research collectively advance our understanding of human-AI interaction and will aid in the development of more effective BCIs and their associated applications.

The remainder of this paper is structured as follows: Section D.2 illustrates the classification problems and provides a description of the classification methods used in this work. Section Section D.3 presents the results and provides an in-depth analysis of the best performing model. This section also contains an extensive discussion of the achieved results. Finally, in Section Section D.4, we draw conclusions and provide possible directions for future research.

D.2 Classification problem

The WithMe EEG data of multivariate time series is described in detail in Section 5.2.1. As mentioned in Section D.1, we expect to observe a P3a ERP when subjects see a target stimulus. The P3a ERP is characterized by a positive voltage deflection between 250ms and 280ms post-stimulus, although the exact timing can vary [94, 263, 276]. As our experiment uses visual stimuli, we expect the P3a ERP to be the most pronounced in the parietal-occipital region of the brain [80]. Figure D.1 shows the evoked response for one subject, averaged over all parietal-occipital electrodes, as indicated in the figure inset. We observe a clear positive deflection between 200ms and 300ms post-stimulus, which is in line with our expectations.

The models trained in this study consider a two-class classification problem (target versus distractor) and take single-trial EEG epochs as input to predict a binary label. As the data is downsampled to 50 Hz, one epoch contains 60 time steps, for 64 electrodes. This means that the input is of shape $(N, 64, 60)$ with N the number of epochs. It is important to note that it is impossible to obtain 100% accuracy for this model. Indeed, the model makes a prediction based on the subject's assessment of a stimulus, and it is possible that a subject did not correctly recognize all targets and distractors. As the ground truth labels are based on the predefined labels of the experiment, it is possible that there is a slight mismatch between the labels and the subject's perceived class. Nevertheless, we assume that this problem is rare, meaning that commonly used metrics, for example accuracy, have a valid interpretation.

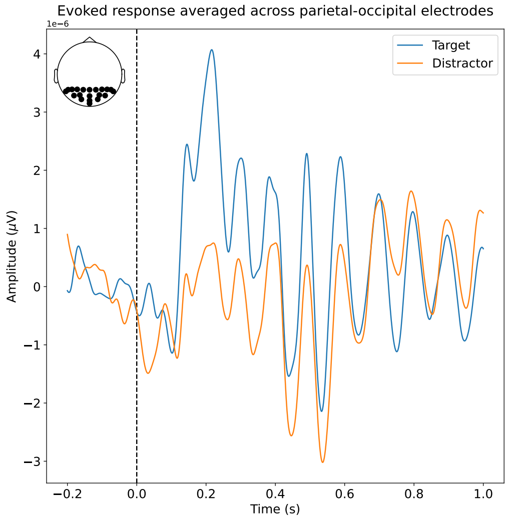


Figure D.1.: The evoked response for targets and distractors for one subject. The data is averaged over all electrodes of the parietal-occipital region in the brain, as indicated in the figure inset.

Ideally, the models should be able to generalize to previously unseen subjects. To investigate this, we train the models in two ways: models trained on IS and models trained on (nearly) all subjects, also called CS models. The former will be evaluated using a randomly sampled test set with a standard 80% train and 20% test set split, while the latter are evaluated using a leave-one-out (LOO) methodology. In general, models perform better when trained and tested on individual subjects [132]. This can be attributed to the variability in subject's EEG data elicited by the same stimuli. However, in practice, EEG classification models should ideally extrapolate to previously unseen subjects. For example, BCIs often need to be calibrated for new end-users, which usually takes 20 to 30 minutes [123, 134, 194]. Therefore, it is interesting to investigate which model architectures are best suited to build subject-independent classifiers.

D.2.1 *Classifiers*

To solve this classification problem, we train and evaluate different existing ML models. Different methodologies for classifying EEG data exist. For example, we can extract features from the EEG data and use these extracted features as input to the classifier. These features can, among others, be extracted from the time domain, frequency domain or the time-frequency domain, or through methods such as principal component analysis [279, 359]. Such methods are referred to as feature-based methods. Another common approach uses raw or preprocessed EEG data as input to the classifier. In this approach, commonly referred to as an end-to-end method, the classifier itself will extract relevant features from the data during training, and use these features to classify a sample. As both methodologies are interesting approaches, we will use methods belonging to both approaches. In this study, we apply four distinct classifiers and compare the results on a novel data set. An overview of the classifiers and their methodologies is presented in Table D.1. First, we will apply the xDAWN pipeline, which has demonstrated significant success in several EEG classification tasks [85, 298]. For example, the BCI challenge organized as part of the IEEE Neural Engineering Conference 2015 was won by an xDAWN-based approach [85]. In this study, we employ a similar approach, consisting of first estimating two sets of xDAWN spatial filters, one for each class (target and distractor) [298]. Subsequently, the grand average evoked potential of each class is filtered using the corresponding filters, after which they are concatenated to each of the trials. Then, the covariance matrix of each resulting trial is used as a feature for the next steps in the pipeline [24, 83]. The next step is to project the covariance matrices on the tangent space using a Riemannian metric, as described in [21, 22]. After these feature extraction steps, a classifier is used to make the final predictions. Based on [85] and [25], we used logistic regression [268]. For the remainder of this study, we will refer to this method as xDAWN+RG (xDAWN+Riemannian Geometry). Calculating the xDAWN covariance matrices and projection to the tangent space were done using the PyRiemann package [25].

The second method we consider is EEGNet [198]. EEGNet exhibits strong performance on a variety of EEG-based classification tasks, such as P300 ERP classification [121, 198] and motor imagery classification [372]. Whereas the previous method used extracted features as input to the classifier, EEGNet performs both the feature extraction and classification. EEGNet is a deep learning model, more specifically a CNN. As its name suggests, EEGNet is optimized for classifying EEG data by employing a set of specific design choices. Firstly, it uses temporal convolutions to learn frequency filters [198]. As suggested by the authors, the length of the temporal kernel used in these convolutions is set to half the sampling rate, which allows the model to capture frequency information at frequencies of 2 Hz and higher [198]. Second, depthwise convolutions are used to learn frequency-specific spatial filters. In this context, depthwise convolutions have two main advantages. First, they noticeably reduce the number of trainable parameters, since these convolutions are not fully connected to the previous layer; instead, they are connected to each feature map individually. This induces the second, EEG-specific advantage: the model learns spatial filters for each temporal filter, which enables the efficient extraction of frequency-specific spatial filters [198]. The last convolutional part consists of a separable convolution, which is a combination of a depthwise convolution and pointwise convolution. The former learns how to summarize individual feature maps in time, while the latter learns how to optimally combine the feature maps [198]. Finally, all features are passed to a dense layer for classification. More details on the EEGNet architecture can be found in [198]. We use the standard EEGNet-8,2 layout, which means that the model learns 8 temporal filters and 2 spatial filters per temporal filter.

The first two methods are designed for EEG specifically. However, since EEG data is essentially a heavily correlated multivariate time series, it is interesting to study the results of a more general method designed to classify such time series. To this end, we applied Rocket (RandOm Convolutional KErnel Transform) [95]. Based on the success of CNNs for time series classification, Rocket uses random convolutional kernels combined with simple linear classifiers. This novel combination achieves state-of-the-art performance on the UCR time series archive using only a fraction of the computational cost of existing methods [91, 95]. As a follow-up to Rocket, the authors also designed MiniRocket [96]. They claim MiniRocket can be trained up to 75 times faster than Rocket, while achieving nearly the same performance. MiniRocket distinguishes itself from Rocket primarily by reducing the degree of randomness that Rocket generates, resulting in MiniRocket being almost deterministic [96]. Since methods to classify EEG data, such as EEGNet, can be very computationally expensive, it is worth exploring the effectiveness of less computationally expensive methods. We used the Rocket and MiniRocket implementations in the sktime package and combined them with the ridge regression classifier implemented in scikit-learn, as suggested by the authors [95, 96, 216, 268].

D.2.2 *Metrics*

To allow the comparison of various approaches, it is essential to have predetermined performance metrics. We will focus on three metrics that are widely used in EEG classification literature: accuracy, F1-score, and area under the receiver operating characteristic curve (ROC AUC) [303]. First, the accuracy states the number of correctly classified samples across both classes. Second, the F1-score assesses the predictive performance of a model by calculating the harmonic mean of the precision and recall metrics. The equations used to calculate the accuracy and F1-score are given in D.1 and D.4 respectively, where we use the following abbreviations: true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Third, by plotting the true positive rate against the false positive rate for different classification thresholds, we obtain the ROC curve. The ROC AUC is defined as the area under this curve and provides a measure for how well a classifier can distinguish between true and false samples, or in our case, targets and distractors respectively. Finally, we also assess the required training time and model complexity of all models.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{D.1}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{D.2}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{D.3}$$

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \tag{D.4}$$

D.3 Experimental results

D.3.1 *Individual subject models*

The performance of the models, assessed using the metrics introduced in Section D.2.2, can be seen in Table D.2 and Figure D.2. Evidently, using an EEG-specific model architecture benefits the performance of IS models. While xDAWN+RG and EEGNet perform equally well, they demonstrate superior accuracy, F1-score and area under the curve (AUC) in comparison to MiniRocket and Rocket. As expected, MiniRocket achieves slightly inferior performance compared to Rocket. However, MiniRocket's training time was 15 times faster on our dataset. Notably, while xDAWN+RG and EEGNet exhibit equal performance, xDAWN+RG is significantly less computationally expensive than EEGNet. On central processing units (CPUs) alone, EEGNet's training time is 9 times longer. Although training times can be accelerated for EEGNet using (expensive) graphics processing units (GPUs), even when using an NVIDIA GTX 1080 GPU, EEGNet still requires 2.5 times as long to train as xDAWN+RG.

D.3.2 *Cross-subject models*

Similar results are obtained for CS models, where EEG-specific approaches perform better than Rocket and MiniRocket, as shown in Table D.3 and Figure D.3. However, in this scenario, EEGNet outperforms xDAWN+RG. We hypothesize that this can be attributed to EEGNet's added complexity and a greater number of parameters compared to xDAWN+RG. This additional capacity is more likely to be able to learn features that extrapolate well to previously unseen data points.

D.3.3 *Individual subject models vs. cross-subject models*

As we discussed in D.2, we expect that the performance of the IS models is better than that of the CS models. Despite having access to a significantly larger amount of data, constructing a CS model is a considerably more challenging task. To

Model	Target domain	Methodology
xDAWN+RG [298]	EEG	feature-based
MiniRocket [96]	time series	feature-based
Rocket [95]	time series	feature-based
EEGNet [198]	EEG	end-to-end

Table D.1.: Overview of the methods that were used in this study, together with their original target domain and methodology.

Model	Accuracy	F1-score	AUC
xDAWN+RG	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.84 ± 0.05**
MiniRocket	0.69 ± 0.05	0.70 ± 0.05	0.76 ± 0.06
Rocket	0.72 ± 0.05	0.72 ± 0.05	0.79 ± 0.06
EEGNet	**0.76 ± 0.04**	**0.76 ± 0.04**	**0.83 ± 0.04**

Table D.2.: Classifier test performance for individual subject models, averaged across the 42 subjects. The best performances are indicated in boldface.

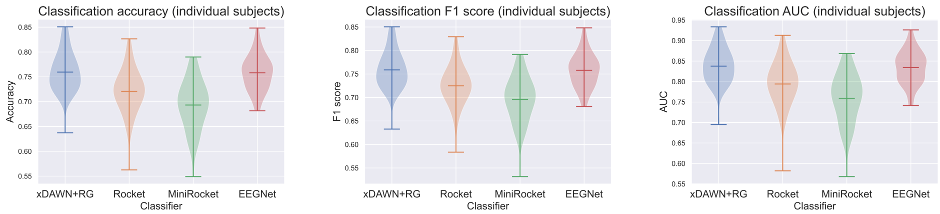


Figure D.2.: Violin plots of the test accuracy, F1-score and AUC for models trained on individual subjects.

Model	Accuracy	F1-score	AUC
xDAWN+RG	0.73 ± 0.04	0.72 ± 0.06	0.81 ± 0.05
MiniRocket	0.69 ± 0.04	0.69 ± 0.05	0.76 ± 0.05
Rocket	0.72 ± 0.05	0.71 ± 0.06	0.79 ± 0.05
EEGNet	**0.76 ± 0.04**	**0.76 ± 0.05**	**0.84 ± 0.05**

Table D.3.: Classifier performance for cross-subject models. Every subject was used as a test subject once, we report the average across all test sets. The best performances are indicated in boldface.

illustrate the performance disparity between the two, we refer to Table D.4 and Figure D.4, which showcase the performance difference by subtracting the CS model's performance from that of the IS model. EEGNet, MiniRocket and Rocket exhibit similar performance for both IS and CS models. However, the xDAWN+RG model demonstrates a noticeable decrease in performance. Given the lower absolute performance of the (Mini)Rocket models compared to EEGNet and xDAWN+RG, we will focus on the latter for the remainder of this discussion. We hypothesize that the inferior performance on CS models when using xDAWN+RG can be attributed to its simpler and lightweight nature. Furthermore, xDAWN+RG works by first calculating the evoked responses for all classes. These can differ significantly from subject to subject, both in P3a peak height and in time [147, 207, 290]. The convolutional nature of EEGNet likely enabled it to capture the temporal dynamics of the elicited responses more effectively across different subjects. It is important to note that the CS models had access to a significantly larger corpus of training data than the IS models, which is part of the reason that they kept up reasonably well with the IS models.

D.3.4 *Analysis of the EEGNet cross-subject model*

This section aims to conduct a further investigation into the CS EEGNet model. We conduct this analysis for the EEGNet model, as it performed the best in both the IS and CS scenario. Furthermore, we include this analysis only for the CS models, as they are the most useful in practice, due to their generalization capabilities. However, the conclusions are similar for the IS models.

D.3.4.1 *Confusion matrices*

First, we investigate whether the model focuses on the correct features to make a prediction. For example, it is possible that we trained a sound detector instead of a target/distractor model. Indeed, conditions Con3 and Con4 contain auditory clues for the target. Theoretically, the model could rely solely on the activation in the auditory stimuli processing region of the brain and achieve acceptable performance. For example, if the model performs perfectly on Con3 and Con4, while predicting all trials belonging to Con1 and Con2 to be distractors (due to the absence of auditory stimuli), it would achieve an accuracy of approximately 75%.

The confusion matrices in Figure D.5 negate this assumption. The model performs comparably in detecting distractors under all conditions. However, the model performs slightly better at identifying targets correctly for Con3 and Con4. The accuracies for specific conditions, shown in Table D.5, also reflect this. Indeed, the accuracies for conditions Con3 and Con4 are higher than the accuracies for Con1 and Con2. We hypothesize that the inclusion of auditory support causes an additional signature in the EEG data, making it easier for the model to recognize targets.

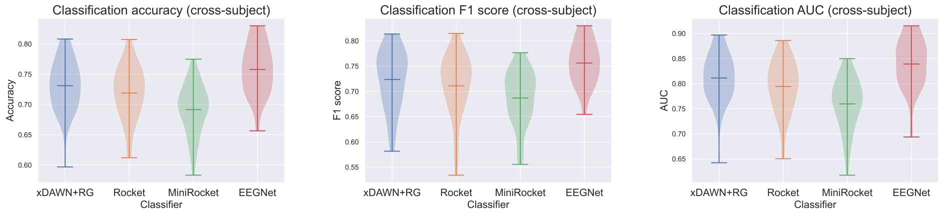


Figure D.3.: Violin plots of the test accuracy, F1-score and AUC for cross-subject models.

Model	**Accuracy**	**F1-score**	**AUC**
xDAWN+RG	0.03 ± 0.02	0.04 ± 0.04	0.03 ± 0.02
MiniRocket	**0.00** ± **0.03**	0.01 ± 0.03	-0.00 ± 0.03
Rocket	**0.00** ± **0.02**	0.01 ± 0.04	-0.00 ± 0.03
EEGNet	**0.00** ± **0.02**	**0.00** ± **0.02**	**-0.01** ± **0.03**

Table D.4.: Drop in performance, calculated by subtracting the test performance of cross-subject models from that of individual subject models. The best performances are indicated in boldface.

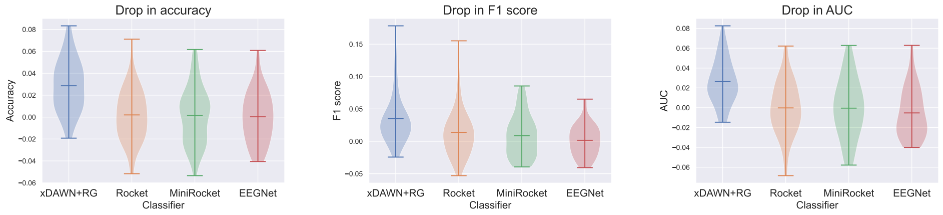


Figure D.4.: Violin plots of the drop in performance, calculated by subtracting the test performance of cross-subject models from that of individual subject models.

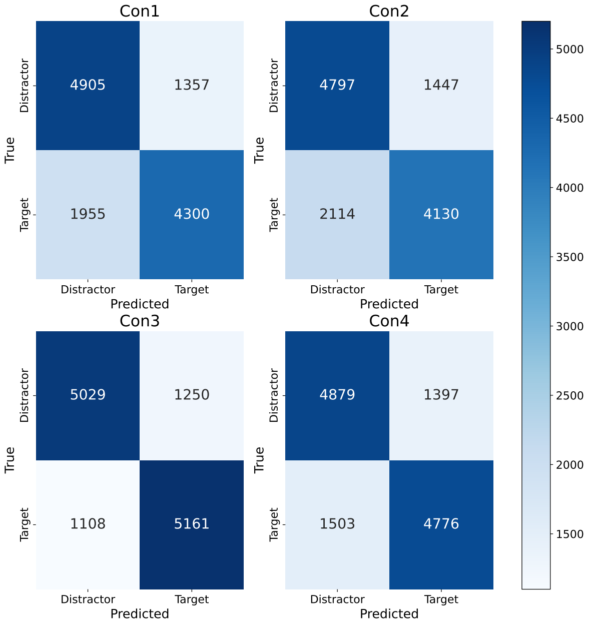


Figure D.5.: Confusion matrices for the cross-subject EEGNet model, split across experimental conditions. The confusion matrices are obtained by aggregating all the test predictions of the CS models.

Additionally, it was already confirmed by a previous analysis that the subjects are able to recall the targets better in conditions with auditory support [93].

D.3.4.2 *Saliency maps*

Next, we explore the electrodes and timings that are predominantly used by our models for making predictions. Trivially, we expect that the model does not use the pre-stimulus ($t < 0$) EEG data. As deep learning methods such as EEGNet are inherently black box models, we resort to xAI methods to obtain (interpretable) insights into the model. A possible technique is a saliency map, which is a visual representation that highlights the degree of importance of regions or features in an input sample on the model prediction [317]. To generate a saliency map, the gradient of the model output with respect to the input sample is computed using backpropagation [121]. More specifically, this process involves fixing the weights of the trained model and propagating the gradient with respect to the layer's inputs back to the first layer that receives the input data.

Figure D.6 shows such a saliency map. This saliency map illustrates the electrodes and timings that had the greatest average impact on the model prediction when identifying a sample as a target. It is computed by first calculating the average saliency map for each test subject individually, then normalizing these saliency maps, and ultimately taking the average across all 42 subjects. In Figure D.7, the same information is repeated, displayed as a topographic map at five time points. From Figure D.6 and Figure D.7, we can see that our model predominantly used the parietal-occipital electrodes and time points between 200ms and 300ms post-stimulus to make its prediction, which is what we expected. We also investigated the saliency maps under different conditions, but noticed no significant difference between the conditions.

D.4 Conclusions

The WithMe project has led to the collection of a large, novel EEG dataset that can be used to create ML methods to automatically detect attention using P3a ERPs in single trial data. This is of great importance to BCIs, as they often rely on the P3a, or, more broadly, the P300 ERP and have a wide range of applications.

We successfully achieved the goal of this work, which was to classify target and distractor stimuli based on the subject's EEG data. To achieve this goal, we studied four classification methods that differed significantly in origin and complexity. We investigated the performance of these methods both as IS and CS models, with the latter the most practically relevant due to its generalization capabilities. For the IS models, xDAWN+RG and EEGNet obtained an accuracy of 76%, outperforming MiniRocket and Rocket. While EEGNet was able to obtain the same accuracy of 76% in the CS case, the accuracy of xDAWN+RG dropped to 0.73%. We attribute this difference to the larger complexity of EEGNet, which likely enables it to generalize better to previously unseen subjects. The drop in performance between IS and CS models is not as pronounced as we expected it to be, and even nonexistent for EEGNet. We attribute this to the fact that the CS models had approximately 42 times more training data available. The EEGNet CS model performed slightly better on samples recorded under conditions Con3 and Con4, which are the conditions that include auditory support. While EEGNet achieved the best performance overall, it also has the highest model complexity (highest number of trainable parameters) and takes the longest time and most compute to train. However, all four models are able to make predictions in real time. This property is essential for real-world human-AI interaction experiments and applications.

Finally, the application of xAI enabled us to investigate which EEG channels and time points were used by the otherwise black-box EEGNet CS model to make its predictions. Indeed, using saliency maps, we conclude that the model primarily based its prediction on the values of the electrodes in the parietal-occipital region between 200ms and 300ms post-stimulus. This is in line with our hypotheses, as we

	Con1	**Con1**	**Con1**	**Con1**
Accuracy	0.74	0.71	0.81	0.77

Table D.5.: The test accuracies of the CS EEGNet model for the different conditions.

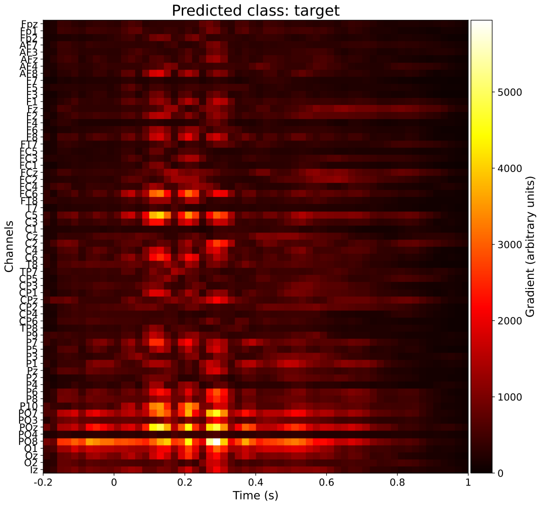


Figure D.6.: Saliency map for epochs labeled as targets by the cross-subject EEGNet model. We averaged normalized saliency maps over all 42 test subjects for the CS model.

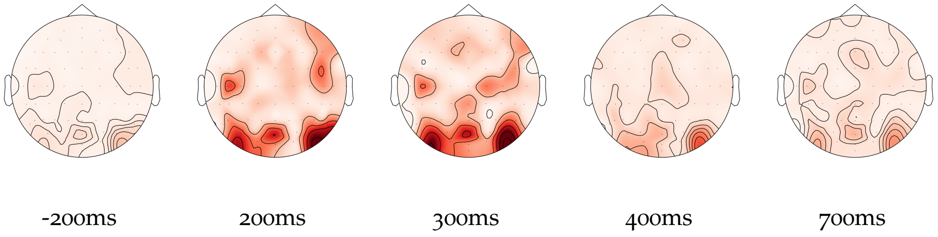


Figure D.7.: The saliency map from Figure D.6 is shown as a topographic map at five different timings. A deeper shade of red indicates a larger gradient. At $t = -0.200$ms and $t = 0.700$ms, the gradients are near zero, indicating that the model does not use these timings. Contrary, for $t \in \{0.200, 0.300, 0.400\}$ms, there is a large gradient in the parietal-occipital region of the brain.

expected to elicit an attention-related P3a ERP in the parietal-occipital region of the brain when the subject saw a target digit.

In conclusion, we achieved the goal of accurately classifying targets and distractors based on a subject's EEG data. At the same time, our work contributes to the development of more effective BCIs and their applications. Finally, we validated the EEG data collected in the WithMe experiment.

While this study provides valuable insights into attention detection using EEG data, it is important to acknowledge some limitations. For example, as mentioned in D.2, part of the data used to train the model was labeled incorrectly, as the ground truth labels are based on the predefined labels of the experiment rather than the subject's perceived class. A possible solution is to limit the data to samples where the entire sequence was reported correctly. However, this would mean that we lose a lot of data, which would in turn decrease the performance of the models. Alternatively, we could remove all "bad sequences", where a bad sequence would be defined as a sequence in which none of the targets were remembered correctly. This could be caused by either incorrectly identifying the stimuli, or by bad memory management, despite correctly identifying the targets and distractors. However, the number of answers that did not include at least one of the target digits (regardless of its place in the sequence) is negligible.

In future work, an experiment dedicated to attention should be used to circumvent the limitations regarding bad labels, as described in Section D.4. This would allow for labels that exactly correspond to the subject's perception of a stimulus, which would in turn lead to more accurate attention detectors. The ultimate goal could then be to use this attention detector in a BCI, to detect whether a subject paid attention. In case they did not, the BCI could repeat the sequence or stimulus, to make sure that the subject can act accordingly. This could also improve learning systems, that is, systems that know whether a student actually paid attention to the provided information [7, 157]. Regarding the training and optimization of ML models, it would be interesting to include an exhaustive feature selection procedure to allow the ML model to focus on the (most) relevant features. Additionally, we want to explore other ways to enable CS generalization, for example using transfer learning [130, 208]. This could further increase the generalization performance of all methods. In particular, this has the potential to elevate the performance of lightweight models such as xDAWN+RG to that of the computationally expensive EEGNet. While this work focuses on the detection of attention using epoched EEG data, the experiment can also be used to study working memory [93]. Indeed, the complete sequence EEG data should permit an investigation regarding working memory and whether it is influenced by auditory and/or rhythmic support.

DATA AVAILABILITY Publicly available datasets were analyzed in this study. The behavioral data can be found here:

https://osf.io/ntmy8/?view_only=88d951c394c7481dba00a1497d64797f.

The preprocessed EEG data are openly available on Figshare at https://doi.org/10.6084/m9.figshare.24278887 [244]. The raw EEG data will be released later as part of a separate publication.

package versions Table D.6 lists the Python libraries used in the computational experiments.

Package	Version	Reference
python	3.9.13	[344]
MNE	1.2.1	[142]
pyriemann	0.3	[25]
torch	1.13.1	[262]
sktime	0.17.1	[216]
scikit-learn	1.1.3	[268]

Table D.6.: The versions of the Python packages used in the project.

Bibliography

[1] Reza Abiri, Soheil Borhani, Yang Jiang, and Xiaopeng Zhao. "Decoding attentional state to faces and scenes using EEG brainwaves." In: *Complexity* 2019 (2019).

[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." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 218–252.

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

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

[5] Swati Aggarwal and Nupur Chugh. "Review of Machine Learning Techniques for EEG Based Brain Computer Interface." In: *Archives of Computational Methods in Engineering* 1 (Jan. 2022), pp. 1–20. ISSN: 18861784. DOI: 10.1007/S11831-021-09684-6/FIGURES/8. URL: https://link.springer.com/article/10.1007/s11831-021-09684-6.

[6] Morsheda Akter, Shafew Ansary, Md Al-Masrur Khan, and Dongwan Kim. "Human activity recognition using attention-mechanism-based deep learning feature combination." In: *Sensors* 23.12 (2023), p. 5715.

[7] Abeer Al-Nafjan and Mashael Aldayel. "Predict Students' Attention in Online Learning Using EEG Data." In: *Sustainability 2022, Vol. 14, Page 6553* 14.11 (May 2022), p. 6553. ISSN: 2071-1050. DOI: 10.3390/SU14116553. URL: https://www.mdpi.com/2071-1050/14/11/6553/htmhttps://www.mdpi.com/2071-1050/14/11/6553.

[8] Fatih Altindis, Bulent Yilmaz, Sergey Borisenok, and Kutay Icoz. "Use of topological data analysis in motor intention based brain-computer interfaces." In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, pp. 1695–1699.

[9] Fatih Altındiş, Bülent Yılmaz, Sergey Borisenok, and Kutay İçöz. "Parameter investigation of topological data analysis for EEG signals." In: *Biomedical Signal Processing and Control* 63 (2021), p. 102196.

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

[11] Keri L Anderson, Jeffrey S Anderson, Sourabh Palande, and Bei Wang. "Topological data analysis of functional MRI connectivity in time and space domains." In: *International Workshop on Connectomics in Neuroimaging*. Springer. 2018, pp. 67–77.

[12] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. "A public domain dataset for human activity recognition using smartphones." In: *Esann*. Vol. 3. 2013, p. 3.

[13] Rushil Anirudh, Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan Turaga. "A Riemannian framework for statistical analysis of topological persistence diagrams." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 68–76.

[14] Aras Asaad and Sabah Jassim. "Topological data analysis for image tampering detection." In: *International Workshop on Digital Watermarking*. Springer. 2017, pp. 136–146.

[15] Zülfikar Aslan and Mehmet Akin. "A deep learning approach in automated detection of schizophrenia using scalogram images of EEG signals." In: *Physical and Engineering Sciences in Medicine* 45.1 (Mar. 2022), pp. 83–96. ISSN: 26624737. DOI: 10.1007/S13246-021-01083-2/TABLES/4. URL: https://link.springer.com/article/10.1007/s13246-021-01083-2.

[16] Eda Akman Aydin, Omer Faruk Bay, and Inan Guler. "P300-based asynchronous brain computer interface for environmental control system." In: *IEEE Journal of Biomedical and Health Informatics* 22.3 (May 2018), pp. 653–663. ISSN: 21682208. DOI: 10.1109/JBHI.2017.2690801.

[17] Fatemeh Bahari and Amin Janghorbani. "EEG-based emotion recognition using recurrence plot analysis and K nearest neighbor classifier." In: *2013 20th Iranian Conference on Biomedical Engineering (ICBME)*. IEEE. 2013, pp. 228–233.

[18] Alexandre Barachant and Stephane Bonnet. "Channel selection procedure using Riemannian distance for BCI applications." In: *2011 5th International IEEE/EMBS Conference on Neural Engineering*. IEEE. 2011, pp. 348–351.

[19] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. "Multiclass brain–computer interface classification by Riemannian geometry." In: *IEEE Transactions on Biomedical Engineering* 59.4 (2011), pp. 920–928.

[20] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. "Classification of covariance matrices using a Riemannian-based kernel for BCI applications." In: *Neurocomputing* 112 (2013), pp. 172–178.

[21] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. "Multiclass brain-computer interface classification by Riemannian geometry." In: *IEEE Transactions on Biomedical Engineering* 59.4 (Apr. 2012), pp. 920–928. ISSN: 00189294. DOI: 10.1109/TBME.2011.2172210.

[22] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. "Classification of covariance matrices using a Riemannian-based kernel for BCI applications." In: *Neurocomputing* 112 (2013), pp. 172–178. DOI: `10.1016/j.neucom.2012.12.039{\"{i}}`. URL: `https://hal.science/hal-00820475`.

[23] Alexandre Barachant and Marco Congedo. "A plug&play P300 BCI using information geometry." In: *arXiv preprint arXiv:1409.0107* (2014).

[24] Alexandre Barachant and Marco Congedo. "A plug&play P300 BCI using information geometry." In: *arXiv preprint arXiv:1409.0107* (2014).

[25] Alexandre Barachant et al. *pyRiemann/pyRiemann: v0.3*. July 2022. DOI: `10.5281/zenodo.7547583`. URL: `https://doi.org/10.5281/zenodo.7547583`.

[26] Danielle S Bassett, Brent G Nelson, Bryon A Mueller, Jazmin Camchong, and Kelvin O Lim. "Altered resting state complexity in schizophrenia." In: *NeuroImage* 59.3 (2012), pp. 2196–2207.

[27] André M Bastos and Jan-Mathijs Schoffelen. "A tutorial review of functional connectivity analysis methods and their interpretational pitfalls." In: *Frontiers in Systems Neuroscience* 9 (2016), p. 175.

[28] Ulrich Bauer. "Ripser: efficient computation of Vietoris-Rips persistence barcodes." In: *Journal of Applied and Computational Topology* (2021). DOI: `10.1007/s41468-021-00071-5`.

[29] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. "Persistent homology analysis of brain artery trees." In: *Annals of Applied Statistics* 10.1 (2016), p. 198.

[30] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. "Testing convexity of figures under the uniform distribution." In: *Random Structures & Algorithms* 54.3 (2019), pp. 413–443.

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

[32] Paul J Besl and Neil D McKay. "A Method for Registration of 3-D Shapes." In: *Sensor Fusion IV: Control Paradigms and Data Structures*. Vol. 1611. Spie. 1992, pp. 586–606.

[33] R. Bhatia, T. Jain, and Y. Lim. "On the Bures–Wasserstein Distance between Positive Definite Matrices." In: *Expositiones Mathematicae* 37.2 (2019), pp. 165–191.

[34] Bryan Bischof and Eric Bunch. "Geometric feature performance under downsampling for EEG classification tasks." In: *arXiv preprint arXiv:2102.07669* (2021).

[35] Christophe AN Biscio and Jesper Møller. "The accumulated persistence function, a new useful functional summary statistic for topological data analysis, with a view to brain artery trees and spatial point process applications." In: *Journal of Computational and Graphical Statistics* (2019), pp. 1–21.

[36] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. "Single-trial analysis and classification of ERP components—a tutorial." In: *NeuroImage* 56.2 (2011), pp. 814–825.

[37] Andrew J Blumberg and Michael Lesnick. "Stability of 2-parameter Persistent Homology." In: *Foundations of Computational Mathematics* (2022), pp. 1–43.

[38] Omer Bobrowski and Sayan Mukherjee. "The topology of probability distributions on manifolds." In: *Probability Theory and Related Fields* 161.3 (2015), pp. 651–686.

[39] Pawel Boguslawski. *Modelling and analysing 3D building interiors with the dual half-edge data structure*. University of South Wales (United Kingdom), 2011.

[40] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Vol. 57. Cambridge University Press, 2018.

[41] 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." In: *The Anatomical Record* 298.1 (2015), pp. 249–276.

[42] Jacob Brown and Tomáš Gedeon. "Structure of the afferent terminals in terminal ganglion of a cricket and persistent homology." In: *PLOS ONE* 7.5 (2012), e37278.

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

[44] Peter Bubenik. "Statistical topological data analysis using persistence landscapes." In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 77–102.

[45] Peter Bubenik and Paweł Dłotko. "A persistence landscapes toolbox for topological statistics." In: *Journal of Symbolic Computation* 78 (2017), pp. 91–114.

[46] Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. "Persistent homology detects curvature." In: *Inverse Problems* 36.2 (2020), p. 025008.

[47] Ed Bullmore and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems." In: *Nature Reviews Neuroscience* 10.3 (2009), pp. 186–198.

[48] Erik Van der Burg, Christian N.L. Olivers, Adelbert W. Bronkhorst, and Jan Theeuwes. "Pip and Pop: Nonspatial Auditory Signals Improve Spatial Visual Search." In: *Journal of Experimental Psychology: Human Perception and Performance* 34.5 (Oct. 2008), pp. 1053–1065. ISSN: 00961523. DOI: 10.1037/0096-1523.34.5.1053.

[49] Erik Van der Burg, Christian NL Olivers, Adelbert W Bronkhorst, and Jan Theeuwes. "Pip and pop: nonspatial auditory signals improve spatial visual search." In: *Journal of Experimental Psychology: Human Perception and Performance* 34.5 (2008), p. 1053.

[50] Niko A Busch and Rufin VanRullen. “Spontaneous EEG oscillations reveal periodic sampling of visual attention.” In: *Proceedings of the National Academy of Sciences* 107.37 (2010), pp. 16048–16053.

[51] Zixuan Cang, Lin Mu, and Guo-Wei Wei. “Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening.” In: *PLOS Computational Biology* 14.1 (2018), e1005929.

[52] Zixuan Cang and Guo-Wei Wei. “TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions.” In: *PLOS Computational Biology* 13.7 (2017), e1005690.

[53] Luigi Caputi, Anna Pidnebesna, and Jaroslav Hlinka. “Promises and pitfalls of Topological Data Analysis for brain connectivity analysis.” In: *NeuroImage* (2021), p. 118245.

[54] Gunnar Carlsson. “Topological pattern recognition for point cloud data.” In: *Acta Numerica* 23 (2014), pp. 289–368.

[55] Gunnar Carlsson and Afra Zomorodian. “The theory of multidimensional persistence.” In: *Discrete & Computational Geometry* 42.1 (2009), pp. 71–93.

[56] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. “PersLay: A neural network layer for persistence diagrams and new graph topological signatures.” In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2786–2796.

[57] Mathieu Carriere, Marco Cuturi, and Steve Oudot. “Sliced Wasserstein kernel for persistence diagrams.” In: *arXiv preprint arXiv:1706.03358* (2017).

[58] Mathieu Carrière, Steve Y Oudot, and Maks Ovsjanikov. “Stable topological signatures for points on 3D shapes.” In: *Computer Graphics Forum*. Vol. 34. 5. Wiley Online Library. 2015, pp. 1–12.

[59] Frédéric Cazals and Marc Pouget. “Estimating differential quantities using polynomial fitting of osculating jets.” In: *Computer Aided Geometric Design* 22.2 (2005), pp. 121–146.

[60] Wojciech Chachólski and Henri Riihimaki. “Metrics and stabilization in one parameter persistence.” In: *SIAM Journal on Applied Algebra and Geometry* 4.1 (2020), pp. 69–98.

[61] Frédéric Chazal and David Cohen-Steiner. *Geometric inference*. 2013.

[62] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. “Proximity of persistence modules and their diagrams.” In: *Proceedings of the twenty-fifth Annual Symposium on Computational Geometry*. ACM. 2009, pp. 237–246.

[63] Frédéric Chazal, David Cohen-Steiner, Leonidas J Guibas, Facundo Mémoli, and Steve Y Oudot. “Gromov-Hausdorff stable signatures for shapes using persistence.” In: *Computer Graphics Forum*. Vol. 28. 5. Wiley Online Library. 2009, pp. 1393–1403.

[64] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. "Geometric inference for probability measures." In: *Foundations of Computational Mathematics* 11.6 (2011), pp. 733–751.

[65] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Vol. 10. Springer, 2016.

[66] Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. "On the bootstrap for persistence diagrams and landscapes." In: *arXiv preprint arXiv:1311.0376* (2013).

[67] 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." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 5845–5884.

[68] Frédéric Chazal, Leonidas J Guibas, Steve Y Oudot, and Primoz Skraba. "Scalar field analysis over point cloud data." In: *Discrete & Computational Geometry* 46.4 (2011), pp. 743–775.

[69] Frédéric Chazal and Bertrand Michel. "An introduction to topological data analysis: fundamental and practical aspects for data scientists." In: *Frontiers in artificial intelligence* 4 (2021), p. 667963.

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

[71] Chao Chen and Michael Kerber. "Persistent homology computation with a twist." In: *Proceedings 27th European Workshop on Computational Geometry*. Vol. 11. 2011, pp. 197–200.

[72] 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*. PMLR. 2019, pp. 2573–2582.

[73] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. "A Benchmark for 3D Mesh Segmentation." In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3 (Aug. 2009).

[74] Yang Chen and Gérard Medioni. "Object Modelling by Registration of Multiple Range Images." In: *Image and vision computing* 10.3 (1992), pp. 145–155.

[75] 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*. Springer. 2019, pp. 16–28.

[76] 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." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[77] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Stability of persistence diagrams." In: *Discrete & Computational Geometry* 37.1 (2007), pp. 103–120.

[78] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. "Lipschitz functions have L_p-stable persistence." In: *Foundations of Computational Mathematics* 10.2 (2010), pp. 127–139.

[79] Mike X Cohen. *Analyzing neural time series data: Theory and practice*. MIT press, 2014.

[80] Mike X Cohen. *Analyzing neural time series data: theory and practice*. MIT press, 2014.

[81] Anne Collins, Afra Zomorodian, Gunnar Carlsson, and Leonidas J Guibas. "A barcode shape descriptor for curve point cloud data." In: *Computers & Graphics* 28.6 (2004), pp. 881–894.

[82] Marco Congedo, Alexandre Barachant, and Anton Andreev. "A new generation of brain-computer interface based on Riemannian geometry." In: *arXiv preprint arXiv:1310.8115* (2013).

[83] Marco Congedo, Alexandre Barachant, and Anton Andreev. "A new generation of brain-computer interface based on riemannian geometry." In: *arXiv preprint arXiv:1310.8115* (2013).

[84] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review." In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.

[85] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review." In: *http://dx.doi.org/10.1080/2326263X.2017.1297192* 4.3 (July 2017), pp. 155–174. ISSN: 23262621. DOI: 10.1080/2326263X.2017.1297192. URL: https://www.tandfonline.com/doi/abs/10.1080/2326263X.2017.1297192.

[86] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. "Deep learning for electroencephalogram (EEG) classification tasks: a review." In: *Journal of Neural Engineering* 16.3 (2019), p. 031001.

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

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

[89] Carina Curto. "What can topology tell us about the neural code?" In: *Bulletin of the American Mathematical Society* 54.1 (2017), pp. 63–78.

[90] Arup Kumar Das, Kriti Kumar, Rahul D Gavas, Dibyanshu Jaiswal, Debatri Chatterjee, Ramesh Kumar Ramakrishnan, M Girish Chandra, and Arpan Pal. "Cognitive Fatigue Detection from EEG Signals using Topological Signal Processing." In: *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE. 2020, pp. 1313–1317.

[91] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Annh Ratanamahatana, and Eamonn Keogh. "The UCR Time Series Archive." In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (Oct. 2018), pp. 1293–1305. ISSN: 23299274.

[92] Jorg De Winne, Paul Devos, Marc Leman, and Dick Botteldooren. "With No Attention Specifically Directed to It, Rhythmic Sound Does Not Automatically Facilitate Visual Task Performance." In: *Frontiers in Psychology* 13 (2022), p. 894366.

[93] Jorg De Winne, Paul Devos, Marc Leman, and Dick Botteldooren. "With No Attention Specifically Directed to It, Rhythmic Sound Does Not Automatically Facilitate Visual Task Performance." In: *Frontiers in Psychology* 13 (June 2022), p. 3221. ISSN: 16641078. DOI: 10.3389/FPSYG.2022.894366/BIBTEX.

[94] Tamer Demiralp, Ahmet Ademoglu, Yorgo Istefanopulos, Canan Başar-Eroglu, and Erol Başar. "Wavelet analysis of oddball P300." In: *International Journal of Psychophysiology* 39.2-3 (Jan. 2001), pp. 221–227. ISSN: 0167-8760. DOI: 10.1016/S0167-8760(00)00143-4.

[95] Angus Dempster, François Petitjean, and Geoffrey I. Webb. "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels." In: *Data Mining and Knowledge Discovery* 34.5 (Sept. 2020), pp. 1454–1495. ISSN: 1573756X.

[96] Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. "MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Dec. 2020), pp. 248–257. DOI: 10.1145/3447548.3467231. URL: http://arxiv.org/abs/2012.08791http://dx.doi.org/10.1145/3447548.3467231.

[97] Tamal K Dey, Herbert Edelsbrunner, and Sumanta Guha. "Computational topology." In: *Contemporary mathematics* 223 (1999), pp. 109–144.

[98] Tamal Krishna Dey and Yusu Wang. *Computational topology for data analysis*. Cambridge University Press, 2022.

[99] Barbara Di Fabio and Massimo Ferri. "Comparing persistence diagrams through complex vectors." In: *Image Analysis and Processing—ICIAP 2015: 18th International Conference, Genoa, Italy, September 7-11, 2015, Proceedings, Part I 18*. Springer. 2015, pp. 294–305.

[100] Meryll Dindin, Yuhei Umeda, and Frédéric Chazal. "Topological data analysis for arrhythmia detection through modular neural networks." In: *Canadian Conference on Artificial Intelligence*. Springer. 2020, pp. 177–188.

[101] Paweł Dłotko and Hubert Wagner. "Simplification of complexes for persistent homology computations." In: *Homology, Homotopy and Applications* 16.1 (2014), pp. 49–63.

[102] Paweł Dłotko and Thomas Wanner. "Rigorous cubical approximation and persistent homology of continuous functions." In: *Computers & Mathematics with Applications* 75.5 (2018), pp. 1648–1666.

[103] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian geometry*. Vol. 6. Springer, 1992.

[104] E. Donchin. "Surprise!... Surprise?" In: *Psychophysiology* 18.5 (Sept. 1981), pp. 493–513. ISSN: 1469-8986. DOI: 10.1111/J.1469-8986.1981.TB01815.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1469-8986.1981.tb01815.xhttps://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.1981.tb01815.xhttps://onlinelibrary.wiley.com/doi/10.1111/j.1469-8986.1981.tb01815.x.

[105] Harish Doraiswamy, Julien Tierny, Paulo JS Silva, Luis Gustavo Nonato, and Claudio Silva. "TopoMap: A 0-dimensional homology preserving projection of high-dimensional data." In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020), pp. 561–571.

[106] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. "Gradient Descent Finds Global Minima of Deep Neural Networks." In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1675–1685. URL: https://proceedings.mlr.press/v97/du19c.html.

[107] Connie C Duncan, Robert J Barry, John F Connolly, Catherine Fischer, Patricia T Michie, Risto Näätänen, John Polich, Ivar Reinvang, and Cyma Van Petten. "Event-related potentials in clinical research: guidelines for eliciting, recording, and quantifying mismatch negativity, P300, and N400." In: *Clinical Neurophysiology* 120.11 (2009), pp. 1883–1908.

[108] Kusumika Krori Dutta. "Multi-class time series classification of EEG signals with recurrent neural networks." In: *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019* (Jan. 2019), pp. 337–341. DOI: 10.1109/CONFLUENCE.2019.8776889.

[109] Edelsbrunner, Letscher, and Zomorodian. "Topological persistence and simplification." In: *Discrete & computational geometry* 28 (2002), pp. 511–533.

[110] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[111] Herbert Edelsbrunner and John Harer. "Persistent homology-a survey." In: *Contemporary Mathematics* 453 (2008), pp. 257–282.

[112] Herbert Edelsbrunner and John Harer. *Computational topology: An introduction*. American Mathematical Society, 2010.

[113] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. "On the shape of a set of points in the plane." In: *IEEE Transactions on Information Theory* 29.4 (1983), pp. 551–559.

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

[115] Herbert Edelsbrunner and Ernst P Mücke. "Three-dimensional alpha shapes." In: *ACM Transactions on Graphics (TOG)* 13.1 (1994), pp. 43–72.

[116] Asaf Elbaz and Yaffa Yeshurun. "Can rhythm-induced attention improve the perceptual representation?" In: *PLOS ONE* 15.4 (2020), e0231200.

[117] Steven P Ellis and Arno Klein. "Describing high-order statistical dependence using "concurrence topology", with application to functional MRI brain data." In: *Homology, Homotopy and Applications* 16.1 (2014), pp. 245–264.

[118] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. "Exploring the Landscape of Spatial Robustness." In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1802–1811. URL: `https://proceedings.mlr.press/v97/engstrom19a.html`.

[119] Charles Epstein, Gunnar Carlsson, and Herbert Edelsbrunner. "Topological data analysis." In: *Inverse Problems* 27.12 (2011), p. 120201.

[120] Paul Expert, Louis-David Lord, Morten L Kringelbach, and Giovanni Petri. *Topological neuroscience*. 2019.

[121] Amr Farahat, Christoph Reichert, Catherine M. Sweeney-Reed, and Hermann Hinrichs. "Convolutional neural networks for decoding of covert attention focus and saliency maps for EEG feature visualization." In: *Journal of Neural Engineering* 16.6 (Oct. 2019), p. 066010. ISSN: 1741-2552. DOI: `10.1088/1741-2552/AB3BB4`. URL: `https://iopscience.iop.org/article/10.1088/1741-2552/ab3bb4https://iopscience.iop.org/article/10.1088/1741-2552/ab3bb4/meta`.

[122] Brittany Fasy, Yu Qin, Brian Summa, and Carola Wenk. "Comparing distance metrics on vectorized persistence summaries." In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. 2020.

[123] Siamac Fazli, Cristian Grozea, Márton Danóczy, Florin Popescu, Benjamin Blankertz, and Klaus-Robert Müller. "Subject independent EEG-based BCI decoding." In: *Advances in Neural Information Processing Systems* 22 (2009).

[124] Rémi Flamary et al. "POT: Python Optimal Transport." In: *Journal of Machine Learning Research* 22.78 (2021), pp. 1–8. URL: `http://jmlr.org/papers/v22/20-451.html`.

[125] Matteo Fraschini, Matteo Demuru, Arjan Hillebrand, Lorenza Cuccu, Silvia Porcu, Francesca Di Stefano, Monica Puligheddu, Gianluca Floris, Giuseppe Borghero, and Francesco Marrosu. "EEG functional network topology is associated with disability in patients with amyotrophic lateral sclerosis." In: *Scientific Reports* 6.1 (2016), pp. 1–7.

[126] Akira Furusaki. *RIKEN Center for Emergent Matter Science Quantum Matter Theory Research Team*. `https://cems.riken.jp/en/laboratory/qmtrt` Data of access: 29-01-2024.

[127] Valentin Gabeff, Tomas Teijeiro, Marina Zapater, Leila Cammoun, Sylvain Rheims, Philippe Ryvlin, and David Atienza. "Interpreting deep learning models for epileptic seizure detection on EEG signals." In: *Artificial Intelligence in Medicine* 117 (July 2021), p. 102084. ISSN: 0933-3657. DOI: `10.1016/J.ARTMED.2021.102084`.

[128] Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. "A Topology Layer for Machine Learning." In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1553–1563. URL: `https://proceedings.mlr.press/v108/gabrielsson20a.html`.

[129] 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)*. IEEE. 2019, pp. 1551–1556.

[130] Nathalie TH Gayraud, Alain Rakotomamonjy, and Maureen Clerc. "Optimal transport applied to transfer learning for P300 detection." In: *BCI 2017-7th Graz Brain-Computer Interface Conference*. 2017, p. 6.

[131] Lulu Ge and Keshab K Parhi. "Classification using hyperdimensional computing: A review." In: *IEEE Circuits and Systems Magazine* 20.2 (2020), pp. 30–47.

[132] Joseph Geraghty and George Schoettle. "Single-Subject vs. Cross-Subject Motor Imagery Models." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13519 LNCS (2022), pp. 442–452. ISSN: 16113349. DOI: `10.1007/978-3-031-17618-0{\_}31/TABLES/11`. URL: `https://link.springer.com/chapter/10.1007/978-3-031-17618-0_31`.

[133] Hamza Ghadyali. "Applications of topological data analysis and sliding window embeddings for learning on novel features of time-varying dynamical systems." PhD thesis. Duke University, 2017.

[134] Parisa Ghane, Narges Zarnaghinaghsh, and Ulisses Braga-Neto. "Comparison of Classification Algorithms Towards Subject-Specific and Subject-Independent BCI." In: *9th IEEE International Winter Conference on Brain-Computer Interface, BCI 2021* (Feb. 2021). DOI: `10.1109/BCI51272.2021.9385339`.

[135] Robert W Ghrist. *Elementary applied topology*. Vol. 1. Createspace Seattle, 2014.

[136] Robert Ghrist, Rachel Levanger, and Huy Mai. "Persistent homology and Euler integral transforms." In: *Journal of Applied and Computational Topology* 2.1 (2018), pp. 55–60.

[137] Chad Giusti, Robert Ghrist, and Danielle S Bassett. "Two's company, three (or more) is a simplex." In: *Journal of computational neuroscience* 41.1 (2016), pp. 1–14.

[138] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. "Clique topology reveals intrinsic geometric structure in neural correlations." In: *Proceedings of the National Academy of Sciences* 112.44 (2015), pp. 13455–13460.

[139] Stephen M. Gordon, Matthew Jaswa, Amelia J. Solon, and Vernon J. Lawhern. "Real world BCI: Cross-domain learning and practical applications." In: *BCI-forReal 2017 - Proceedings of the 2017 ACM Workshop on An Application-Oriented Approach to BCI Out of the Laboratory, co-located with IUI 2017* (Mar. 2017), pp. 25–28. DOI: 10.1145/3038439.3038444. URL: https://dl.acm.org/doi/10.1145/3038439.3038444.

[140] Zeus Gracia-Tabuenca, Juan Carlos Díaz-Patiño, Isaac Arelio-Ríos, Martha Beatriz Moreno-García, Fernando A Barrios, and Sarael Alcauter. "Development of the Functional Connectome Topology in Adolescence: Evidence from Topological Data Analysis." In: *eNeuro* 10.2 (2023).

[141] Zeus Gracia-Tabuenca, Juan Carlos Díaz-Patiño, Isaac Arelio, and Sarael Alcauter. "Topological data analysis reveals robust alterations in the whole-brain and frontal lobe functional connectomes in attention-deficit/hyperactivity disorder." In: *eNeuro* 7.3 (2020).

[142] Alexandre Gramfort et al. "MEG and EEG data analysis with MNE-Python." In: *Frontiers in Neuroscience* 0.7 DEC (2013), p. 267. ISSN: 1662453X. DOI: 10.3389/FNINS.2013.00267/BIBTEX.

[143] Heather M Gray, Nalini Ambady, William T Lowenthal, and Patricia Deldin. "P300 as an index of attention to self-relevant stimuli." In: *Journal of experimental social psychology* 40.2 (2004), pp. 216–224.

[144] Heather M. Gray, Nalini Ambady, William T. Lowenthal, and Patricia Deldin. "P300 as an index of attention to self-relevant stimuli." In: *Journal of Experimental Social Psychology* 40.2 (Mar. 2004), pp. 216–224. ISSN: 0022-1031. DOI: 10.1016/S0022-1031(03)00092-1.

[145] David Griffiths. *Point cloud classification with PointNet*. https://keras.io/examples/vision/pointnet/. Accessed: 2022-02-01.

[146] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. "PCPNET learning local shape properties from raw point clouds." In: *Computer Graphics Forum*. Vol. 37. 2. Wiley Online Library. 2018, pp. 75–85.

[147] Christoph Guger, Shahab Daban, Eric Sellers, Clemens Holzner, Gunther Krausz, Roberta Carabalona, Furio Gramatica, and Guenter Edlinger. "How many people are able to control a P300-based brain–computer interface (BCI)?" In: *Neuroscience Letters* 462.1 (Sept. 2009), pp. 94–98. ISSN: 0304-3940. DOI: `10.1016/J.NEULET.2009.06.045`.

[148] Guangxing Guo, Yanli Zhao, Chenxu Liu, Yongcan Fu, Xinhua Xi, Lizhong Jin, Dongli Shi, Lin Wang, Yonghong Duan, Jie Huang, et al. "Method for persistent topological features extraction of schizophrenia patients' electroencephalography signal based on persistent homology." In: *Frontiers in Computational Neuroscience* 16 (2022), p. 1024205.

[149] William H Guss and Ruslan Salakhutdinov. "On characterizing the capacity of neural networks using algebraic topology." In: *arXiv preprint arXiv:1802.04443* (2018).

[150] Violaine Guy, Marie Hélène Soriani, Mariane Bruno, Théodore Papadopoulo, Claude Desnuelle, and Maureen Clerc. "Brain computer interface with the P300 speller: Usability for disabled people with amyotrophic lateral sclerosis." In: *Annals of Physical and Rehabilitation Medicine* 61.1 (Jan. 2018), pp. 5–11. ISSN: 1877-0657. DOI: `10.1016/J.REHAB.2017.09.004`.

[151] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. "GANSpace: Discovering Interpretable GAN Controls." In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 9841–9850. URL: `https://proceedings.neurips.cc/paper_files/paper/2020/file/6fe43269967adbb64ec6149852b5cc3e-Paper.pdf`.

[152] Thalía Harmony, Thalía Fernández, Juan Silva, Jorge Bernal, Lourdes Díaz-Comas, Alfonso Reyes, Erzsébet Marosi, Mario Rodríguez, and Miguel Rodríguez. "EEG delta activity: an indicator of attention to internal processing during performance of mental tasks." In: *International Journal of Psychophysiology* 24.1-2 (1996), pp. 161–171.

[153] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

[154] Michael Hauser and Asok Ray. "Principles of Riemannian Geometry in Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper_files/paper/2017/file/0ebcc77dc72360d0eb8e9504c78d38bd-Paper.pdf`.

[155] Tong He, Haibin Huang, Li Yi, Yuqian Zhou, Chihao Wu, Jue Wang, and Stefano Soatto. "GeoNet: Deep geodesic networks for point cloud analysis." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6888–6897.

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

[157] Bin Hu, Xiaowei Li, Shuting Sun, and Martyn Ratcliffe. "Attention Recognition in EEG-Based Affective Learning Research Using CFS+KNN Algorithm." In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 15.1 (Jan. 2018), pp. 38–45. ISSN: 15455963. DOI: 10.1109/TCBB.2016.2616395.

[158] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. "Topology-preserving deep image segmentation." In: *Advances in Neural Information Processing Systems* 32 (2019).

[159] Hanne Huygelier, Raymond van Ee, Armien Lanssens, Johan Wagemans, and Céline R Gillebert. "Audiovisual looming signals are not always prioritised: evidence from exogenous, endogenous and sustained attention." In: *Journal of Cognitive Psychology* 33.3 (2021), pp. 282–303.

[160] Shiro Ikegami, Kouji Takano, Kiyohiko Kondo, Naokatsu Saeki, and Kenji Kansaku. "A region-based two-step P300-based brain–computer interface for patients with amyotrophic lateral sclerosis." In: *Clinical Neurophysiology* 125.11 (Nov. 2014), pp. 2305–2312. ISSN: 1388-2457. DOI: 10.1016/J.CLINPH.2014.03.013.

[161] Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing. "VoiceHD: Hyperdimensional computing for efficient speech recognition." In: *2017 IEEE international conference on rebooting computing (ICRC)*. IEEE. 2017, pp. 1–8.

[162] Mohammed Saidul Islam, Iqram Hussain, Md Mezbaur Rahman, Se Jin Park, and Md Azam Hossain. "Explainable Artificial Intelligence Model for Stroke Prediction Using EEG Signal." In: *Sensors 2022, Vol. 22, Page 9859* 22.24 (Dec. 2022), p. 9859. ISSN: 1424-8220. DOI: 10.3390/S22249859. URL: https://www.mdpi.com/1424-8220/22/24/9859/htmhttps://www.mdpi.com/1424-8220/22/24/9859.

[163] Umar Islambekov, Monisha Yuvaraj, and Yulia R Gel. "Harnessing the power of Topological Data Analysis to detect change points in time series." In: *arXiv preprint arXiv:1910.12939* (2019).

[164] Mahdi Jalili. "Functional brain networks: Does the choice of dependency estimator and binarization method matter?" In: *Scientific Reports* 6.1 (2016), pp. 1–12.

[165] Mahdi Jalili and Maria G Knyazeva. "Constructing brain functional networks from EEG: Partial and unpartial correlations." In: *Journal of Integrative Neuroscience* 10.02 (2011), pp. 213–232.

[166] Mahdi Jalili and Maria G Knyazeva. "EEG-based functional networks in schizophrenia." In: *Computers in Biology and Medicine* 41.12 (2011), pp. 1178–1186.

[167] TWP Janssen, A Hillebrand, A Gouw, K Geladé, R Van Mourik, Athanasios Maras, and J Oosterlaan. "Neural network topology in ADHD; evidence for maturational delay and default-mode network alterations." In: *Clinical Neurophysiology* 128.11 (2017), pp. 2258–2267.

[168] 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*. Springer. 2016, pp. 193–203.

[169] Christina Yi Jin, Jelmer P Borst, and Marieke K van Vugt. "Predicting task-general mind-wandering with EEG." In: *Cognitive, Affective, & Behavioral Neuroscience* 19.4 (2019), pp. 1059–1073.

[170] R Johnson. "The amplitude of the P300 component of the event-related potential: Review and synthesis." In: *Advances in psychophysiology* 3.April (1988), pp. 69–137.

[171] Amir Joudaki, Niloufar Salehi, Mahdi Jalili, and Maria G Knyazeva. "EEG-based functional brain networks: Does the network size matter?" In: *PLOS ONE* 7.4 (2012), e35673.

[172] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational homology*. Vol. 3. 7. Springer, 2004.

[173] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational homology*. Vol. 157. Springer Science & Business Media, 2006.

[174] Daniel Kahneman. *Attention and effort*. Vol. 1063. Citeseer, 1973.

[175] Jules Raymond Kala, Serestina Viriri, Deshendran Moodley, and Jules Raymond Tapamo. "Leaf classification using convexity measure of polygons." In: *International Conference on Image and Signal Processing*. Springer. 2016, pp. 51–60.

[176] Lida Kanari, Paweł Dłotko, Martina Scolamiero, Ran Levi, Julian Shillcock, Kathryn Hess, and Henry Markram. "A topological representation of branching neuronal morphologies." In: *Neuroinformatics* 16.1 (2018), pp. 3–13.

[177] Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.

[178] Pentti Kanerva. "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors." In: *Cognitive computation* 1.2 (2009), pp. 139–159.

[179] N Kannathal, Min Lim Choo, U Rajendra Acharya, and PK Sadasivan. "Entropies for detection of epilepsy in EEG." In: *Computer Methods and Programs in Biomedicine* 80.3 (2005), pp. 187–194.

[180] Leonid V Kantorovich. "On the translocation of masses." In: *Journal of Mathematical Sciences* 133.4 (2006), pp. 1381–1382.

[181] Matthias Kaper, Peter Meinicke, Ulf Grossekathoefer, Thomas Lingner, and Helge Ritter. "BCI competition 2003 - Data set IIb: Support vector machines for the P300 speller paradigm." In: *IEEE Transactions on Biomedical Engineering* 51.6 (June 2004), pp. 1073–1076. ISSN: 00189294. DOI: 10.1109/TBME.2004.826698.

[182] Yufeng Ke, Hongzhi Qi, Feng He, Shuang Liu, Xin Zhao, Peng Zhou, Lixin Zhang, and Dong Ming. "An EEG-based mental workload estimator trained on working memory task can work well under simulated multi-attribute task." In: *Frontiers in human neuroscience* 8 (2014), p. 703.

[183] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. *The UCI Machine Learning Repository*. https://archive.ics.uci.edu.

[184] Arshi Khalid, Byung Sun Kim, Moo K Chung, Jong Chul Ye, and Daejong Jeon. "Tracing the evolution of multi-scale functional networks in a mouse model of depression using persistent brain network homology." In: *NeuroImage* 101 (2014), pp. 351–363.

[185] Hanni Kiiski, Laura M Rueda-Delgado, Marc Bennett, Rachel Knight, Laura Rai, Darren Roddy, Katie Grogan, Jessica Bramham, Clare Kelly, and Robert Whelan. "Functional EEG connectivity is a neuromarker for adult attention deficit hyperactivity disorder symptoms." In: *Clinical Neurophysiology* 131.1 (2020), pp. 330–342.

[186] Eunkyung Kim, Hyejin Kang, Hyekyoung Lee, Hyo-Jeong Lee, Myung-Whan Suh, Jae-Jin Song, Seung-Ha Oh, and Dong Soo Lee. "Morphological brain network assessed using graph theory and network filtration in deaf adults." In: *Hearing Research* 315 (2014), pp. 88–98.

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

[188] Ron Kimmel and Xue-Cheng Tai. *Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*. Elsevier, 2019.

[189] Denis Kleyko, Abbas Rahimi, Dmitri A Rachkovskij, Evgeny Osipov, and Jan M Rabaey. "Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics." In: *IEEE transactions on neural networks and learning systems* 29.12 (2018), pp. 5880–5898.

[190] Wolfgang Klimesch, Michael Doppelmayr, Harald Russegger, Thomas Pachinger, and Jens Schwaiger. "Induced alpha band power changes in the human EEG and attention." In: *Neuroscience Letters* 244.2 (1998), pp. 73–76.

[191] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. "Similarity of Neural Network Representations Revisited." In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 3519–3529. URL: https://proceedings.mlr.press/v97/kornblith19a.html.

[192] J. P. Kulasingham, V. Vibujithan, and A. C. De Silva. "Deep belief networks and stacked autoencoders for the P300 Guilty Knowledge Test." In: *IECBES 2016 - IEEE-EMBS Conference on Biomedical Engineering and Sciences* (2016), pp. 127–132. DOI: 10.1109/IECBES.2016.7843428.

[193] Vitaliy Kurlin. "A fast and robust algorithm to count topologically persistent holes in noisy clouds." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1458–1463.

[194] O. Yeon Kwon, Min Ho Lee, Cuntai Guan, and Seong Whan Lee. "Subject-Independent Brain-Computer Interfaces Based on Deep Convolutional Neural Networks." In: *IEEE Transactions on Neural Networks and Learning Systems* 31.10 (Oct. 2020), pp. 3839–3852. ISSN: 21622388. DOI: 10.1109/TNNLS.2019.2946869.

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

[196] 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)*. Vol. 1. IEEE. 2000, pp. 424–429.

[197] Mario Lavanga, Laura Smets, Bieke Bollen, Katrien Jansen, Els Ortibus, Sabine Van Huffel, Gunnar Naulaers, and Alexander Caicedo. "A perinatal stress calculator for the neonatal intensive care unit: an unobtrusive approach." In: *Physiological Measurement* 41.7 (2020), p. 075012.

[198] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces." In: *Journal of Neural Engineering* 15.5 (2018), pp. 1–30. ISSN: 17412552. DOI: 10.1088/1741-2552/aace8c.

[199] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces." In: *Journal of Neural Engineering* 15.5 (2018), p. 056013.

[200] Hoanh Le. "Geometric invariance of PointNet." In: *Science and Engineering* (2021).

[201] Yann LeCun. *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/. 1998.

[202] Hyekyoung Lee, Moo K Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. "Discriminative persistent homology of brain networks." In: *2011 IEEE International Symposium on Biomedical Imaging: From nano to macro*. IEEE. 2011, pp. 841–844.

[203] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. "Weighted functional brain network modeling via network filtration." In: *NIPS Workshop on Algebraic Topology and Machine Learning*. Vol. 3. Citeseer. 2012.

[204] Yongjin Lee, Senja D Barthel, Paweł Dłotko, S Mohamad Moosavi, Kathryn Hess, and Berend Smit. "Quantifying similarity of pore-geometry in nanoporous materials." In: *Nature Communications* 8.1 (2017), pp. 1–8.

[205] 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*. Springer. 2014, pp. 40–51.

[206] J. Leygonie, S. Oudot, and U. Tillmann. "A Framework for Differential Calculus on Persistence Barcodes." In: *Foundations of Computational Mathematics* 22 (2022), 1069–1131. DOI: https://doi.org/10.1007/s10208-021-09522-y.

[207] Fali Li, Qin Tao, Wenjing Peng, Tao Zhang, Yajing Si, Yangsong Zhang, Chanlin Yi, Bharat Biswal, Dezhong Yao, and Peng Xu. "Inter-subject P300 variability relates to the efficiency of brain networks reconfigured from resting- to task-state: Evidence from a simultaneous event-related EEG-fMRI study." In: *NeuroImage* 205 (Jan. 2020), p. 116285. ISSN: 1053-8119. DOI: 10.1016/J.NEUROIMAGE.2019.116285.

[208] Feng Li, Yi Xia, Fei Wang, Dengyong Zhang, Xiaoyu Li, and Fan He. "Transfer Learning Algorithm of P300-EEG Signal Based on XDAWN Spatial Filter and Riemannian Geometry Classifier." In: *Applied Sciences 2020, Vol. 10, Page 1804* 10.5 (Mar. 2020), p. 1804. ISSN: 2076-3417. DOI: 10.3390/APP10051804. URL: https://www.mdpi.com/2076-3417/10/5/1804/htmhttps://www.mdpi.com/2076-3417/10/5/1804.

[209] Lu Li, Connor Thompson, Gregory Henselman-Petrusek, Chad Giusti, and Lori Ziegelmeier. "Minimal Cycle Representatives in Persistent Homology Using Linear Programming: An Empirical Study With User's Guide." In: *Frontiers in Artificial Intelligence* 4 (2021). DOI: 10.3389/frai.2021.681117. URL: https://www.frontiersin.org/articles/10.3389/frai.2021.681117.

[210] 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." In: *Plant Physiology* 177.4 (2018), pp. 1382–1395.

[211] 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*. IEEE. 2010, pp. 115–120.

[212] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. "Explainable AI: A Review of Machine Learning Interpretability Methods." In: *Entropy 2021, Vol. 23, Page 18* 23.1 (Dec. 2020), p. 18. ISSN: 1099-4300. DOI: 10.3390/E23010018. URL: https://www.mdpi.com/1099-4300/23/1/18/htmhttps://www.mdpi.com/1099-4300/23/1/18.

[213] Chang Liu, Xiaoyu Ma, Jiaojiao Wang, Jiefang Zhang, Honggang Zhang, Songyun Xie, and Dingguo Yu. "Neurophysiological Assessment of Image Quality from EEG Using Persistent Homology of Brain Network." In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2021, pp. 1–6.

[214] Ning-Han Liu, Cheng-Yu Chiang, and Hsuan-Chin Chu. "Recognizing the degree of human attention using EEG signals from mobile sensors." In: *Sensors* 13.8 (2013), pp. 10273–10286.

[215] Tian Liu, Yanni Chen, Pan Lin, and Jue Wang. "Small-world brain functional networks in children with attention-deficit/hyperactivity disorder revealed by EEG synchrony." In: *Clinical EEG and Neuroscience* 46.3 (2015), pp. 183–191.

[216] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J Király. "sktime: A unified interface for machine learning with time series." In: *arXiv preprint arXiv:1909.07872* (2019).

[217] Sandra K Loo and Russell A Barkley. "Clinical utility of EEG in attention deficit hyperactivity disorder." In: *Applied Neuropsychology* 12.2 (2005), pp. 64–76.

[218] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update." In: *Journal of Neural Engineering* 15.3 (2018), p. 031005.

[219] Joel F Lubar. "Discourse on the development of EEG diagnostics and biofeedback for attention-deficit/hyperactivity disorders." In: *Biofeedback and Self-Regulation* 16.3 (1991), pp. 201–225.

[220] Steven J. Luck, Geoffrey F. Woodman, and Edward K. Vogel. "Event-related potential studies of attention." In: *Trends in Cognitive Sciences* 4.11 (Nov. 2000), pp. 432–440. ISSN: 13646613. DOI: 10.1016/S1364-6613(00)01545-X. URL: http://www.cell.com/article/S136466130001545X/fulltexthttp://www.cell.com/article/S136466130001545X/abstracthttps://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(00)01545-X.

[221] Bo Qun Ma, He Li, Wei Long Zheng, and Bao Liang Lu. "Reducing the subject variability of eeg signals with adversarial domain generalization." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11953 LNCS (2019), pp. 30–42. ISSN: 16113349. DOI: 10.1007/978-3-030-36708-4{_}3/COVER. URL: https://link.springer.com/chapter/10.1007/978-3-030-36708-4_3.

[222] Shouvik Majumder, Fabio Apicella, Filippo Muratori, and Koel Das. "Detecting Autism Spectrum Disorder Using Topological Data Analysis." In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 1210–1214.

[223] J. N. Mak, Y. Arbel, J. W. Minett, L. M. McCane, B. Yuksel, D. Ryan, D. Thompson, L. Bianchi, and D. Erdogmus. "Optimizing the P300-based brain–computer interface: current status, limitations and future directions." In: *Journal of Neural Engineering* 8.2 (Mar. 2011), p. 025003. ISSN: 1741-2552. DOI: 10.1088/1741-2560/8/2/025003. URL: https://iopscience.iop.org/article/10.1088/1741-2560/8/2/025003https://iopscience.iop.org/article/10.1088/1741-2560/8/2/025003/meta.

[224] Joseph N. Mak and Jonathan R. Wolpaw. “Clinical Applications of Brain—Computer Interfaces: Current State and Future Prospects.” In: *IEEE Reviews in Biomedical Engineering* 2 (2009), pp. 187–199. ISSN: 19411189. DOI: 10.1109/RBME.2009.2035356.

[225] Usman Masud, Muhammad Iram Baig, Faraz Akram, and Tae Seong Kim. “A P300 brain computer interface based intelligent home control system using a random forest classifier.” In: *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings* 2018-January (Feb. 2018), pp. 1–5. DOI: 10.1109/SSCI.2017.8285449.

[226] Jérémie Mattout, Manu Maucle, and Wendy Kan. “BCI Challenge @NER 2015.” In: *https://kaggle.com/competitions/inria-bci-challenge* (2014).

[227] Sarah McGuire, Shane Jackson, Tegan Emerson, and Henry Kvinge. “Do neural networks trained with topological features learn different internal representations?” In: *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*. PMLR. 2023, pp. 122–136.

[228] Dan Meller and Nicolas Berkouk. “Singular Value Representation: A New Graph Perspective On Neural Networks.” In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 3353–3369.

[229] Zhengqing Miao, Meirong Zhao, Xin Zhang, and Dong Ming. “LMDA-Net:A lightweight multi-dimensional attention network for general EEG-based brain-computer interfaces and interpretability.” In: *NeuroImage* 276 (Aug. 2023), p. 120209. ISSN: 1053-8119. DOI: 10.1016/J.NEUROIMAGE.2023.120209.

[230] CM Michel, D Brandeis, W Skrandies, R Pascual, WK Strik, T Dierks, HL Hamburger, and W Karniski. “Global field power: A” time-honoured” index for EEG/EP map analysis.” In: (1993).

[231] Sifis Micheloyannis, Ellie Pachou, Cornelis Jan Stam, Michael Breakspear, Panagiotis Bitsios, Michael Vourkas, Sophia Erimaki, and Michael Zervakis. “Small-world networks and disturbed functional connectivity in schizophrenia.” In: *Schizophrenia Research* 87.1-3 (2006), pp. 60–66.

[232] Yuriy Mileyko, Sayan Mukherjee, and John Harer. “Probability measures on the space of persistence diagrams.” In: *Inverse Problems* 27.12 (2011), p. 124007.

[233] George A Miller. “The magical number seven, plus or minus two: Some limits on our capacity for processing information.” In: *Psychological Review* 63.2 (1956), p. 81.

[234] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. “The discrete geodesic problem.” In: *SIAM Journal on Computing* 16.4 (1987), pp. 647–668.

[235] Bart Moens, Leon van Noorden, and Marc Leman. “D-Jogger: Syncing music with walking.” In: *7th Sound and Music Computing Conference*. Universidad Pompeu Fabra. 2010, pp. 451–456.

[236] Bart Moens, Leon Van Noorden, Wim De Wilde, Micheline Lesaffre, Dirk Cambier, Dobri Dotov, Patrick Santens, Jana Blomme, Heleen Soens, and Marc Leman. "Effects of adaptive-tempo music-based RAS for Parkinson's disease patients." In: *ESCOM 2017*. 2017.

[237] Zainab Mohamed, Mohamed El Halaby, Tamer Said, Doaa Shawky, and Ashraf Badawi. "Characterizing focused attention and working memory using EEG." In: *Sensors* 18.11 (2018), p. 3743.

[238] Christoph Molnar, Giuseppe Casalicchio, and B. Bischl. "Interpretable Machine Learning - A Brief History, State-of-the-Art and Challenges." In: *PKDD/ECML Workshops*. 2020.

[239] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks." In: *Digital Signal Processing* 73 (2018), pp. 1–15. URL: `https://www.sciencedirect.com/science/article/pii/S1051200417302385`.

[240] Jair Montoya-Martínez, Alexander Bertrand, and Tom Francart. "Optimal number and placement of EEG electrodes for measurement of neural tracking of speech." In: *bioRxiv* (2019), p. 800979.

[241] Guido Montúfar, Nina Otter, and Yuguang Wang. "Can neural networks learn persistent homology features?" In: *arXiv preprint arXiv:2011.14688* (2020).

[242] Hiroshi Morioka, Atsunori Kanemura, Jun ichiro Hirayama, Manabu Shikauchi, Takeshi Ogawa, Shigeyuki Ikeda, Motoaki Kawanabe, and Shin Ishii. "Learning a common dictionary for subject-transfer decoding with resting calibration." In: *NeuroImage* 111 (May 2015), pp. 167–178. ISSN: 1053-8119. DOI: `10.1016/J.NEUROIMAGE.2015.02.015`.

[243] Dmitriy Morozov. "Persistence algorithm takes cubic time in worst case." In: *BioGeometry News, Dept. Comput. Sci., Duke Univ* 2 (2005).

[244] Steven Mortier, Jorg De Winne, Pengfei Sun, Wannes Vanransbeeck, Renata Turkes, Zhongju Yuan, Tim Verdonck, Marc Leman, Paul Devos, and Dick Botteldooren. *WithMe preprocessed dataset*. Oct. 2023. DOI: `10.6084/m9.figshare.24278887.v1`. URL: `https://figshare.com/articles/dataset/WithMe_preprocessed_dataset/24278887`.

[245] Steven Mortier, Renata Turkeš, Dick Botteldooren, Jorf De Winne, Steven Latré, Marc Leman, Wannes Van Ransbeeck, and Tim and Verdonck. "Classification of targets and distractors in an audiovisual attention task based on EEG." In: *Sensors* 28.31 (2023), p. 3743.

[246] Thomas Mulholland. "The concept of attention and the electroencephalographic alpha rhythm." In: *Attention in Neurophysiology* (1969), pp. 100–127.

[247] R Näätänen. "Selective attention and evoked potentials in humans — A critical review." In: *Biological Psychology* 2.4 (1975), pp. 237–307.

[248] R Näätänen, AW Gaillard, and S Mäntysalo. "Early selective-attention effect on evoked potential reinterpreted." In: *Acta psychologica* 42.4 (1978), pp. 313–329.

[249] Risto Näätänen. “The role of attention in auditory information processing as revealed by event-related potentials and other brain measures of cognitive function.” In: *Behavioral and brain sciences* 13.2 (1990), pp. 201–233.

[250] Alexander Nabutovsky and Shmuel Weinberger. “Algorithmic aspects of homeomorphism problems.” In: *Contemporary Mathematics* 231 (1999), pp. 245–250.

[251] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. “Topology of Deep Neural Networks.” In: *The Journal of Machine Learning Research* 21.184 (2020), pp. 1–40.

[252] Bradley J Nelson and Yuan Luo. “Topology-Preserving Dimensionality Reduction via Interleaving Optimization.” In: *arXiv preprint arXiv:2201.13012* (2022).

[253] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. “Brain Computer Interfaces, a Review.” In: *Sensors 2012, Vol. 12, Pages 1211-1279* 12.2 (Jan. 2012), pp. 1211–1279. ISSN: 1424-8220. DOI: 10.3390/S120201211. URL: https://www.mdpi.com/1424-8220/12/2/1211/htmhttps://www.mdpi.com/1424-8220/12/2/1211.

[254] Kamel Nidal and Aamir Saeed Malik. *EEG/ERP analysis: methods and applications*. Crc Press, 2014.

[255] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. “Finding the homology of submanifolds with high confidence from random samples.” In: *Discrete & Computational Geometry* 39.1-3 (2008), pp. 419–441.

[256] Yoji Okahara, Kouji Takano, Tetsuo Komori, Masahiro Nagao, Yasuo Iwadate, and Kenji Kansaku. “Operation of a P300-based brain-computer interface by patients with spinocerebellar ataxia.” In: *Clinical Neurophysiology Practice* 2 (Jan. 2017), pp. 147–153. ISSN: 2467-981X. DOI: 10.1016/J.CNP.2017.06.004.

[257] Nina Otter. “Magnitude meets persistence. Homology theories for filtered simplicial sets.” In: *arXiv preprint arXiv:1807.01540* (2018). to appear in *Homology, Homotopy and Applications*.

[258] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. “A roadmap for the computation of persistent homology.” In: *EPJ Data Science* 6.1 (2017), p. 17.

[259] Steve Oudot and Elchanan Solomon. “Inverse Problems in Topological Persistence.” In: *Topological Data Analysis*. Ed. by Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule. Cham: Springer International Publishing, 2020, pp. 405–433.

[260] Gaoxiang Ouyang, Xiaoli Li, Chuangyin Dang, and Douglas A Richards. “Using recurrence plot for determinism analysis of EEG recordings in genetic absence epilepsy rats.” In: *Clinical Neurophysiology* 119.8 (2008), pp. 1747–1755.

[261] Sourabh Palande, Vipin Jose, Brandon Zielinski, Jeffrey Anderson, P Thomas Fletcher, and Bei Wang. "Revisiting abnormalities in brain network architecture underlying autism using topology-inspired statistical inference." In: *International Workshop on Connectomics in Neuroimaging*. Springer. 2017, pp. 98–107.

[262] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library." In: *Advances in neural information processing systems* 32 (2019).

[263] Salil H Patel and Pierre N Azzam. "Characterization of N200 and P300: Selected Studies of the Event-Related Potential." In: *International Journal of Medical Sciences* 2.4 (2005), pp. 147–154. URL: `www.medsci.org`.

[264] Kaushik Patowary. *What mathematics has to do with the seven bridges of Königsberg*. `https://www.amusingplanet.com/2018/08/the-seven-bridges-of-konigsberg.html` Data of access: 29-01-2024. 2018.

[265] Rahul Paul and Stephan K Chalup. "A study on validating non-linear dimensionality reduction using persistent homology." In: *Pattern Recognition Letters* 100 (2017), pp. 160–166.

[266] Rahul Paul and Stephan Chalup. "Estimating Betti Numbers Using Deep Learning." In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–7.

[267] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[268] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830.

[269] 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." In: *BMC Bioinformatics* 16.1 (2015), p. 257.

[270] Jose A Perea and John Harer. "Sliding windows and persistence: An application of topological methods to signal analysis." In: *Foundations of Computational Mathematics* 15.3 (2015), pp. 799–838.

[271] Arnaldo E Pereira, Dereck Padden, Jay J Jantz, Kate Lin, and Ramses E Alcaide-Aguirre. "Cross-Subject EEG Event-Related Potential Classification for Brain-Computer Interfaces Using Residual Networks." In: *Preprint* (Sept. 2018). DOI: `10.13140/RG.2.2.16257.10086`. URL: `https://hal.science/hal-01878227https://hal.science/hal-01878227/document`.

[272] Giovanni Petri, Paul Expert, Federico Turkheimer, Robin Carhart-Harris, David Nutt, Peter J Hellyer, and Francesco Vaccarino. "Homological scaffolds of brain functional networks." In: *Journal of The Royal Society Interface* 11.101 (2014), p. 20140873.

[273] Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. "Topological strata of weighted complex networks." In: *PLOS ONE* 8.6 (2013), e66506.

[274] Gabriel Peyré, Marco Cuturi, et al. "Computational optimal transport: With applications to data science." In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[275] Terence W Picton and others. "The P300 wave of the human event-related potential." In: *Journal of clinical neurophysiology* 9 (1992), p. 456.

[276] John Polich. *Detection of change: event-related potential and fMRI findings*. Springer, 2003.

[277] John Polich. "Updating P300: An Integrative Theory of P3a and P3b." In: *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology* 118.10 (Oct. 2007), p. 2128. ISSN: 13882457. DOI: 10.1016/J.CLINPH.2007.04.019. URL: /pmc/articles/PMC2715154//pmc/articles/PMC2715154/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC2715154/.

[278] James R Pomerantz. "Wholes, holes, and basic features in vision." In: *Trends in Cognitive Sciences* 7.11 (2003), pp. 471–473.

[279] Pooja, S. K. Pahuja, and Karan Veer. "Recent Approaches on Classification and Feature Extraction of EEG Signal: A Review." In: *Robotica* 40.1 (Jan. 2022), pp. 77–101. ISSN: 0263-5747. DOI: 10.1017/S0263574721000382. URL: https://www.cambridge.org/core/journals/robotica/article/abs/recent-approaches-on-classification-and-feature-extraction-of-eeg-signal-a-review/52F9E04A6EE5A5872C5089874CE91DCF.

[280] Rolandos Alexandros Potamias, Alexandros Neofytou, Kyriaki Margarita Bintsi, and Stefanos Zafeiriou. "GraphWalks: Efficient Shape Agnostic Geodesic Shortest Path Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2968–2977.

[281] 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.

[282] 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*. 2017, pp. 652–660.

[283] Dmitriy A Rachkovskiy, Sergey V Slipchenko, Ivan S Misuno, Ernst M Kussul, and Tatyana N Baidyk. "Sparse binary distributed encoding of numeric vectors." In: *Journal of Automation and Information Sciences* 37.11 (2005).

[284] Abbas Rahimi, Simone Benatti, Pentti Kanerva, Luca Benini, and Jan M Rabaey. "Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition." In: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE. 2016, pp. 1–8.

[285] Abbas Rahimi, Sohum Datta, Denis Kleyko, Edward Paxon Frady, Bruno Olshausen, Pentti Kanerva, and Jan M Rabaey. "High-dimensional computing as a nanoscalable paradigm." In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.9 (2017), pp. 2508–2521.

[286] Abbas Rahimi, Pentti Kanerva, and Jan M Rabaey. "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing." In: *Proceedings of the 2016 international symposium on low power electronics and design*. 2016, pp. 64–69.

[287] Karthikeyan Natesan Ramamurthy, Kush Varshney, and Krishnan Mody. "Topological data analysis of decision boundaries with application to model selection." In: *International Conference on Machine Learning*. 2019, pp. 5351–5360.

[288] 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*. Springer. 2019, pp. 736–744.

[289] William J Ray and Harry W Cole. "EEG alpha activity reflects attentional demands, and beta activity reflects emotional and cognitive processes." In: *Science* 228.4700 (1985), pp. 750–752.

[290] Robert M.G. Reinhart, Daniel H. Mathalon, Brian J. Roach, and Judith M. Ford. "Relationships between pre-stimulus gamma power and subsequent P300 and reaction time breakdown in schizophrenia." In: *International Journal of Psychophysiology* 79.1 (Jan. 2011), pp. 16–24. ISSN: 0167-8760. DOI: 10.1016/J.IJPSYCHO.2010.08.009.

[291] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. "A stable multi-scale kernel for topological machine learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4741–4748.

[292] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. "Adversarial attacks and defenses in deep learning." In: *Engineering* 6.3 (2020), pp. 346–360.

[293] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August-2016 (Aug. 2016), pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[294] Angela Riccio, Luca Simione, Francesca Schettini, Alessia Pizzimenti, Maurizio Inghilleri, Marta Olivetti Belardinelli, Donatella Mattia, and Febo Cincotti. "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis." In: *Frontiers in Human Neuroscience* 7.NOV (Nov. 2013), p. 732. ISSN: 16625161. DOI: 10.3389/FNHUM.2013.00732/BIBTEX.

[295] Bastian Rieck. "On the Expressivity of Persistent Homology in Graph Learning." In: *arXiv preprint arXiv:2302.09826* (2023).

[296] Bastian Rieck and Heike Leitte. "Persistent homology for the evaluation of dimensionality reduction schemes." In: *Computer Graphics Forum*. Vol. 34. 3. Wiley Online Library. 2015, pp. 431–440.

[297] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. "Uncovering the topology of time-varying fMRI data using cubical persistence." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6900–6912.

[298] Bertrand Rivet, Antoine Souloumiac, Virginie Attina, and Guillaume Gibert. "xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain-Computer Interface." In: *IEEE Transactions on Biomedical Engineering* 56.8 (2009), pp. 2035–2043. DOI: 10.1109/TBME.2009.2012869.

[299] Bertrand Rivet, Antoine Souloumiac, Virginie Attina, and Guillaume Gibert. "xDAWN algorithm to enhance evoked potentials: application to brain–computer interface." In: *IEEE Transactions on Biomedical Engineering* 56.8 (2009), pp. 2035–2043.

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

[301] Christian Ronse. "A bibliography on digital and computational convexity (1961–1988)." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.2 (1989), pp. 181–190.

[302] David Rouse, Adam Watkins, David Porter, John Harer, Paul Bendich, Nate Strawn, Elizabeth Munch, Jonathan DeSena, Jesse Clarke, Jeff Gilbert, et al. "Feature-aided multiple hypothesis tracking using topological and statistical behavior classifiers." In: *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIV*. Vol. 9474. International Society for Optics and Photonics. 2015, p. 94740L.

[303] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. "Deep learning-based electroencephalography analysis: a systematic review." In: *Journal of neural engineering* 16.5 (2019), p. 051001.

[304] Martin Royer, Frédéric Chazal, Clément Levrard, Yuichi Ike, and Yuhei Umeda. "ATOL: Measure Vectorisation for Automatic Topologically-Oriented Learning." In: *arXiv preprint arXiv:1909.13472* (2019).

[305] Mikail Rubinov and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations." In: *NeuroImage* 52.3 (2010), pp. 1059–1069.

[306] Andrea Saltelli. "Sensitivity Analysis for Importance Assessment." In: *Risk Analysis* 22.3 (2002), pp. 579–590. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/0272-4332.00040.

[307] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications." In: *Proceedings of the IEEE* 109 (2021), pp. 247–278.

[308] Paul Sauseng, Wolfgang Klimesch, Waltraud Stadler, Manuel Schabus, Michael Doppelmayr, Simon Hanslmayr, Walter R Gruber, and Niels Birbaumer. "A shift of visual spatial attention is selectively associated with human EEG alpha activity." In: *European Journal of Neuroscience* 22.11 (2005), pp. 2917–2926.

[309] Christian Scharinger, Alexander Soutschek, Torsten Schubert, and Peter Gerjets. "Comparison of the working memory load in N-back and working memory span tasks by means of EEG frequency band power and P300 amplitude." In: *Frontiers in Human Neuroscience* 11 (Jan. 2017), p. 212526. ISSN: 16625161. DOI: 10.3389/FNHUM.2017.00006/BIBTEX.

[310] Gunnar Schmidtmann, Ben J Jennings, and Frederick AA Kingdom. "Shape recognition: convexities, concavities and things in between." In: *Scientific Reports* 5.1 (2015), pp. 1–11.

[311] Benjamin Schweinhart. "Fractal dimension and the persistent homology of random geometric complexes." In: *Advances in Mathematics* 372 (2020), p. 107291.

[312] Lee M Seversky, Shelby Davis, and Matthew Berger. "On time-series topological data analysis: New data and opportunities." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 59–67.

[313] Foroogh Shamsi, Ali Haddad, Laleh Najafizadeh, Boyu Zang, Yanfei Lin, Zhiwen Liu, al, Bashar Awwad Shiekh Hasan, and John Q Gan. "Deep learning for electroencephalogram (EEG) classification tasks: a review." In: *Journal of Neural Engineering* 16.3 (Apr. 2019), p. 031001. ISSN: 1741-2552. DOI: 10.1088/1741-2552/AB0AB5. URL: https://iopscience.iop.org/article/10.1088/1741-2552/ab0ab5https://iopscience.iop.org/article/10.1088/1741-2552/ab0ab5/meta.

[314] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features Through Propagating Activation Differences*. July 2017. URL: https://proceedings.mlr.press/v70/shrikumar17a.html.

[315] Thomas Sikora. "The MPEG-7 visual standard for content description–An overview." In: *IEEE Transactions on Circuits and Systems for Video Technology* 11.6 (2001), pp. 696–702.

[316] Fernando Lopes da Silva. "EEG and MEG: Relevance to neuroscience." In: *Neuron* 80.5 (2013), pp. 1112–1128.

[317] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." In: *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings* (Dec. 2013). URL: `https://arxiv.org/abs/1312.6034v2`.

[318] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." In: *CoRR* abs/1312.6034 (2013). URL: `http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SimonyanVZ13`.

[319] 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*. Springer. 2014, pp. 231–239.

[320] Primoz Skraba and Katharine Turner. "Wasserstein Stability for Persistence Diagrams." In: *arXiv preprint arXiv:2006.16824* (2020).

[321] Wolfgang Skrandies. "Global field power and topographic similarity." In: *Brain Topography* 3 (1990), pp. 137–141.

[322] Laura Smets, Werner Van Leekwijck, Ing Jyh Tsang, and Steven Latré. "An Encoding Framework for Binarized Images using HyperDimensional Computing." In: *arXiv preprint arXiv:2312.00454* (2023).

[323] Laura Smets, Werner Van Leekwijck, Ing Jyh Tsang, and Steven Latre. "Training a HyperDimensional Computing Classifier using a Threshold on its Confidence." In: *arXiv preprint arXiv:2305.19007* (2023).

[324] EJ Speckmann, CE Elger, and A Gorji. "Neurophysiologic basis of EEG and DC potentials." In: *Niedermeyer's Electroencephalography* (2011), pp. 17–32.

[325] Gard Spreemann, Benjamin Dunn, Magnus Bakke Botnan, and Nils A Baas. "Using persistent homology to reveal hidden information in neural data." In: *arXiv preprint arXiv:1510.06629* (2015).

[326] Cornelis J Stam, BF Jones, G Nolte, M Breakspear, and Ph Scheltens. "Small-world networks and functional connectivity in Alzheimer's disease." In: *Cerebral Cortex* 17.1 (2007), pp. 92–99.

[327] Bernadette J Stolz, Tegan Emerson, Satu Nahkuri, Mason A Porter, and Heather A Harrington. "Topological data analysis of task-based fMRI data from experiments on Schizophrenia." In: *arXiv preprint arXiv:1809.08504* (2018).

[328] Bernadette J Stolz, Heather A Harrington, and Mason A Porter. "Persistent homology of time-dependent functional networks constructed from coupled time series." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.4 (2017), p. 047410.

[329] Bernadette Stolz. "Computational topology in neuroscience." In: *Master's thesis (University of Oxford, 2014)* (2014).

[330] Thibault de Surrel, Felix Hensel, Mathieu Carrière, Théo Lacombe, Yuichi Ike, Hiroaki Kurihara, Marc Glisse, and Frédéric Chazal. "RipsNet: a general architecture for fast and robust estimation of the persistent homology of point clouds." In: *arXiv preprint arXiv:2202.01725* (2022).

[331] The GUDHI Project. *GUDHI User and Reference Manual*. 3.3.0. GUDHI Editorial Board, 2020. URL: `https://gudhi.inria.fr/doc/3.3.0/`.

[332] The RIVET Developers. *RIVET*. Version 1.1.0. 2020. URL: `https://github.com/rivetTDA/rivet/`.

[333] Anthony Thomas, Sanjoy Dasgupta, and Tajana Rosing. "Theoretical foundations of hyperdimensional computing." In: *arXiv preprint arXiv:2010.07426* (2020).

[334] Brigitta Tóth, Dávid Farkas, Gábor Urbán, Orsolya Szalárdy, Gábor Orosz, László Hunyadi, Botond Hajdu, Annamária Kovács, Beáta Tünde Szabó, Lidia B Shestopalova, et al. "Attention and speech-processing related functional brain networks activated in a multi-speaker environment." In: *PLOS ONE* 14.2 (2019), e0212754.

[335] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. "Ripser.py: A Lean Persistent Homology Library for Python." In: *The Journal of Open Source Software* 3.29 (2018), p. 925. DOI: `10.21105/joss.00925`. URL: `https://doi.org/10.21105/joss.00925`.

[336] Loring W Tu. "Manifolds." In: *An Introduction to Manifolds*. Springer, 2011, pp. 47–83.

[337] Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré. "Noise robustness of persistent homology on greyscale images, across filtrations and signatures." In: *PLOS ONE* 16.9 (2021), e0257215.

[338] Renata Turkeš, Guido Montúfar, and Nina Otter. "On the Effectiveness of Persistent Homology." In: *Advances in Neural Information Processing Systems*. 2022. URL: `https://openreview.net/forum?id=DRjUkfExCix`.

[339] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. "Persistent homology transform for modeling shapes and surfaces." In: *Information and Inference: A Journal of the IMA* 3.4 (2014), pp. 310–344.

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

[341] Yuhei Umeda. "Time series classification via topological data analysis." In: *Information and Media Technologies* 12 (2017), pp. 228–239.

[342] Edith Van Dyck, Bart Moens, Jeska Buhmann, Michiel Demey, Esther Coorevits, Simone Dalla Bella, and Marc Leman. "Spontaneous entrainment of running cadence to music tempo." In: *Sports Medicine Open* 1.1 (2015), pp. 1–14.

[343] Raymond Van Ee, Jeroen JA Van Boxtel, Amanda L Parker, and David Alais. "Multisensory congruency as a mechanism for attentional control over perceptual selection." In: *Journal of Neuroscience* 29.37 (2009), pp. 11641–11649.

[344] Guido Van Rossum and Fred L Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[345] Robin Vandaele. "Topological inference in graphs and images." PhD thesis. Ghent University, 2020.

[346] Kush R Varshney and Karthikeyan Natesan Ramamurthy. "Persistent topology of decision boundaries." In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 3931–3935.

[347] Fabrizio de Vico Fallani, Laura Astolfi, Febo Cincotti, Donatella Mattia, Daria La Rocca, Elira Maksuti, Serenella Salinari, Fabio Babiloni, Balazs Vegso, Gyorgy Kozmann, et al. "Evaluation of the brain network organization from EEG signals: A preliminary evidence in stroke patient." In: *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology: Advances in Integrative Anatomy and Evolutionary Biology* 292.12 (2009), pp. 2023–2031.

[348] Leopold Vietoris. "Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen." In: *Mathematische Annalen* 97.1 (1927), pp. 454–472.

[349] Oliver Vipond. "Multiparameter Persistence Landscapes." In: *The Journal of Machine Learning Research* 21.61 (2020), pp. 1–38.

[350] 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." In: *Nature Methods* 17.3 (2020), pp. 261–272.

[351] Hubert Wagner, Chao Chen, and Erald Vuçini. "Efficient computation of persistent homology for cubical data." In: *Topological methods in data analysis and visualization II*. Springer, 2012, pp. 91–106.

[352] Zhijiang Wan, Jian He, and Alicia Voisine. "An attention level monitoring and alarming system for the driver fatigue in the pervasive environment." In: *Brain and Health Informatics: International Conference, BHI 2013, Maebashi, Japan, October 29-31, 2013. Proceedings*. Springer. 2013, pp. 287–296.

[353] Menglun Wang, Zixuan Cang, and Guo-Wei Wei. "A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation." In: *Nature Machine Intelligence* 2.2 (2020), pp. 116–123.

[354] Yan Wang, Yining Zhao, Shihui Ying, Shaoyi Du, and Yue Gao. "Rotation-Invariant Point Cloud Representation for 3-D Model Recognition." In: *IEEE Transactions on Cybernetics* (2022).

[355] Yuan Wang, Hernando Ombao, and Moo K Chung. "Topological seizure origin detection in electroencephalographic signals." In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2015, pp. 351–354.

[356] Yuan Wang, Hernando Ombao, and Moo K Chung. "Statistical persistent homology of brain signals." In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 1125–1129.

[357] Matthew Wheeler, Jose Bouza, and Peter Bubenik. "Activation Landscapes as a Topological Summary of Neural Network Performance." In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 3865–3870.

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

[359] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis." In: *Chemometrics and Intelligent Laboratory Systems* 2.1-3 (Aug. 1987), pp. 37–52. ISSN: 0169-7439. DOI: 10.1016/0169-7439(87)80084-9.

[360] Chi-Chong Wong and Chi-Man Vong. "Persistent homology based graph convolution network for fine-grained 3D shape segmentation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7098–7107.

[361] Geoffrey F. Woodman. "A brief introduction to the use of event-related potentials in studies of perception and attention." In: *Attention, Perception, & Psychophysics 2010 72:8* 72.8 (2010), pp. 2031–2046. ISSN: 1943-393X. DOI: 10.3758/BF03196680. URL: https://link.springer.com/article/10.3758/BF03196680.

[362] Dongrui Wu. "Online and Offline Domain Adaptation for Reducing BCI Calibration Effort." In: *IEEE Transactions on Human-Machine Systems* 47.4 (Aug. 2017), pp. 550–563. ISSN: 21682291. DOI: 10.1109/THMS.2016.2608931.

[363] Dongrui Wu, Yifan Xu, and Bao-Liang Lu. "Transfer learning for EEG-based brain–computer interfaces: A review of progress made since 2016." In: *IEEE Transactions on Cognitive and Developmental Systems* 14.1 (2020), pp. 4–19.

[364] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. "A leaf recognition algorithm for plant classification using probabilistic neural network." In: *2007 IEEE International Symposium on Signal Processing and Information Technology*. IEEE. 2007, pp. 11–16.

[365] Peter Xenopoulos, Gromit Chan, Harish Doraiswamy, Luis Gustavo Nonato, Brian Barr, and Claudio Silva. "Topological representations of local explanations." In: *arXiv preprint arXiv:2201.02155* (2022).

[366] 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*. 2021, pp. 826–835.

[367] Xiaoqi Xu, Nicolas Drougard, and Raphaëlle N Roy. "Topological Data Analysis as a New Tool for EEG Processing." In: *Frontiers in Neuroscience* 15 (2021), p. 761703.

[368] Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. "Link prediction with persistent homology: An interactive view." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11659–11669.

[369] Jae Jun Yoo, Jae Seung Chang, Moo Chung, and Jong Chul Ye. "Topological Analysis of EEG Connectivity Patterns of Depressed Patients using Persistence Landscape." In: *Organization for Human Brain Mapping, OHBM*. Organization for Human Brain Mapping, OHBM. 2014.

[370] Jaejun Yoo, Eun Young Kim, Yong Min Ahn, and Jong Chul Ye. "Topological persistence vineyard for dynamic functional brain connectivity during resting and gaming stages." In: *Journal of Neuroscience Methods* 267 (2016), pp. 1–13.

[371] Meichen Yu, Alida A Gouw, Arjan Hillebrand, Betty M Tijms, Cornelis Jan Stam, Elisabeth CW van Straaten, and Yolande AL Pijnenburg. "Different functional connectivity and network topology in behavioral variant of frontotemporal dementia and Alzheimer's disease: an EEG study." In: *Neurobiology of Aging* 42 (2016), pp. 150–162.

[372] Alberto Zancanaro, Giulia Cisotto, João Ruivo Paulo, Gabriel Pires, and Urbano J. Nunes. "CNN-based Approaches For Cross-Subject Classification in Motor Imagery: From the state-of-the-art to DynamicNet." In: *2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2021* (2021). DOI: 10.1109/CIBCB49929.2021.9562821.

[373] Matthias Zeppelzauer, Bartosz Zieliński, Mateusz Juda, and Markus Seidl. "Topological descriptors for 3D surface analysis." In: *International Workshop on Computational Topology in Image Context*. Springer. 2016, pp. 77–87.

[374] Hongfei Zhang, Zehui Wang, Yinhu Yu, Haojun Yin, Chuangquan Chen, and Hongtao Wang. "An improved EEGNet for single-trial EEG classification in rapid serial visual presentation task." In: *Brain Science Advances* 2022.2 (2022), pp. 111–126. ISSN: 2096-5958. DOI: 10.26599/BSA.2022.9050007. URL: https://doi.org/10.26599/BSA.2022.9050007.

[375] 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)*. IEEE. 2020, pp. 200–209.

[376] 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)*. IEEE. 2019, pp. 204–213.

[377] Ziwei Zhang, Xin Wang, Zeyang Zhang, Peng Cui, and Wenwu Zhu. "Revisiting Transformation Invariant Geometric Deep Learning: Are Initial Representations All You Need?" In: *arXiv preprint arXiv:2112.12345* (2021).

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

[379] Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang. "Persistence enhanced graph neural network." In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2896–2906.

[380] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. "Point-cloud saliency maps." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1598–1606.

[381] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing." In: *arXiv:1801.09847* (2018).

[382] Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y Ng, Gunnar Carlsson, and Stefano Ermon. "Evaluating the disentanglement of deep generative models through manifold topology." In: *arXiv preprint arXiv:2006.03680* (2020).

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

[384] Afra Zomorodian and Gunnar Carlsson. "Computing persistent homology." In: *Discrete & Computational Geometry* 33.2 (2005), pp. 249–274.

[385] Jovisa Zunic and Paul L Rosin. "A new convexity measure for polygons." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.7 (2004), pp. 923–934.

Notation

Tables D.7-D.9 summarize the notation respectively for the input data, filtration and persistence signatures. For the purpose of clarity, throughout the manuscript we aim for the following typographic convention:

uppercase	set, e.g., S, X, Z, K
lowercase	scalar, e.g., n, m, k, r
boldface lowercase	vector, e.g., $\mathbf{x}, \mathbf{y}$
Greek alphabet lowercase	function, e.g., $\phi, \psi, \tau, \lambda, \gamma, \omega$

with the exception of persistent homology encoding f which sends the input data to a persistence signature, and standard notation in the literature, such as d for distance function and W_p, l_p and L_p for Wasserstein and Euclidean distances, N for number of simplices, or C for the SVM regularization parameter.

Notation	Interpretation
S	topological space
$[N]$	set $\{1, 2, \ldots, N\}$
X	point cloud, $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^D$
x	point cloud point, $x \in X$
n	number of point cloud points, $n = \|X\|$
D	point cloud dimension
Z	image as a two-dimensional matrix, $Z = [z_{uv}]$
z_{uv}	greyscale value of pixel (u, v)
z_0	threshold greyscale value (to obtain binary image)
n	number of pixels
n_x	number of pixels in x direction
n_y	number of pixels in y direction
$\mathcal{X}$	data set, $\mathcal{X} = \{X_1, X_2, \ldots, X_m\}$
m	number of data samples or observations, $m = \|\mathcal{X}\|$
$\\| \cdot \\|_p$	Euclidean distance between vectors
$\omega : X \to Y$	bijection between X and Y
$\Omega(X, Y)$	set of bijections between X and Y
W_p	Wasserstein distance between point clouds or images
$\pi : \mathcal{X} \to \mathcal{X}$	perturbation
$\rho : \mathcal{X} \to \mathbb{R}$	feature
$T_X\mathcal{X}$	tangent space at X on $\mathcal{X}$
$V : \mathcal{X} \to \sqcup_X T_X\mathcal{X}$	vector field
$v = V(X)$	tangent vector at X assigned by V
V_π	vector field induced by perturbation π
$\nabla\rho$	gradient vector field of function ρ

Table D.7.: Notation about the input data (space).

Notation	Interpretation
σ	simplex (e.g., point, edge, triangle)
K	simplicial or cubical complex
r	filtration scale, resolution or time, $r \in \mathbb{R}$
K_r	simplicial or cubical complex describing the data at scale $r \in \mathbb{R}$
$VR(X, r)$	Vietoris-Rips simplicial complex with resolution r
$\{K_r\}_{r\in\mathbb{R}}$	filtration
ϕ or $\psi : K \to \mathbb{R}$	filtration function
$\delta_X : \mathbb{R}^D \to \mathbb{R}$	distance (filtration) function, i.e., distance from X
$\delta_{X,m} : \mathbb{R}^D \to \mathbb{R}$	distance-to-a-measure (DTM) (filtration) function with parameter m (reflecting the number of nearest neighbors)
$\phi_{z_0} : K \to \mathbb{R}$	binary filtration function with parameter z_0
$\phi_z : K \to \mathbb{R}$	greyscale filtration function
$\phi_{d_0,z_0} : K \to \mathbb{R}$	density filtration function with parameters d_0 and z_0
$\phi_{(u_0,v_0),z_0} : K \to \mathbb{R}$	radial filtration function with parameters (u_0, v_0) and z_0

Table D.8.: Notation about the filtration, input for persistent homology.

Notation	Interpretation
PH	persistent homology, multi-set of persistence intervals
(b_i, d_i)	persistence interval, i.e., pair of birth and death values for cycle i
$l_i = d_i - b_i$	lifespan or persistence of a cycle i
k	homological dimension
β_k	Betti number, i.e., the number of idependent k-dimensional cycles
PD	persistence diagram
W_p	Wasserstein distance (referred to as bottleneck distance for $p = +\infty$)
PI	persistence image
$\gamma_{(x_0,y_0),\sigma^2} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$	Guassian kernel centered around (x_0, y_0) and with variance σ^2
$\alpha : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$	weight function
$\eta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$	persistence surface
p	PI resolution, number of PI pixels in x and y direction
PL	vectorized persistence landscape
$\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$	persistence landscape function
$\lambda_i : \mathbb{R} \to \mathbb{R}$	persistence landscape function for the i-th most persistent cycle, $\lambda_i(r) = \lambda(i, r)$
$\mathcal{P}$	data set of persistence signatures
$f : \mathcal{X} \to \mathcal{P}$	PH map or encoding
J_X^f, J^f, J_X, J	Jacobian mapping (matrix) of map f at X
G_X^f, G^f, G_X, G	Gram matrix of map f at X
q_i^f, q_i	i-th eigenvector of map f
λ_i^f, λ_i	i-th largest singular value of the Jacobian mapping
$\|\cdot\|_f$	pull-back norm induced by map f
BW	Bures-Wasserstein distance between positive-definite matrices

Table D.9.: Notation about the signature, output for persistent homology.