Department of Mathematics
TUM School of Computation, Information and Technology
Technical University of Munich

Topological Data Analysis on Multimodal Brain Data

Classification into Anesthesia and Sleep Stages with Persistent Homology

Pia Baronetzky

Thesis for the attainment of the academic degree

Master of Science

at the TUM School of Computation, Information and Technology of the Technical University of Munich

Supervisor:
Bastian Rieck

Advisors:
Bastian Rieck
Silviu Bodea

Submitted:
Munich, 30. September 2024

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

P.Baronetzky

Munich, 30. September 2024 Pia Baronetzky

Zusammenfassung

Diese Masterarbeit untersucht die Anwendung der Topologischen Datenanalyse (TDA) bei der Klassifizierung multimodaler Mausgehirndaten in Narkose- und Schlafstadien. Wir verwenden Persistent Homology, ein zentrales Werkzeug der TDA, um topologische Merkmale in den Daten zu erkennen, die sich aus Elektroenzephalographie (EEG), Elektromyographie (EMG) und Gehirnbildgebungsdaten zusammensetzen.

Nach der Einführung in die mathematischen Grundlagen der TDA untersuchen wir die Daten explorativ. Dabei berechnen wir Persistence Diagrams aus den Daten, welche die topologischen Merkmale zusammenfassen, sowie Functional Summaries, alternative Darstellungen zu den gängigen Persistence Diagrams.

Bei der Datenexploration finden wir nur eine geringe Korrelation zwischen der Komplexität der Zeitreihen und der topologischen Komplexität, beobachteten jedoch, wie sich die Daten topologisch zwischen verschiedenen Narkose- und Schlafstadien unterscheiden.

Abschließend verwenden wir aus den Persistence Diagrams und Functional Summaries berechnete Statistiken als Features für zwei Machine-Learning-Classifier und vergleichen die Signifikanz der Features. Eine interessante Erkenntnis ist, dass die Adcock-Carlsson-Koordinaten, ein instabiles Functional Summary, vermutlich das wichtigste Merkmal für die Klassifizierung sind.

Die in dieser Masterarbeit verwendeten maschinellen Machine-Learning-Classifier erzielen mit Persistent Homology Features eine deutlich bessere Accuracy als mit traditionellen statistischen Features.

Abstract

This master thesis explores the application of Topological Data Analysis (TDA) in the classification of multimodal mouse brain data into anesthesia and sleep stages. We use persistent homology, a key tool in TDA, to detect topological features in the data, which consists of Electroencephalography (EEG), Electromyography (EMG), and brain imaging datasets.

After introducing the mathematical foundations of TDA, we explored the topological characteristics of the data. This involves converting it to persistence diagrams, which summarize the topological features in data, and functional summaries, which are alternative depictions to the common persistence diagrams.

In the data exploration, we only find a slight correlation between time series complexity and topological complexity, but observe how the data differs between different anesthesia and sleep stages topologically.

Finally, we use statistics obtained from the persistence diagrams and functional summaries as features for two machine learning classifiers, and compare the importance of the features. It is an interesting finding that arguably the most important feature in the classification is Adcock-Carlsson coordinates, which is an unstable summary statistic that can extract insight from small-scale topological features particularly well.

The machine learning classifiers used in this master thesis perform significantly better with features obtained by persistent homology than with traditional statistical features, suggesting that TDA can provide valuable insights into the complexities of brain activity and consciousness.

Contents

1 Introduction 1

2 Topological Foundations 4
2.1 Mathematical Perspectives on Shapes 4
2.2 Topological spaces 5
2.3 Homology Theory 7
2.4 Persistent Homology - Finding Holes in Real-Word Data 16
2.5 Stability 22

3 Functional Summaries 25
3.1 Persistence Landscape 25
3.2 Silhouette 26
3.3 Betti Curve 27
3.4 Heatkernel 28
3.5 Persistence Images 29
3.6 Amplitude 30
3.7 Number of Points 31
3.8 Persistence of the Most Prominent Point 31
3.9 Persistence Entropy 31
3.10 Complex Polynomials 32
3.11 ATOL vectorization 32
3.12 Adcock-Carlsson Coordinates 33

4 Data Exploration 34
4.1 Data 34
4.2 Visualizing Persistence Diagrams 37
4.3 Visualizing Basic Summary Statistics 38
4.4 Visualizing Signatures 39
4.5 Searching Clusters in Functional Summaries 42
4.6 Quantifying the Correlation Between Topological Complexity and Time Series Complexity 44
4.7 Quantifying the Correlation Between the Data Modalities 45

5 Methods for Classification 47
5.1 Preprocessing 47
5.2 Creating EEG/EMG Data Persistence Diagrams 47
5.3 Creating Brain Imaging Data Persistence Diagrams 50
5.4 Noise in Persistence Diagrams 50
5.5 Data Fusion of Multimodal Brain Data 50
5.6 Classification of Multimodal Brain Data into Anesthesia and Sleep Stages 52

6 Results 58
6.1 Classification Results 58
6.2 Feature Importance 59

7 Discussion and Outlook 62

A Appendix **64**
References 67

Bibliography **71**

1 Introduction

Analyzing the brain through the lens of data science promises unprecedented insights with profound real-world applications. Data has helped us understand “brain and behavior” and develop treatments and innovative technologies already today. Researchers have, for instance, invented artificial voices for mute people by analyzing their brain recordings [Doc21], which can potentially impact millions of lives in the future. In a time where data sharing and open access are becoming increasingly important, the role of data in neuroscience will only continue to grow. We just need the appropriate methods to analyze brain data and find patterns the human eye cannot detect.

In recent years, machine learning models have gained popularity to handle large datasets. In machine learning, often a crucial step is extracting relevant data features to feed into the model. Such features can, for instance, be statistical features like the mean or skewness of a time series. However, using statistical features for machine learning can perform worse than using features from one recent data analysis approach, Topological Data Analysis (TDA), particularly on complex brain data [Bil+21]. TDA offers an alternative perspective to traditional data analysis and helps create features that capture *essential* information behind data.

The core principle behind Topological Data Analysis is the idea that behind all real-world data, there is an underlying “intrinsic shape.” The core dogma of TDA is often cited as “data has shape, shape has meaning, and meaning drives value” [EYCO23]. The intrinsic shape behind data is often directly visible when the data is a 3D point cloud. The point cloud in Figure 1.1, for instance, shows a donut as its underlying intrinsic shape. However, also image data and time series data (like EEG data) harbours intrinsic shapes, as we will see in the course of this master thesis.

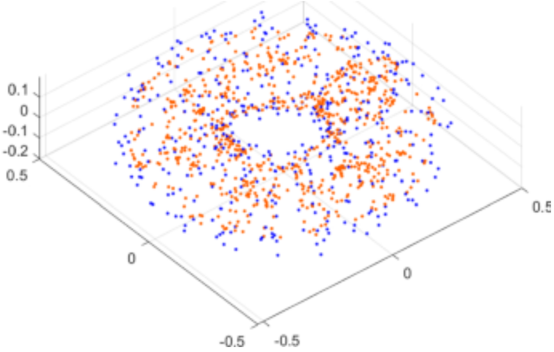

Figure 1.1 Data points showing a donut as an “intrinsic shape” [YM21].

TDA combines the century-old studies of topology and geometry to extract insights about the intrinsic shapes behind today’s real-world data. Topology and geometry complement each other in examining shapes. While geometry offers a quantitative perspective on shapes, focusing on measurements such as distances, topology offers a qualitative perspective that focuses on the overall characteristics of shapes. In topology, length measurements are not of interest, and shapes that we can continuously deform into each other, such as through stretching and bending, are considered to be the same, regardless of their length measurements. For example, as we can continuously transform a donut into a coffee mug and back (as shown in Figure 1.2), they are topologically equivalent.

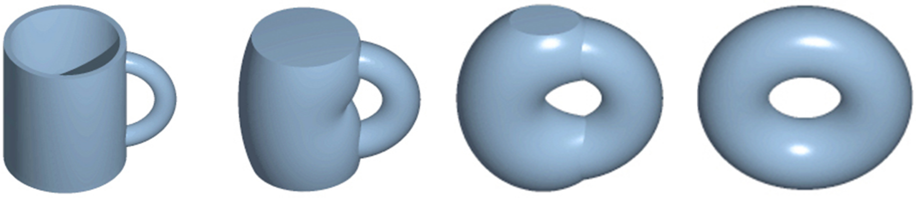

Figure 1.2 Coffee mug transformed to a donut [Sch18].

Topological classification is challenging and still an active area of research. Doing further research could, however, be valuable in real-world applications: In [CZS03], the authors find that even honeybees do well at discriminating non-equivalent shapes and poorly at discriminating topologically equivalent shapes. The concept of "topological equivalence" seems quite relevant in capturing the nature of a shape we perceive.

For topological classification, topologists examine topological *invariants*, those properties of shapes that stay invariant under continuous deformations. Equivalent shapes, like a donut and a coffee mug, share their invariants. We can, therefore, learn about the topological identity of shapes by looking for their topological invariants.

One example of a topological invariant is the number of *holes* of different dimensions. A donut and a coffee mug have the same number of holes, with the most prominent hole visible in the center of the donut and the handle of the coffee mug representing just one of their holes. There are, in fact, *five* holes in coffee mugs or donuts, and we will learn later why.

Persistent homology, a fundamental tool of TDA, enables us to identify and analyze holes in real-world data. It is a method that counts topological holes of all dimensions and evaluates their significance using geometric tools, allowing us to distinguish "persistent" large-scale holes from noise (See Figure 1.3).

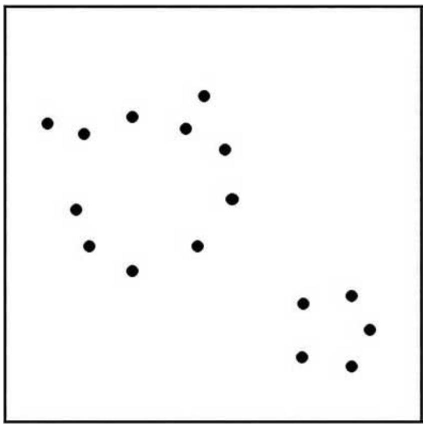

Figure 1.3 Data points looking like circles each enclosing a tunnel (a type of hole). The tunnel in between the left upper data points is larger and more persistent than the tunnel in between the right lower data points [LR21].

The data we will apply persistent homology to in this master thesis is mouse brain data. The mice were anesthetized in one experiment and observed regarding their sleep behavior in another experiment. We want to classify the data into anesthesia and sleep stages, respectively, by finding topological holes in the data and feeding properties of these holes into a machine learning model. While we are using mouse brain data for this master thesis, similar methods could be applied to human brain data.

The data is multimodal, combining simultaneous Electroencephalography (EEG), Electromyography (EMG), and brain imaging datasets. These data modalities have inherently different strengths and shortcomings; EEG and EMG data have a high temporal resolution but no spatial information, while brain imaging data have a much lower temporal resolution but show spatial information. Therefore, we want to combine the different data modalities to benefit from all of their strengths.

The goal of this master thesis will be to explore the effectiveness of TDA in classifying brain states by comparing the performance of topological machine learning features with more traditional statistical ones. Addionally, we want to compare the performance of different functional summaries by assessing their feature importances.

The data recorded on the anesthetized mice will be the focus of this master thesis. Observing the depth of anesthesia with modern data analysis methods can be helpful, for example, for monitoring medical operations.

Given that TDA has already had promising applications in medicine, such as in detecting breast cancer [Nic+11], we aim to provide helpful insights into the stages of consciousness by analyzing the mouse brain data with TDA.

2 Topological Foundations

2.1 Mathematical Perspectives on Shapes

Observing shapes has fueled our understanding of the world. We have the remarkable skill to recognize patterns in shapes, which allows us to make sense of the things we see. However, instead of relying on our intuition when observing shapes, we can also use mathematical models to study them, which has proven invaluable throughout history.

"[The universe] cannot be read until we have learnt the language and become familiar with the characters in which it is written. It is written in mathematical language, and the letters are triangles, circles and other geometrical figures, without which means it is humanly impossible to comprehend a single word." [Galilei] [LL06]

Geometricians like Galilei see shapes, such as circles, squares, and triangles, as the building blocks of the universe. Each shape has specific characteristics, and the ensemble of shapes forms our universe.

Geometry dates back millennia [Ede14], and is concerned with measuring shapes by length, angles, areas, curvature, and more. It focuses on the quantitative aspects of shapes and their *local* characteristics [Am20], meaning characteristics that we can observe within a small shape region without considering the overall shape. *Fourier analysis*, for instance, is a geometric tool useful to capture the outline of a shape [Am20].

Topology, on the other hand, dates back only a few centuries [CVJ21] and examines qualitative, *global* properties of shapes [Am20], so the overall structure and relationships between different parts of a shape. An example of a property reflecting the overall shape structure is the number of *holes* and the *connectivity* of the shape.

The observation of connectivity also marked the beginning of topology; in the 18th century, Leonhard Euler introduced a form of the notable Königsberg bridge problem, depicted in Figure 2.1, and found that the essential information in the problem was the connectivity of the islands, and not their exact positions [CVJ21].

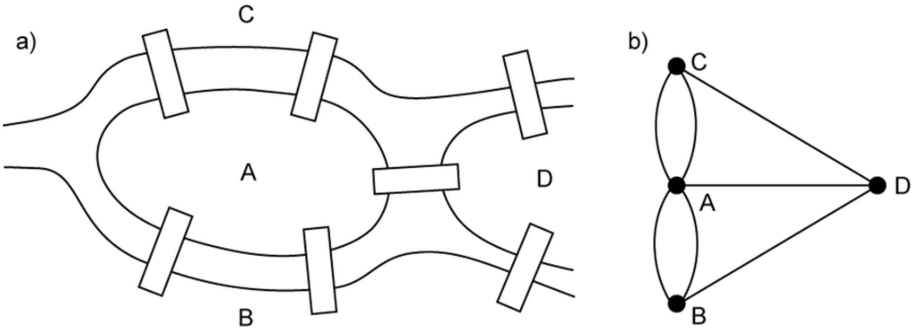

Figure 2.1 Euler's representation of the Königsberg bridge problem with a graph, in which the nodes represent the islands and the edges represent the bridges [Ven+21].

Exact positions and length measurements do not play a role in topology. Topological shapes are flexible, as if made from rubber. We can bend and stretch them without changing their topological identity. A donut and a coffee mug are distinct geometric objects, but as we can continuously deform them to each other, they are indistinguishable in topology. The same holds for a circle and a square.

As this might seem peculiar to a reader not familiar with the foundations of topology, we will now get an overview of the topological foundations relevant in this thesis. Specifically, we will introduce a theory that studies connectivity and holes in shapes, called *homology theory*.

Later, we will introduce *persistent homology*, the primary tool in this master thesis, able to find holes in real-world data using a combination of geometry and topology, enabling us to capture qualitative aspects of data in a quantitative and computable way.

2.2 Topological spaces

Topology is a branch of mathematics that deals with shapes from a global, qualitative perspective. The "shapes" that topology deals with are *topological spaces*, a type of mathematical spaces. In our definition, a topological space is (like any mathematical space) a set X endowed with a structure τ, which we call *topology*. A topology is defined as a set of neighborhoods for each element ("point") of X that satisfies three axioms, formalizing a concept of closeness.

Definition 2.2.1 (Topological space) *Let X be a set, and the structure τ be a collection of subsets, called open sets, of X. τ is called a topology and the ordered pair (X, τ) is called a topological space if [Ede14]*
i) $\emptyset, X \in \tau$,
ii) $U, V \in \tau$ implies $U \cap V \in \tau$, and
iii) $\{U_\alpha \mid \alpha \in I\}$ implies $\cup_{\alpha \in I} U_\alpha \in \tau$.

In words, the conditions are that both X and the empty set are open sets (and therefore part of the topology), the intersection of any two open sets is an open set, and the the union of any family of open sets is an open set.

Note. The terminology "open sets" is simply a name for the elements of the topology, and it might not align exactly with the intuitive notion of openness.

In geometry, determining proximity between points relies on calculating distances. In topology, we replace the notion of distance with the concept of neighborhoods [Moo08]. By defining collections of "open sets", we are specifying closeness (or "nearness") in a topological space. The idea is: If every open set that contains an element $u \in U$ also contains the element $v \in V$, then u and v are "close" [Moo08]. Otherwise, they are not close.

Manifolds

Manifolds are a type of topological space and play a crucial role in almost every branch of mathematics, as well as in physical theories like the string theory or the relativity theory [Lee00]. They also play a significant role in the theory behind TDA.

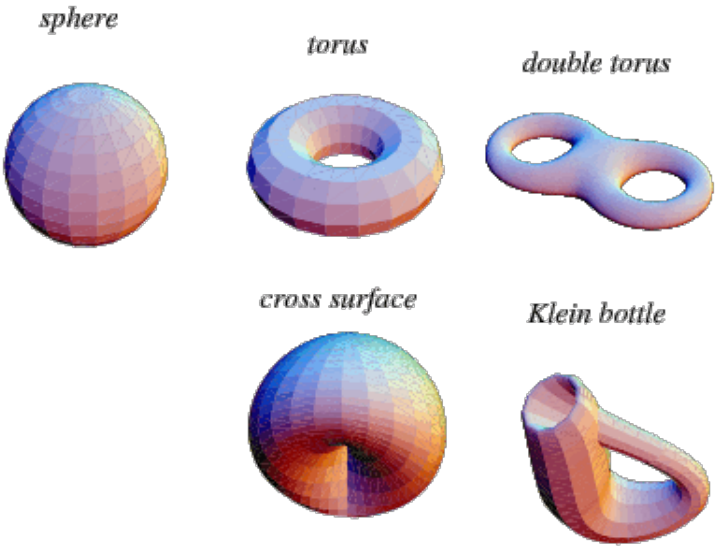

Figure 2.2 Examples of manifolds [Ken18]

Figure 2.2 shows some manifold examples. Some of the manifolds, particularly the cross surface and the Klein bottle, do not seem to resemble those shapes we perceive in our daily lives. They do not fit into the typical framework of Euclidean space. This is because manifolds are only required to look like $\mathbb{R}^n$ *locally.*

Manifolds are topological spaces that are *locally homeomorphic* to a subset of $\mathbb{R}^n$ [Lal14]. We call two topological spaces locally homeomorphic if there is a *local homeomorphism* between them, which, intuitively, is a function that preserves local structure.

Definition 2.2.2 (Local Homeomorphism) *Let X and Y be two spaces. A local homeomorphism between X and Y is a continuous function f: $X \to Y$, such that for every point $x \in X$, there is an open neighborhood U of x such that $f|_U : U \to f(U)$ is a homeomorphism [Wil10].*

Note. Recall that a homeomorphism is a bijective and continuous function whose inverse is also continuous [CVJ21].

To understand how a sphere, for example, is a manifold according to the above definition, let us intuitively compare it to the Earth's surface, so to a globe.

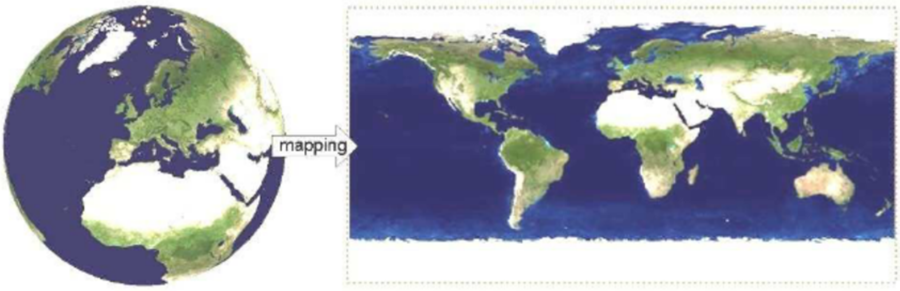

Figure 2.3 Mapping a sphere to a plane [LA10].

If we want to map the surface of a globe to a plane $(\mathbb{R}^2)$ as depicted in Figure 2.3, we cannot directly project it with a homeomorphism. However, for any point on the globe (except the poles), we can consider a small region around that point. We can then use a homeomorphism to project this local region onto a subset of the plane [LA10].

Such a globe is an example of a 2-manifold [Ede14], a manifold of dimension 2, also referred to as *surface* [Lal14]. The dimension of a manifold is, in principle, the number of independent parameters necessary to specify a point in the manifold [Lee00]. The sphere is a 2-manifold as each point on a sphere can be uniquely specified by latitude and longitude or, alternatively, with angular coordinates.

2.3 Homology Theory

Manifolds play an important role in TDA. The "manifold hypothesis" suggests that most real-world data harbors an underlying manifold [FMN16], or at the very least, an intrinsic shape. If we were able to distinguish the underlying manifolds, or shapes, of two data samples, this might help us classify the data samples into different classes.

Distinguishing Topological Shapes

In geometry, differentiating shapes like a circle and a square, which are distinct geometric shapes, seems feasible. In topology, where a circle and a square are equivalent, differentiating shapes appears more challenging.

The most precise way to distinguish topological shapes, or manifolds, is to classify them up to homeomorphism. Classifying up to homeomorphism means classifying shapes that we can transform into each other with a homeomorphism, such as a sphere and a cube, into the same class and classifying other shapes into different classes. If two topological spaces X and Y are of the same topological class, we generally write $X \approx Y$, and say they are "topologically equivalent" [Ede14].

In theory, we can prove that two manifolds are topologically equivalent relatively quickly by finding an explicit homeomorphism between them. It is feasible to show that we can continuously transform a donut and a coffee mug into each other [Cas06], and the same holds for a circle and a square [Cas06].

In practice, finding an explicit homeomorphism between two arbitrary manifolds might be challenging, even if such a homeomorphism exists. There is a whole theory dedicated to classifying manifolds of dimension 2, which is the only dimension for which we know there is only a restricted number of classes [Lal14]. Classifying manifolds of dimension 3 and 4 gets more complex, let alone dimension 5 or greater [Har08]. For any dimension greater than 2, research on classifying manifolds up to homeomorphism is not complete [Lal14], and it has been shown that there cannot be an algorithm for classifying manifolds of dimension greater than 3 [VM05].

Invariants are those properties of manifolds that stay the same under homeomorphism, and one might think that they are useful for topological classification. One example of invariants are *holes* [CM21] and will be defined later. However, if two manifolds have the same holes, they do not have to be of the same class [Sti12], which makes holes useless for exact topological classification. If we know that two manifolds have the same hole structure, we do not get any new insights.

Therefore, we slightly change the question. Instead of asking if two manifolds are of the same class, we only assess if they are of different classes by examining their holes. If two manifolds have different hole structures, they must be of different classes.

If the hole structures of two manifolds differ particularly much, we could say that the manifolds differ much. In a data classification problem, we might label two data samples with different labels if their topological classes differ particularly much and with the same label if their topological classes are similar (even if not the same).

While we can prove on the theoretical side that looking at holes enables us to tell shapes apart, there are also empirical reasons to look at them. There has been extensive research in psychology providing evidence that topological holes are a very effective characteristic for people to discriminate shapes [Pom03] [Wu+21]. This makes them attractive for classification tasks in computer vision and related areas.

Now, what are holes? In fact, there is more than one definition of a hole. Just intuitively, something has a 0-dimensional hole if it is connected, a 1-dimensional hole if you can put your finger through it, 2-dimensional holes if it has holes like cheese, and it can also have higher-dimensional holes.

Different tools exist to measure the hole structure of a manifold, and they may detect different holes. “Homotopy” and “homology” are both such tools [Hat02].

They are both ways of associating groups with topological spaces and detecting holes as characteristics of these groups [Hat02]. The number of holes is counted by the *rank* of the “homotopy groups” and “homology groups”, respectively [Hat02]. In this thesis, we will only introduce homology, as it is the preferred tool in Topological Data Analysis due to its computational efficiency [Car09].

The Number of Holes

Before defining holes in homology theory, we will consider the holes of some exemplary manifolds as an intuition, namely a point, a circle, a torus (previously called donut), and a coffee mug. The number of their k-dimensional holes is denoted by β_k.

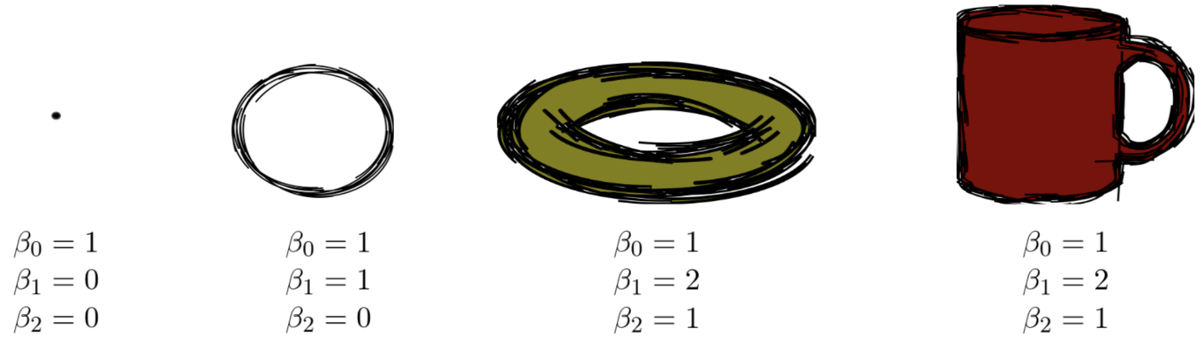

Figure 2.4 Numbers of holes of a point, a circle, a torus and a coffee mug [Dey14].

As shown in Figure 2.4, all of these manifolds count one 0-dimensional hole, called a connected component. The circle also counts a 1-dimensional hole, called a tunnel. This seems plausible with our intuition of “connected components” and “tunnels”.

The number of holes in the torus and the coffee mug might be surprising initially. First, they contain two tunnels, while intuitively, one might only see one.

Second, the torus and the coffee mug also have a 2-dimensional hole, a void. This void is, in essence, their surrounding, which we can consider as one giant void [Tau+20].

Note. The torus and the coffee mug also have the exact same number of holes. This is because they have the same topological class. As depicted in Figure 1.2, we can transform a torus into a coffee mug with a homeomorphism by stretching out one half of the torus to become the cup and shrinking the other half to become the handle, and the other way around [Lee00].

Simplicial complexes

To calculate the number of holes, we first *triangulate* the manifold, which is a way of discretizing it such that the union of the discrete parts can reconstruct the manifold [Car09]; we then look for holes in such a triangulation. The most prominent way of triangulating a manifold is to convert it to a *simplicial complex*, made of small bricks called *simplices* [Qui13].

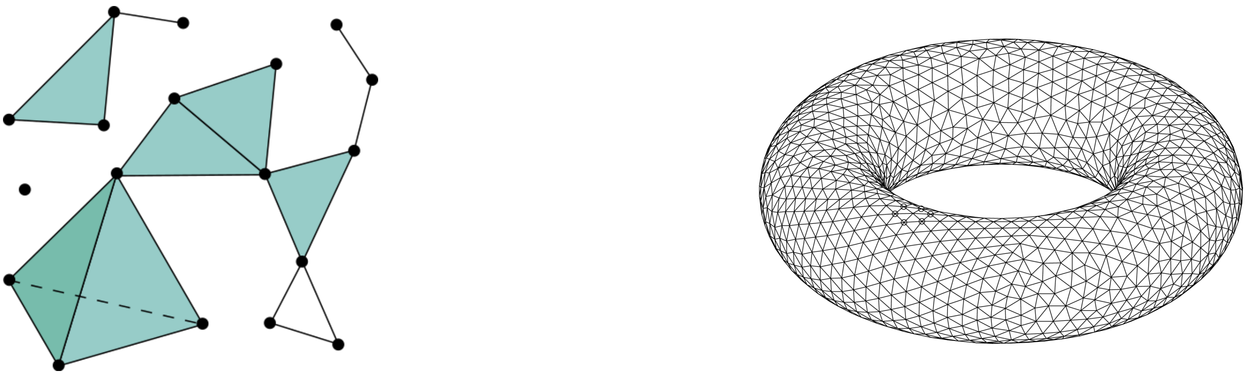

Figure 2.5 Two simplicial complexes. Simple simplicial complex (left) [cfl09] and triangulation of a torus (right) [Ag214].

Two examples of what such simplicial complexes can look like are given in Figure 2.5. On the right, we see a triangulation of a torus; on the left, we see a simpler example of a simplicial complex. It might be reminiscent of a graph. Indeed, simplicial complexes are a generalization of graphs [MHJ22], resulting from adding triangles that link edges and tetrahedra that link triangles, and so on.

The elements of a simplicial complex are vertices, intervals, triangles, tetrahedra, and their higher-dimensional counterparts, and we call them $k-$simplices [EH10]. Figure 2.6 shows examples of $k-$simplices for $k = 0, 1, 2, 3$. We will now introduce a range of definitions, formalizing $k-$simplices and simplicial complexes and eventually leading to the definition of homology groups and, finally, holes.

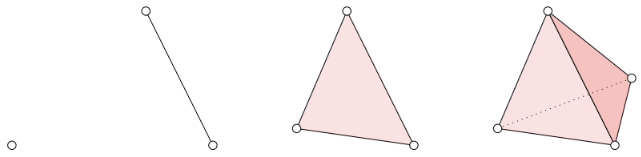

Figure 2.6 Simplices of dimension 0, 1, 2 and 3 [EH10]. We call these simplices vertex, interval, triangle and tetrahedron.

Definition 2.3.1 (Affine Combination, Convex Combination) *Let $A = \{v_0, v_1, \ldots, v_k\} \subset \mathbb{R}^n$ be a set of points in $\mathbb{R}^n$. A point $x = \sum_{i=0}^{k} \lambda_i v_i$ is called affine combination of the v_i if $\sum_{i=0}^{k} \lambda_i = 1$. It is called a convex combination if all λ_i are additionally non-negative.*

The set of affine combinations of points is called an *affine hull*, and the set of convex combinations is called a *convex hull* [EH10].

Definition 2.3.2 (Affine Independent) *A is called an affine independent if any two affine combinations, $\sum \lambda_i v_i$ and $\sum \mu_i v_i$, are the same iff $\lambda_i = \mu_i$ for all i [EH10].*

Now, a simplex is the convex hull of such an affine indepenent set of vertices A.

Definition 2.3.3 (k-simplex) *Let $A = \{v_0, v_1, \dots, v_k\}$ be $k+1$ affine independent points ("vertices") in $\mathbb{R}^n$. A* **k***-simplex is the convex hull of A [EH10]. It is denoted by σ or $[v_0, \dots \hat{v}_i, \dots v_k]$, and has dimension k. Therefore,*

$$\sigma = \left\{ \sum_{i=0}^{k} \lambda_i v_i \mid \lambda_i \geqslant 0, \sum_{i=0}^{k} \lambda_i = 1 \right\} \subset \mathbb{R}^n. \tag{2.1}$$

The convexity of the simplicial complex σ means that we can express all points on the surface of and inside σ as a convex combination of the vertices of σ, which is apparent in Figure 2.6. The affine independence of the vertices implies that no three vertices are collinear and no four vertices coplanar [Gal23], which we can also perceive in Figure 2.6.

Definition 2.3.4 (Face) *A face τ of σ is the convex hull of a non-empty subset of A. We denote this by $\tau \leq \sigma$. A face is called proper, denoted by $\tau < \sigma$, if the subset is not A [EH10].*

Therefore, one obtains a proper face by leaving out at least one vertex in a $k-$simplex.

Definition 2.3.5 (Boundary) *The boundary $\operatorname{bd}\sigma$ of σ is the union of all proper faces of σ [EH10]. It consists of all points for which all $\lambda_i = 0$.*

Definition 2.3.6 (Interior) *The interior int σ of σ is "everything else", so int $\sigma = \sigma - \operatorname{bd}\sigma$ [EH10].*

Intuitively, the interior is the "inner area" of a simplex. It is the set of points of the simplex for which all $\lambda_i > 0$ [EH10].

Example 2.3.1 *The boundary of a triangle consists of 3 connected intervals without the interior (the "inner area") of the triangle. Each of these intervals is a proper face of the triangle.*

Taking a collection of simplices, we obtain a *simplicial complex*, which has dimension k_{max}, with k_{max} being the maximum dimension of all $k-$simplices in the collection [EH10].

Definition 2.3.7 (Simplicial Complex) *A (geometric) simplicial complex K is a finite collection of simplices σ that fulfill the following conditions [EH10]:*
i) ***the face condition:*** *$\sigma \in K$ and $\tau \leq \sigma$ implies $\tau \in K$, and*
ii) ***the intersection condition:*** *$\sigma, \sigma_0 \in K$ implies $\sigma \cap \sigma_0$ is either empty or a face of both σ and σ_0.*

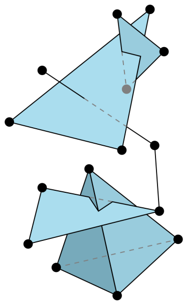

Figure 2.7 Structure of simplices that is not a simplicial complex [Smi09].

To put the face and intersection condition into words, in a simplicial complex, all faces of its simplices are also in the simplicial complex, and if two simplices intersect, then the intersection must be a face of both simplices. While the simplicial complexes in Figure 2.5 fulfill these conditions, Figure 2.7 shows a collection of simplices that is not a simplicial complex. The intersections of some of the simplices are not faces at the same time.

Note. A "geometric" simplicial complex is the "geometric realization" of an "abstract" simplicial complex in certain cases [Dey22]. Because geometric and abstract simplicial complexes are commonly handled equivalently [Dey22], we will drop the words "abstract" and "geometric" and not make a difference between those two complexes.

Homology Groups

Once we have constructed a simplicial complex from our manifold, we can look for holes in the complex. In simple words, holes are, as we might expect, "empty spaces". However, defining them mathematically and finding them computationally is not as simple, such that we instead define and look for their *duals*: "non-bounding cycles" [Zom05]. The numbers of holes and non-bounding cycles of a simplicial complex are the same [Zom05], and these numbers are what we are eventually interested in.

Returning to our example of a triangulated torus, its two non-bounding cycles of dimension one are shown in Figure 2.8. It is becoming clearer why, as we observed in Figure 2.4, a torus has two holes (and, equivalently, non-bounding cycles) of dimension 1 and not just one, as we might expect.

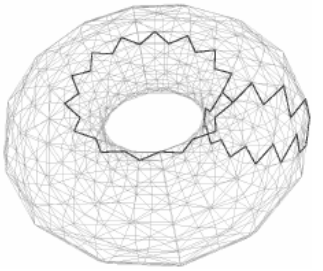

Figure 2.8 1-dimensional non-bounding cycles of a torus [Pel+07].

Non-bounding cycles (and therefore also holes) are counted by the rank of so-called *homology groups*. To define homology groups, we rely on the concept of *boundary operators*.

A function that maps combinations of triangles to combinations of intervals is an example of a "boundary operator". This boundary operator is of dimension 2, but boundary operators exist for all dimensions.

More generally, boundary operators map so-called *chain groups*, consisting of collections of $k-$simplices (so-called $k-$*chains*), to lower-dimensional chain groups.

Definition 2.3.8 ($k-$chains) *Let K be a simplicial complex. A k-chain c is a formal sum $c = \sum_{i=1}^{n} a_i \sigma_i$ of k-simplices σ_i, $i = 1, ..., n$, in K [EH10].*

This formal sum is not to be understood as an arithmetic operation; it rather expresses a way of combining $k-$simplices. Its coefficients a_i indicate how often a simplex appears in the combination [Car13]. We often use module 2 coefficients in practice, so either 0 or 1 [EH10].

Definition 2.3.9 (Chain Group) *For each k, all possible k-chains in K, together with the addition operation, form the group of k-chains, which we denote as V_k [EH10].*

Recall that what we want to achieve by defining chain groups is to give a formal definition of boundary operators.

Definition 2.3.10 ($k-$th Boundary Operator) *The $k-$th boundary operator, denoted by ∂_k, is a linear transformation which maps from a chain group V_k to the lower-dimensional chain group V_{k-1} [EH10]. The boundary operator applied on a single k-simplex $\sigma = [v_0, \dots, v_k]$ is given by the formal sum [Buc18]*

$$\partial_k \left([v_0, \dots, v_k]\right) = \sum_{i=0}^{k} (-1)^i \left[v_0, \dots \hat{v}_i, \dots v_k\right], \tag{2.2}$$

where $[v_0, \dots \hat{v}_i, \dots v_k]$ denotes the $(k-1)$ simplex obtained by removing v_i.

Due to its linearity, the boundary operator applied on combinations of $k-$simplices ($k-$chains) is a combination of boundary operators applied on the simplices: $\partial_k \left(\sum a_i \sigma_i\right) = \sum a_i \partial_k \left(\sigma_i\right)$.

Note. After applying the boundary operator to a $k-$simplex, it is common for some resulting $(k-1)$-simplices to have a negative sign. This negative sign reverses the "orientation" of the $k-1-$simplices. Orientation is an important concept in topology. Visually expressed by arrows in a simplex as in Figure 2.9, orientation determines the path one would traverse if walking along the simplex [ST80].

Example 2.3.2 (Boundary Map of a Triangle) *Figure 2.9 shows the 1st and 2nd boundary operator applied to an oriented interval $[v_0, v_1]$ and an oriented triangle $[v_0, v_1, v_2]$, respectively. We obtain $\partial_1 \left([v_0, v_1]\right) = [v_1] - [v_0]$ and $\partial_2 \left([v_0, v_1, v_2]\right) = [v_1, v_2] - [v_0, v_2] + [v_0, v_1]$ [Buc18]. If we inverted the orientation of $[v_0, v_1]$, we would obtain $\partial_1 \left(-[v_0, v_1]\right) = [v_0] - [v_1]$ [EH10].*

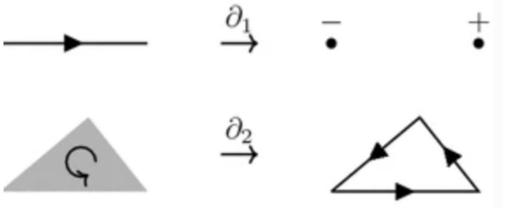

Figure 2.9 Examples of the boundary operator applied on an oriented interval and an oriented triangle [Buc18].

The output of a boundary operator applied to a $k+1$-chain is a $k-$chain, a formal sum of boundaries of simplices in the $k+1-$chain, collectively called the "$k-$boundary" of the $k+1-$chain. The output of the boundary operator applied to the *group* of $k+1-$chains (the "chain group" V_{k+1}), again forms a group called *boundary group.*

Definition 2.3.11 (Boundary Group) *We define the boundary group of dimension k as $B_k(K) := \mathrm{Im}\left(\partial_{k+1}\right)$ [CVJ21].*

The elements of the boundary group are called $k-$boundaries.

Sometimes, the $k+1-$th boundary operator applied to $k+1-$chains yields zero. For instance, the output of a boundary operator applied to a vertex is, by definition, zero [EH10], and the same holds for a collection of vertices. For higher-dimensional $k+1-$chains, an example in which the output of the $k+1-$th boundary operator becomes zero is given in Example 2.3.3.

We refer to $k+1-$chains on which the $k+1-$th boundary operator yields zero as "cycles", and collectively, they form a "cycle group".

Definition 2.3.12 (Cycle Group) *We define the k-th cycle group $Z_k(K)$ as $Z_k(K) := \mathrm{Ker}\left(\partial_k\right)$ [CVJ21].*

Example 2.3.3 *The triangle in Figure 2.9 is a cycle: $\partial_2 \left([v_0, v_1, v_2]\right) = [v_1, v_2] - [v_0, v_2] + [v_0, v_1] = 0.$*

Now, *every $k-$boundary is also a $k-$cycle.* This is stated by the following theorem.

Theorem 2.3.13 (Fundamental Lemma of Homology) $\partial_k \circ \partial_{k+1} = 0$*, for all k [EH10].*

A proof of this theorem can be found in [Car13].

The statement that the output of the $k+1-$th boundary operator (a $k-$boundary) is also a $k-$cycle is shown by the fact that when applying, in turn, the $k-$th boundary operator ∂_k on this output, we receive zero, meaning that it belongs to the kernel of ∂_k. This means that every boundary is also a cycle.

The converse, that every cycle is also a boundary, is not true; it is simple to find a counter-example [Tan17]. Such cycles are called *non-bounding cycles*, which is the structure we are looking for in order to count holes.

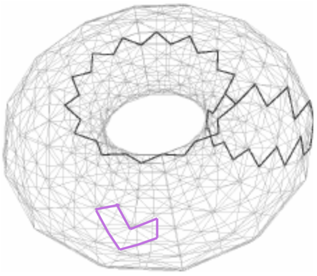

Figure 2.10 Non-bounding 1-cycles (black) and bounding 1-cycle (pink) of a torus [Pel+07].

Example 2.3.4 (Bounding Cycles, Non-Bounding Cycles) *Figure 2.10 shows the two previously seen non-bounding* $1-$*cycles (black) on the triangulation of a torus. Now, it also shows an example of a bounding* $1-$*cycle (pink). The pink cycle is bounding as it is the boundary of a combination of triangles.*

Non-bounding cycles add to the number of holes in a simplicial complex, bounding cycles do not. We can count the number of non-bounding cycles by the rank of *homology groups*, which are just the groups of non-bounding $k-$cycles that are not $k-$boundaries at the same time. We can express such a homology group $H_k(K)$ as the *quotient group* of the boundary group $B_k(K)$ and the cycle group $Z_k(K)$.

Definition 2.3.14 (Normal Subgroup) *Let G be a group and N be a subgroup of G. If $gNg^{-1} = N$ for all $g \in G$, N is called a normal subgroup [Lee00]. Here, $gN = \{gn : n \in N\}$, and $Ng = \{ng : n \in N\}$ [Lee00].*

Definition 2.3.15 (Quotient Group) *Let G be a group and N be a normal subgroup of G. Then, we define the quotient group as $G/N := \{gN \mid g \in G\}$ [Lee00].*

The k-th homology group is a quotient group defined by removing (the equivalence classes of) $k-$cycles that are also $k-$boundaries.

Definition 2.3.16 ($k-$th Homology Group) *Let K be a simplicial complex. The $k-$th homology group $H_k(K)$ is defined as $H_k(K) := Z_k(K)/B_k(K) = \operatorname{Ker} \partial_k / \operatorname{Im} \partial_{k+1}$.[Lee00].*

The elements of a homology group $H_k(K)$ are equivalence classes, where the equivalence relation is called homology [Car13]. Each equivalence class consists of non-bounding cycles that we can continuously deform into each other. In Figure 2.10, the two black non-bounding cycles are just representatives of their homology class. They can also arbitrarily be deformed continuously while still representing the same hole.

Betti numbers, the rank of the homology groups, count the numbers of non-bounding cycles and, simultaneously, of holes.

Definition 2.3.17 (Betti Number) *Let K be a simplicial complex. The $k-th$ Betti number of K, $\beta_k(K)$, is the rank of $\mathrm{H}_k(K)$ [EH10].*

As $\mathrm{H}_k = \mathrm{Z}_k/\mathrm{B}_k$, Betti numbers can be calculated as [CVJ21]

$$\operatorname{rank} \mathrm{H}_k = \operatorname{rank} Z_k - \operatorname{rank} \mathrm{B}_k. \tag{2.3}$$

The $0-$th Betti number β_0 counts the number of distinct connected components (which are $0-$dimensional holes) of a simplicial complex [ELZ02].

The $1-$st Betti number β_1 counts the number of tunnels, which are $1-$dimensional holes, and the $2-$nd Betti number β_2 counts the number of voids, which are $2-$dimensional holes [ELZ02].

Betti numbers count both non-bounding cycles and holes in a simplicial complex. They are an important quantity that helps classify manifolds into different topological classes. Note that the numbers $\beta_k(K)$, $k = 0, 1, 2$, we looked at in Figure 2.4 are Betti numbers.

Why the Number of Holes Distinguishes Shapes

To understand why two manifolds with a different number of holes are not topologically equivalent, we will rely on a weaker form of equivalence, called homotopy equivalence, which can still help distinguish topological features.

Homotopy equivalence will also be a crucial part of later understanding why *persistent homology*, the primary tool of this master thesis, works.

Intuitively, two manifolds X and Y are *homotopy equivalent* if there is a way to transform X into Y and Y into X by *homotopic* functions.

Definition 2.3.18 (Homotopic Functions) *Let X, Y be two topological spaces. Two continuous functions $f, g : X \longrightarrow Y$ are homotopic if there is a continuous function H between $[0, 1] \times X$ and Y, such that $H(0, x) = f(x)$ and $H(1, x) = g(x)$ for all x in X. We denote homotopic functions by $f \cong g$ [Car09].*

Here, the interval $[0, 1]$ can be understood as a parameter measuring the progress of the continuous transformation. At $(0, x)$, H represents the initial state of the transformation, at $(1, x)$, it represents the final state.

Definition 2.3.19 (Homotopy Equivalence) *Let X, Y be two topological spaces. A map $f : X \longrightarrow Y$ is called a homotopy equivalence if there is another map $g : Y \longrightarrow X$ such that f and g fulfill the following conditions [Car09]*
i) $f \circ g$ and the identity map on Y ($id_Y : Y \longrightarrow Y$) are homotopic
ii) $g \circ f$ and the identity map on X ($id_X : X \longrightarrow X$) are homotopic.

While topological equivalence means that *one* map f can transform a manifold X to a manifold Y and back, in *homotopy* equivalence, we introduce another map g. More precisely, while in topological equivalence, $f \circ f^{-1}$ is the identity map from Y to Y and $f^{-1} \circ f$ is the identity map from X to X, in homotopy equivalence, g serves as a substitute for f^{-1}. This makes homotopy equivalence a weaker equivalence relation than topological equivalence.

We can show that if two manifolds have different Betti numbers, they are not homotopy equivalent, as a direct consequence of Theorem 2.3.20.

Theorem 2.3.20 *If two manifolds are homotopy equivalent, their Betti numbers are the same [Car09].*

A proof for this theorem can be found in [Hat02] and [RV07]. The theorem is, essentially, a corollary of the *functoriality* of homology [Hat02]. Homology is "functorial", which means that it does not just assign a group $H_k(X)$ to each topological space X, it also assigns an *induced map* $f_* : H_k(X) \to H_k(Y)$ to each map $f : X \to Y$, and preserves important properties of the of the topological space and the map, respectively [Hat02].

Now the groups $H_k(X)$ and $H_k(Y)$ are the same for two homotopy equivalent spaces X and Y (which is stated in Theorem 2.3.20). It follows that if two manifolds X and Y have different homology groups $H_k(X)$ and $H_k(Y)$ and therefore different Betti numbers, they are not homotopy equivalent. As topological equivalence is stronger than homotopy equivalence, we then also know that X and Y are not topologically equivalent. We can conclude that even though we might not know that two spaces are of the same topological class if they have the *same* Betti numbers, we do know that they are of different classes if they have *different* Betti numbers.

In a data classification problem, we could identify if the manifolds behind two data samples are of different topological classes. In a geometric setting, we can then ask the question of how different their topological classes are in order to classify data samples that harbour similar manifolds into the same class and data samples with very different manifolds into different classes.

Cubical Complexes

We will see that simplicial complexes are suitable for analyzing time series data. For image data, *cubical complexes* can be more suitable. Betti numbers do not depend on the type of triangulation [RV07], but the choice of triangulation can make a difference regarding computational efficiency.

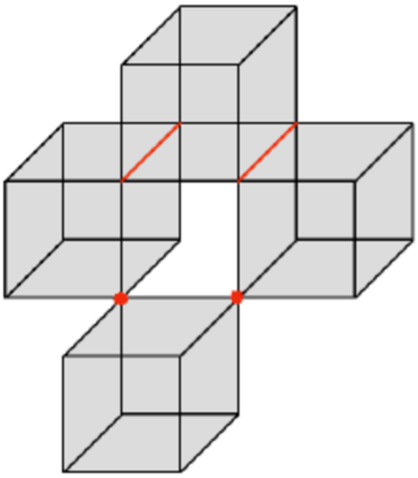

Figure 2.11 Exemplary cubical complex [GDJM11].

While simplicial complexes consist of points, intervals, triangles, tetrahedra, and their higher-dimensional counterparts, cubical complexes consist of points, line segments, squares, cubes, and n-dimensional counterparts. Figure 2.11 shows an example of a cubical complex.

Cubical complexes provide a perfect fit to the inherent cubical structure of images, which consist of pixels [KMM04]. To understand cubical complexes, let us first define *elementary intervals* and *cubes.*

Definition 2.3.21 (Elementary Interval) *For some $l \in \mathbb{Z}$, the corresponding elementary interval is one of the two closed intervals $I \subset \mathbb{R}$ [KMM04]*

$$I = [l, l + 1] \quad \text{or} \quad I = [l, l]. \tag{2.4}$$

Elementary cubes are the the product of elementary intervals. They are the pendant to simplices in cubical homology theory and the building blocks of cubical complexes [KMM04].

Definition 2.3.22 (Elementary Cube) *An elementary cube Q is the finite product of elementary intervals $I_1, \cdots, I_N$ [KMM04]*

$$Q = I_1 \times \cdots \times I_N \subset \mathbb{R}^N. \tag{2.5}$$

The dimension N of an elementary cube is the number of non-degenerate components it contains, where non-degenerate intervals are those elementary intervals that are those of the form $[l, l+1]$ [KMM04].

Cubical sets are finite unions of elementary cubes [KMM04]. We can define boundary operators and chain groups V_k on a cubical set A similarly as in simplicial homology theory [KMM04]. The boundary operators ∂_k are maps that map chain groups to lower-dimensional chain groups again.

Finally, cubical complexes are defined as follows.

Definition 2.3.23 (Cubical Complex) *The cubical (chain) complex for a cubical set $A \subset \mathbb{R}^N$ is [KMM04]*

$$C(A) := \{V_k(A), \partial_k\}_{k \in \mathbb{Z}}. \tag{2.6}$$

We can define homology groups of a cubical set X similarly as we defined them for simplicial complexes: As the quotient groups $H_k(X) := Z_k(A)/B_k(A)$, where $Z_k(A) := \ker \partial_k$ and $B_k(A) := \operatorname{im} \partial_{k+1}$ [KMM04].

Summing Up

In this chapter, we learned how to identify holes in the underlying manifolds behind real-world data. Manifolds can have holes of various dimensions, and the holes of the first three dimensions are called connected components, tunnels, and voids.

The most prominent method for detecting such holes in a manifold is to discretize the manifolds to a simplicial or cubical complex. In such complexes, we do not look for holes ("empty spaces") directly, but we look for a closely related structure with the same amount as holes: Non-bounding cycles.

We aim to detect non-bounding cycles in the manifolds behind our brain data, which will help us classify the data into different consciousness stages, as different Betti numbers of such manifolds indicate different topological classes of the manifolds.

We have seen some examples of what manifolds can look like. However, considering image or time series data, the question of what manifold may lie behind the data is not intuitive. The next chapter will explore a way to detect non-bounding cycles in real-world data.

2.4 Persistent Homology - Finding Holes in Real-Word Data

So far, we have learned a method for counting holes in triangulations of topological spaces like manifolds. However, the data we deal with in this thesis is not an abstract and idealistic manifold. We are dealing with real-world data.

In the theory of Topological Data Analysis, all real-world data, at least when converted to point clouds, may noisily resemble some underlying manifold or, more generally, some underlying shape.

For instance, our real-world data D might be a point cloud concentrated very near a circle, like in Figure 2.12. The topological properties of the underlying manifold X, in this case a circle, provide valuable information about the data.

Figure 2.12 Data sampled from a circle with little noise (left) and much noise (right).

The underlying manifold behind data points is not always obvious. There are approaches that aim to identify the manifolds behind data, summarized as "Manifold Learning" approaches [Ize12]. We will, however, use a direct way to analyze the data's intrinsic topological structure without explicitly identifying the underlying manifold. Instead of finding such a manifold and converting it into a simplicial complex, we want to generate a complex on our data points directly.

There are several ways to construct complexes from data. For constructing, for example, a so-called *Čech complex* from data points, we first define a parameter ϵ ($\epsilon \geq 0$). We define the vertices of our complex to be our data points, and we now imagine open balls with radius ϵ around each vertex. If the $\epsilon-$balls around two vertices have a common point of intersection, we construct an interval between the vertices. If the $\epsilon-$balls around three vertices have a common point of intersection, we construct a triangle between the vertices, and so on.

Definition 2.4.1 (Čech Complex) *Let D be a point cloud. For a given $\epsilon > 0$, we define balls $B_\epsilon(z_i)$ around each data point $z_i \in D$, which we simultaneously define as vertices. Then, the Čech complex is a simplicial complex constructed by creating a k-simplex for every $(k+1)$-tuple of intersecting balls $\{B(z_i, \epsilon)\}_{i=1}^{k+1}$ [NSW08] [CVJ21]. We denote the Čech complex by $C^{\check{C}ech}(D, \epsilon)$.*

As the Čech complex can be computationally expensive, we often use an alternative which is, in a sense, an approximation of the Čech complex: The *Vietoris-Rips complex* [Car09].

The idea of constructing a Vietoris-Rips complex is similar to constructing a Čech complex: We construct open $\epsilon-$balls around all data points, which we define as our vertices. If two vertices are less than 2ϵ apart, such that the $\epsilon-$balls around them touch, we connect them with an interval. If three vertices are pairwise close enough, we construct a triangle on them.

The difference to the Čech complex is that to construct a triangle on three vertices, the balls around them do not need one common point of intersection; they only need to intersect pairwise (and similarly for higher dimensions). Figure 2.13 shows an example of a Vietoris-Rips complex constructed on a point cloud that looks different than the respective Čech complex. Note that the filtration parameter is denoted by α here, not ϵ.

Definition 2.4.2 (Vietoris-Rips Complex) *Let D be a point cloud. For a given $\epsilon > 0$. we define the point $z_i \in D$ as the vertices of a simplicial complex. For $k \geq 1$, define $k-$simplices between vertices $\{z_0, \dots, z_k\}$ iff, for $0 \leq i < j \leq k$ [CVJ21],*

$$d\left(z_i, z_j\right) \leq 2\epsilon. \tag{2.7}$$

We call this simplicial complex the Vietoris-Rips complex and denote it by $C^{VR}(D, \epsilon)$.

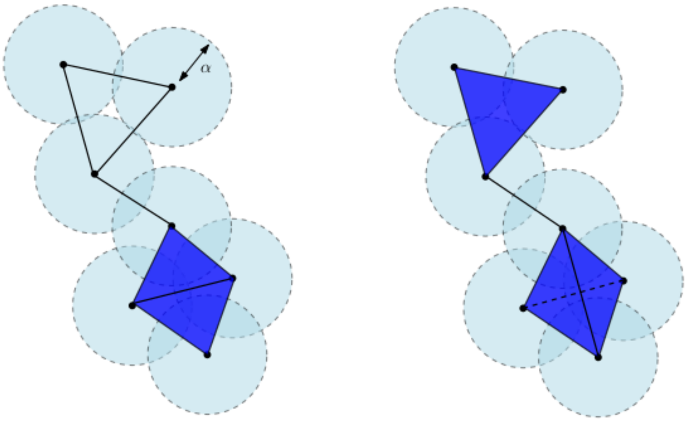

Figure 2.13 Čech complex $C^{\check{\text{C}}\text{ech}}(D, \alpha)$ (left) and Vietoris-Rips complex $C^{\text{VR}}(D, \alpha)$ (right) constructed on a point cloud D [CM21].

Filtrations

The most critical question for constructing a simplicial complex from a point cloud is how to set the threshold of the parameter ϵ. It has been found that this is a "very hard, if not unsolvable problem" [Car19].

In 1999, [Rob99] introduced the idea of *persistence*, later extended by [ELZ02]. Instead of choosing just one threshold for the parameter ϵ, we observe how a simplicial complex evolves over a range of parameter values.

The idea is that those holes that *persist* over a larger range of parameters are more *significant* than those holes that persist only over a small range of parameters [Klo10]. More persistent holes exist over a larger subsequence of simplicial complexes we obtain when varying ϵ, which we call a *filtration*.

Definition 2.4.3 (Filtration) *A filtration on a simplicial complex K is a nested sequence of simplicial complexes*

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{m-1} \subseteq \mathrm{K}_m = \mathrm{K}, \tag{2.8}$$

in which K^i is a subcomplex of K^{i+1} and each K^i a subcomplex of K [CVJ21].

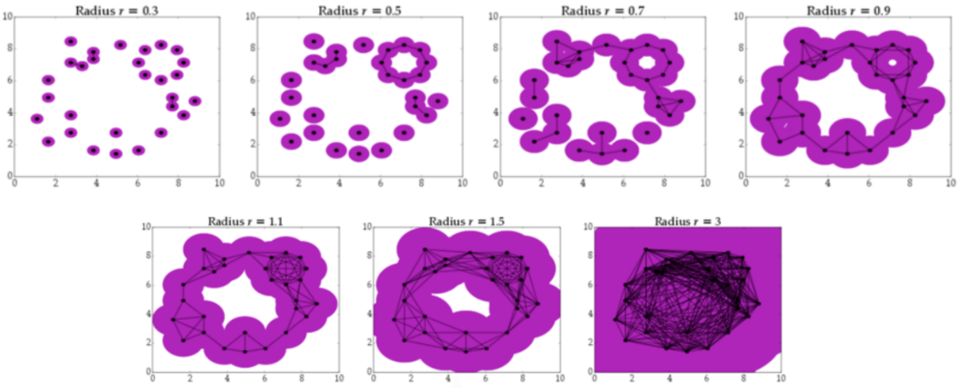

Figure 2.14 Vietoris-Rips filtration of a point cloud [Mun17].

We receive such a filtration, an increasing sequence of complexes, by defining a sequence of filtration parameter values $\{\epsilon_i\}_{i=0}^{m}$ and defining complexes for all parameter values. Figure 2.14 shows an exemplary Vietoris-Rips filtration constructed on a point cloud. Each of the seven subcomplexes in the filtration has a different number of holes.

The Vietoris-Rips filtration and the Čech filtration are particularly suitable for point clouds. We will later see how to convert time series to point clouds to be able to apply these filtrations on time series. Another, more direct way to construct a filtration from a time series is to connect all points that lie in the *sublevel set* to simplices. It is called *sublevel set filtration.*

Definition 2.4.4 (Sublevel Set) *Given a space X and a real-valued function $f : X \longrightarrow \mathbb{R}$, the sublevel set for $a \in \mathbb{R}$ consists of all the points in X with $f(x) \leq a$ [Mak+18].*

A filtration particularly suitable for images is called *radial filtration* [Tau+20]. In a radial filtration, a greyscale value is assigned to each pixel of a binary image based on a reference pixel called the "center" and a filtration parameter, the "radius." If the binary pixel is active and lies within a ball defined by the center and the radius, then the assigned value equals the distance to the center. Otherwise, the assigned value equals the maximum distance between any pixel of the image and the center pixel, plus one [Tau+20].

Persistence Diagrams

In persistent homology, we are interested in the parameter range during which each hole persists. Those holes that have the property that they persist over a larger parameter range also appear more significant in the data [Klo10].

Holes can be "born" or "die" in a filtration, in the beginning ϵ_i and in the end ϵ_j of the parameter range. In dimension 0, for example, a hole is born when a new connected component arises, and it dies when two connected components merge into one [Dey22].

We can put the birth-death tuples (ϵ_i, ϵ_j), $i \leq j$, into a diagram in which each tuple represents a hole. This diagram is called the *persistence diagram* [Mar17], and the the distance of a hole to the diagram diagonal is called the *persistence*; again, persistence is a measure of the significance of the hole in the data [Dey22].

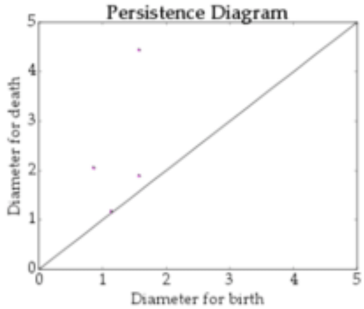

Figure 2.15 Persistence diagram of the filtration in Figure 2.14 [Mun17].

Figure 2.15 shows the persistence diagram with the birth-death tuples of 1-dimensional holes ("tunnels") of the filtration in Figure 2.14. We observe that one hole, the most prominent hole, is clearly distinguishable from less persistent holes in the diagram.

This hole is the hole in the center of the large, central circle in Figure 2.14. We observe that it exists over a particularly long range of the filtration parameter ϵ (called r in the figure). The tunnel in the center of the smaller circle persists over a smaller range of the filtration parameter, and additional tunnels might appear over such a short parameter range that they could be considered noise.

Figure 2.16 shows another example of persistence diagrams. It shows the persistence diagrams of the two circle point clouds shown in Figure 2.12, showing holes of dimensions 0, 1 and 2. Again, the prominent tunnel of the circle (in green) is clearly distinguishable from other holes.

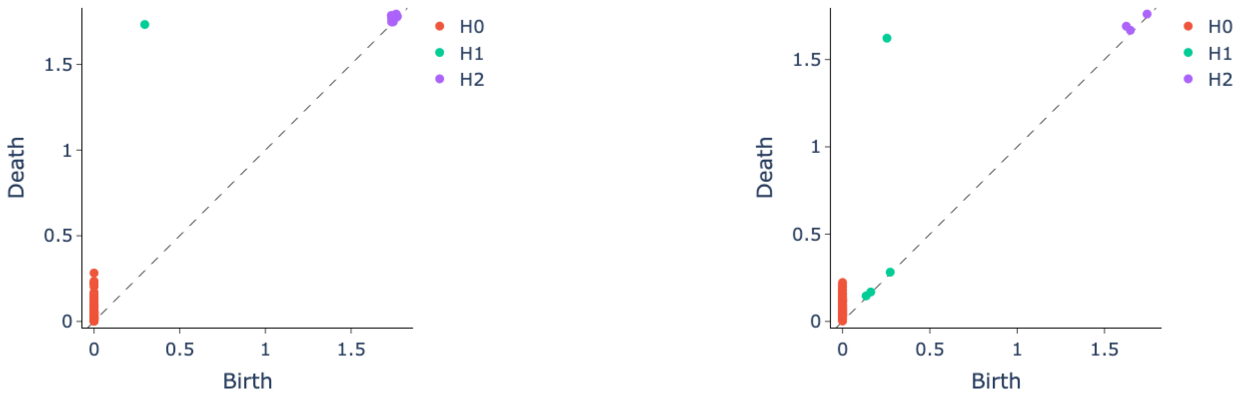

Figure 2.16 Persistence diagrams of a point cloud sampled from a circle with little (left) and much (right) noise.

Why Persistent Homology Works

If our data points D are sampled from a manifold X, how do we know that persistent homology reflects the homology of X? In fact, this depends on how densely we sampled D.

Based on previous work [CSEH07], Niyogi et al. provide conditions under which we can conclude with high confidence that the homology of a Čech complex constructed on data points D sampled from a manifold X equals the homology of X [NSW08].

Apart from conditions on the underlying manifold X, these conditions include an estimation of how dense we need to sample data points and an optimal range of the filtration parameter ϵ [NSW08].

The contribution of Niyogi et al. is to show that if data points $D = \{z_1, \ldots, z_n\}$ are sampled sufficiently densely (but noisily) from a manifold X, there is such a range of ϵ for which the homology of the union of open $\epsilon-$balls around the data points, $\bigcup_{z_i \in D} B_\epsilon(z_i)$, equals the homology of the manifold X [NSW08].

How does this help us? In fact, we need an additional theorem to make use of this knowledge, the so-called *Nerve Lemma*. By this lemma, we learn that the homology of the Čech complex is the same as the homology of $\bigcup_{z_i \in D} B_\epsilon(z_i)$. Therefore, by combining the Nerve Lemma with the findings of Niyogi et al., we know that *the Čech complex can reconstruct the homology of the original manifold X.*

To understand the Nerve Lemma, we need to have a new perspective on Čech complexes. A new perspective on Čech complexes, which consist of $k-$simplices that are created for each $k+1-$tuples of intersecting $B_\epsilon(z_i)$), is that they are the *nerves* of $\{B_\epsilon(z_i)\}_{z_i \in D}$ [NSW08].

Definition 2.4.5 (Nerve) *Let $\{U_1, \ldots, U_n\}$ be a finite collection of sets. Then, the nerve of $\{U_1, \ldots, U_n\}$ is a simplicial complex constructed by creating a k-simplex for each non-empty intersection of $k + 1$ distinct sets* *from $\{U_1, \ldots, U_n\}$. More precisely, a k-simplex is formed for every subset $\{U_{i_0}, U_{i_1}, \ldots, U_{i_k}\}$ of $\{U_1, \ldots, U_n\}$ such that $U_{i_0} \cap U_{i_1} \cap \cdots \cap U_{i_k} \neq \emptyset$. [CVJ21] [NSW08].*

To understand the Nerve Lemma, we also need to know the concept of "open coverings". Open coverings are collections of open subsets of some set A whose union reconstruct A [CVJ21].

Theorem 2.4.6 (Nerve Lemma) *Let $\{U_1, \ldots, U_n\}$ be an open covering of A, a subset of $\mathbb{R}^n$. If each intersection of the U_i is either empty or contractible, then the nerve of $\{U_1, \ldots, U_n\}$ is homotopy equivalent to A [CVJ21].*

Note that in Theorem 2.4.6, *contractible* refers to the property of a form that makes us able to continuously deform (or "contract") the form to a point within itself [Car09].

The ϵ-balls $B_\epsilon(z_i)$, $z_i \in D$, form an open covering of $\bigcup_{z_i \in D} B_\epsilon(z_i)$ [CVJ21]. Their intersections are contractible as they are convex [NSW08], and any convex set in $\mathbb{R}^n$ is contractible to a point [ALS19]. With the Čech complex as the nerve of the $B_\epsilon(z_i)$, $z_i \in D$, we can apply the Nerve Lemma to our setting.

We know that the nerve of open ϵ-balls around some data points D is a Čech complex, and therefore can conclude from the Nerve Lemma that the Čech complex is homotopy equivalent to the union of the ϵ-balls, $\bigcup_{z_i \in D} B_\epsilon(z_i)$. Recall that homotopy equivalence, which we defined in Section 2.2, implies the same Betti numbers, so we can conclude that the Čech complex and $\bigcup_{z_i \in D} B_\epsilon(z_i)$ have the same homology.

With the work of Niyogi et al. [NSW08], we can now infer that there is a range of ϵ for which the Čech complex constructed from open $\epsilon-$balls around data points D, which are sampled from a manifold X, has the same homology as the original manifold X (if the data points D are sampled densely enough). [ALS13] proves that the same result for the Vietoris-Rips complex.

Recall that finding ideal parameter values of ϵ in practice is a "hard, if not unsolvable problem". In persistent homology, we, therefore, choose a large range of ϵ that will likely include such "ideal" parameter values. Those holes with a large persistence are more likely to fall into the suitable range of ϵ, which is one reason they are significant. In contrast, holes with a smaller persistence are less likely to fall into a suitable range of ϵ, such that it could be reasonable to exclude such non-persistent holes from our analysis and label them as "noise".

Persistent Homology - Combining Geometry and Topology

Unlike manifolds, real-world data is not as flexible as if made from rubber. Imagine our data points D to be a time series, potentially an EEG recording. In such a recording, the size of a fluctuation *does* play a role. Additionally, small-scale fluctuations *can* play a role in EEG, as we can infer neuroscientific knowledge from them, so we do not want to discard them.

We will see in Section 5.2 how to construct a Vietoris-Rips filtration from a time series. For now, we can assume that small fluctuations in a time series tend to be reflected in small-persistence holes in the respective persistence diagram.

Therefore, it is a natural question to ask if it is a good strategy to exclude small-scale holes ("noise") from our analysis. Traditionally, research that has applied persistent homology for data analysis tended to focus on large-scale holes [Pat+18], such that excluding noise is a traditional and well-explored way. Small-scale holes likely do not encode any important topological information, as it is unlikely they reflect a hole that is present in the manifold X behind our data D.

However, persistent homology inherently combines topology and geometry. It measures the *geometric* size of holes behind data (and can additionally detect geometric quantities such as curvature and convexity [TMO22]) and counts large-scale holes that encode *topological*, qualitative information.

As said, small-scale holes will likely not encode *any* topological information of the underlying manifold behind the data. However, according to [Pat+18], they can reflect local, *geometric* information of the data that might play an important role in data analysis.

Figure 2.17 Flock of birds [AM21].

A picture of a flock of birds, as shown in Figure 2.17, contains both local and global information. We can infer knowledge about the whole flock from global information, and about the single birds in the flock from local information. In a data analysis problem, solely concentrating on global information, which is reflected in large-scale holes in persistence diagrams, might limit our understanding of the intricate structures within the data. It might therefore be a good strategy not to exclude small-scale holes from our analysis.

2.5 Stability

The goal of this master thesis will be to classify data samples $D_1, D_2, \ldots$ into consciousness stages, such as different depths of anesthesia. Two brain recordings D_1 and D_2 that are very similar from a purely geometric point of view, such that the (geometric) distance between them is very small, likely come from the same anesthesia stage and should be classified into the same class. As our classification is based on persistence diagrams, it would be therefore desirable for persistence diagrams to be very similar for any two data samples that are geometrically very similar.

When measuring the geometric similarity of data samples by the so-called *Gromov-Hausdorff distance* $d_{GH}(\cdot)$, a geometric distance [Mó8] which we refrain from defining precisely, we indeed know that the respective persistence diagrams hold such a desirable property. This property is called the *stability* of persistence diagrams.

Before introducing the notion of stability, we will introduce the distances by which we measure topological similarity between persistence diagrams.

Topological Distances

Two common topological distances, which compute a topological similarity between persistence diagrams, are the *Bottleneck distance* and the *Wasserstein distance*.

$\|(x, y)\|_\infty = \max\{|x|, |y|\}$ denotes the usual $L_\infty-$norm.

Definition 2.5.1 (Wasserstein Distance) *For $p > 0$, define the $p-$Wasserstein distance between two persistence diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$ as [Tau+20][CVJ21]*

$$d_{\mathrm{W}}(\mathcal{D}_1, \mathcal{D}_2) = \inf_{\gamma} \left(\sum_{x \in \mathcal{D}_1} \|x - \gamma(x)\|_{\infty}^{p} \right)^{1/p}, \tag{2.9}$$

γ are bijections from $\mathcal{D}_1$ to $\mathcal{D}_2$. We also include the points on the diagonals of $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $\mathcal{D}_1$ and $\mathcal{D}_2$ do not need to have the same number of points [KMN17].

Conceptually, the Wasserstein distance measures the minimal cost of moving the points in $\mathcal{D}_1$ to align with those in $\mathcal{D}_2$.

For $p \longrightarrow \infty$, we call the $p-$Wasserstein distance *Bottleneck distance.*

Definition 2.5.2 (Bottleneck Distance) *The Bottleneck distance is the $p-$Wasserstein distance for $p \longrightarrow \infty$ [CVJ21]*

$$d_{\mathrm{B}}(\mathcal{D}_1, \mathcal{D}_2) = \inf_{\gamma} \sup_{x} \|x - \gamma(x)\|_{\infty}. \tag{2.10}$$

The Bottleneck distance has a limitation in that it is solely influenced by the largest distance between pairs of points and does not consider the proximity of the other point pairs [CVJ21].

Stability of Persistence Diagrams

The first who studied the stability of persistence diagrams (with respect to the Bottleneck distance) were Cohen-Steiner, Edelsbrunner, and Harer [Cha+09]. Subsequently, several stability theorems for persistent homology were found. The following theorem is the most relevant one for our purpose. In the following, we can imagine X and Y to be the metric spaces in which two samples of our data lie.

Theorem 2.5.3 (Stability of Persistence Diagrams) *Let X and Y be two compact metric spaces, $\mathrm{Filt}(X)$ and $\mathrm{Filt}(Y)$ Vietoris-Rips or Čech filtrations built on top X and Y, and let $\mathcal{D}\,\mathrm{Filt}(X)$ $\mathcal{D}\,\mathrm{Filt}(Y)$ be the corresponding persistence diagrams. Then, [CVJ21]*

$$d_B(\mathcal{D}\,\mathrm{Filt}(X)), \mathcal{D}\,\mathrm{Filt}(X))) \leq 2d_{GH}(X, Y). \tag{2.11}$$

This theorem states that the Bottleneck distance between the persistence diagrams of the filtrations on X and Y is at most twice the Gromov-Hausdorff distance between X and Y, and, therefore, establishes a relationship between a topological and a geometric distance.

We often call this theorem the "stability theorem" because it shows that small changes between two data samples result in small changes in the corresponding persistence diagram. The stability theorem implies that the persistence diagram computed from input data will not change drastically when perturbing the data (for example, with minor noise). This is important because it makes TDA a robust tool for analyzing data, which often contains noise.

Note. Theorem 2.5.3 is a stability result for the Vietoris-Rips (and Čech) filtration, which we will use for our EEG and EMG data. For the image data, we use radial filtration. Persistence diagrams from a radial filtration are less robust in practical applications than persistence diagrams from Vietoris-Rips filtrations [Tur+21].

Most stability results involve the Bottleneck distance between persistence diagrams, but there are also results using $p-$Wasserstein distances, $p < \infty$ [ST23]. Note that p-Wasserstein stability results, $p < \infty$, are weaker than stability results for the Bottleneck distance [BHG22].

Stability of Functional Summaries

In the next chapter, we will list *functional summaries*, mapping persistence diagrams to alternative representations or statistics. Many of these functional summaries are shown to be stable with respect to small perturbations in the original data D.

However, the stability of persistence diagrams does not guarantee the stability of all functional summaries. A functional summary could rely on features that can change dramatically with small perturbations of the persistence diagram, which would mean that the functional summary is not stable with respect to the persistence diagram. We will see that not all of the functional summaries we are using are stable.

The fact that we are using stable as well as unstable functional summaries might, however, not be a direct disadvantage for our classification task and could actually turn out to be advantageous. As mentioned in Section 5.4, including noise that might encode local geometric information into our analysis can turn out to be valuable. There is no particular reason to assume such noise to be less important than persistent points in machine learning tasks [CVJ21]. When only using stable functional summaries, we might fail to distinguish similar datasets on large scales and can better be differentiated on smaller scales [Ali+23]. Combining functional summaries that are sensitive to noise (local information) with globally stable summaries might be a good strategy.

3 Functional Summaries

As our data is of a relatively small size, which traditional machine learning models usually can handle better than deep learning models, we decided to use traditional machine learning models for our classification task. This poses a challenge when dealing with persistence diagrams, which are inherently (multi-)sets [Tau+20]. While some machine learning models can handle persistence diagrams directly [HMR21], traditional classifiers commonly require input data in tabular format [Ped+11].

Functional summaries are functions that map persistence diagrams to vectors [Tau+20], creating features that are processable by traditional machine learning models. No summary statistic captures all characteristics of a persistence diagram, and they might capture complementary characteristics. This chapter, therefore, introduces a variety of functional summaries collectively aimed at summarizing the topological characteristics of our data. To assess the contributions of individual summaries to our classification accuracy, we will compare their feature importance later in Section 6.2.

Some functional summaries map persistence diagrams directly to scalars or small vectors, so we can easily take the few coordinates of these vectors as (scalar) features for our traditional classifiers. We call them "basic summary statistics".

Other functional summaries, called "signatures", map the diagrams to large vectors. Using all such vector coordinates as features can entail computational inefficiency. Another problem is the risk of reduced accuracy, as a single vector coordinate might have a very small feature importance, and classification performance tends to suffer from a large number of insignificant features. Therefore, it can be advisable to compute statistics from these vectors to feed into the classifiers. We will compare the feature importance of coordinate features with the importance of such statistics in Section 6.2 to determine the most suitable approach.

3.1 Persistence Landscape

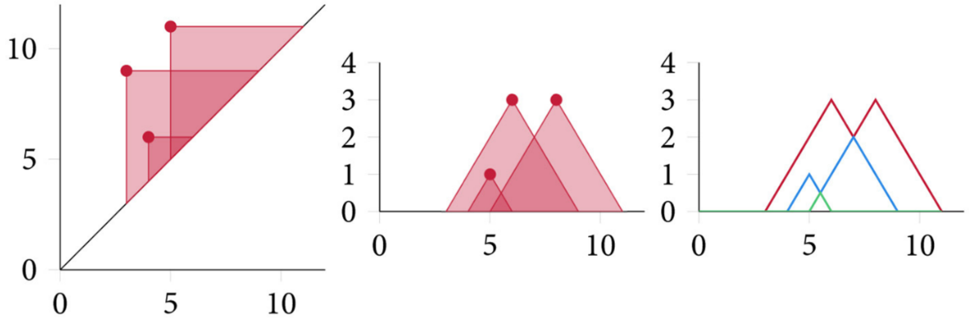

Figure 3.1 Construction of the persistence landscape [HMR21].

The persistence landscape, introduced by [Bub15], is a popular functional summary [Ber+20]. Rather than directly showing single persistent holes, the persistence landscape is a histogram-like summary capturing the distribution of topological features.

To construct a persistence landscape from a persistence diagram, we consider each homology dimension separately. As shown in Figure 3.1, for each point in the persistence diagram of one dimension, we then create an isosceles triangle with the point as one vertex and the other two vertices on the diagonal of the persistence diagram. [Ber+20]. We then rotate the subdiagram of the respective homology dimension by 45 degrees clockwise. The triangles now collectively form a set of triangle functions, and by taking the largest values of this triangle set, we derive the landscape function [Ber+20].

Definition 3.1.1 (Persistence Landscape) *Given a persistence diagram $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^{n}$, its persistence landscape is a sequence of functions $\{\lambda_k\}_{k\in\mathbb{N}}$, where each $\lambda_k : \mathbb{R} \to \overline{\mathbb{R}}$ is defined as follows [Bub15]:*

For each $k \in \mathbb{N}$, the function $\lambda_k(\epsilon)$ is the k-th largest value among the functions $\{\Lambda_i(\epsilon)\}_{i\in I}$, where

$$\Lambda_i(\epsilon) = [\min\{\epsilon - b_i, d_i - \epsilon\}]_+ \tag{3.1}$$

Here, $c_+ = \max(c, 0)$ and $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the extended real number line. The functions $\{\Lambda_i(\epsilon)\}_{i\in I}$ are called *triangle* functions because of their triangular shape.

We consider the $k-$th largest values of the set of triangles for *several k* to identify the most significant topological features at different levels of persistence. Note that k is not the homology dimension here.

Landscape Distance

The distance between the persistence landscapes of two persistence diagrams is called the "landscape distance" [Bub15]. If $\mathcal{D}, \mathcal{D}'$ denote two persistence diagrams and λ, λ' their respective persistence landscapes, the $p-$landscape distance ($p > 0$) between $\mathcal{D}, \mathcal{D}'$ is defined by [Bub15]

$$\Lambda_p\,(D, D') = \|\lambda - \lambda'\|_p, \tag{3.2}$$

where

$$\|\lambda\|_p^p = \sum_{k=1}^{\infty} \|\lambda_k\|_p^p\,. \tag{3.3}$$

Stability of Persistence Landscapes

The landscape supremum norm $\|\lambda\|_\infty$ is shown to be the same as $\frac{1}{2}\,\mathrm{pers}_\infty(D)$, where $\mathrm{pers}_\infty(D)$ is the largest persistence in a persistence diagram, and is bounded by the Bottleneck distance [Bub15]. Therefore, the persistence landscape is stable with respect to this supremum norm [Bub15]. This makes it a robust tool for data analysis.

3.2 Silhouette

Silhouettes are a variation of the persistence landscape and are also defined in terms of triangle functions $\Lambda_i(\epsilon)$. Giotto TDA uses power-weighted silhouettes of persistence diagrams [Tau+20], based on the idea of Chazal et al. [Cha+14].

Definition 3.2.1 (Silhouette) *Let $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^{n}$ be a persistence diagram and $w = \{w_i\}_{i=1}^{n}$ weights, given by $w_i = |d_i - b_i|^p$, $p \geq 1$, [Cha+14], where $\mid d_i - b_i \mid$ denotes the persistence of the i-th feature. The silhouette of $\mathcal{D}$ weighted by w is the function $\phi : \mathbb{R} \to \mathbb{R}$ defined by [Tau+20]*

$$\phi(t) = \frac{\sum_{i=1}^{n} w_i \Lambda_i(\epsilon)}{\sum_{i=1}^{n} w_i}, \tag{3.4}$$

where $\Lambda_i(\epsilon)$ are defined in 3.1.

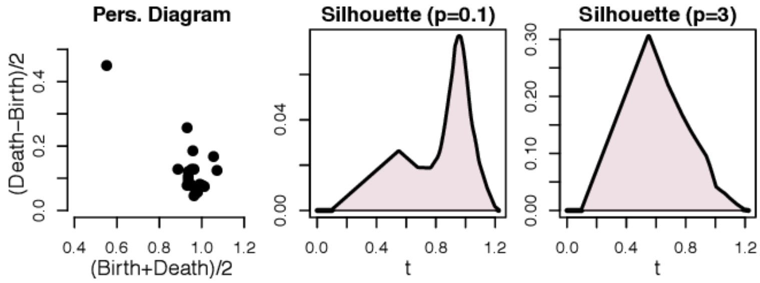

Figure 3.2 Exemplary power-weighted silhouettes for different choices of p, namely $p = 0.1$ and $p = 3$ [Cha+14].

The definition of silhouettes depends on a parameter p [Cha+14]. When p is large, the most persistent pairs are dominating the silhouette function [Cha+14]. When p is small, the points in the persistence diagram are treated more uniformly, or the less persistent holes are even more dominant [Cha+14]. The choice of p depends on what structures we are looking for in the data.

Figure 3.2 shows an example of power-weighted silhouettes of a persistence diagram (for one homology dimension). Like in constructing a persistence landscape, the first step to constructing a silhouette is to rotate the respective persistence diagram clockwise by 45 degrees [Cha+14]. Then, we construct triangle functions defined in 3.1, but instead of taking their largest values, we sum their weighted contributions [Cha+14].

Stability of the Silhouette

The silhouette is stable with respect to the Bottleneck distance [LS21].

3.3 Betti Curve

The Betti curve (also Betti sequence), originally defined in 2017 [Ume17] [JJ21], is a very simplified representation of a persistence diagram, which shows the number of features that are alive at a given filtration parameter threshold.

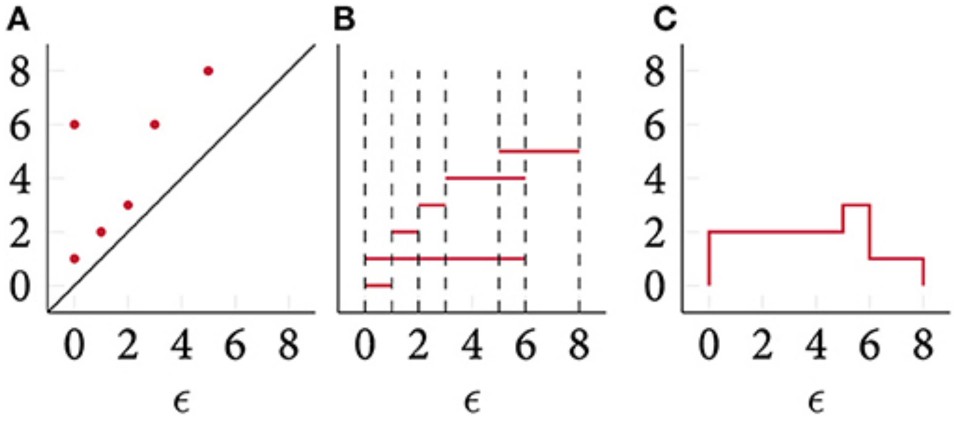

Figure 3.3 Construction of the Betti curve [HMR21]. Labels of the barcode axis are exluded.

Figure 3.3 shows how to construct the Betti curve for one homology dimension of a persistence diagram. First, we convert the persistence diagram into a “persistence barcode”. A persistence barcode is a representation in which each point becomes a horizontal line with the length of the point’s persistence [Ume17]. [Car+05]. Then, for each filtration parameter threshold ϵ, the number of lines that are “active” at ϵ is counted. These numbers define the value of the Betti sequence at ϵ [Ume17]. For example, we identify two active intervals when at $\epsilon = 0$ in Figure 3.3. At $\epsilon = 1$, we again find two intervals, with the 0−th interval already having died.

Definition 3.3.1 (Betti Curve) *Let $\mathcal{D} = \{(b_i, d_{ib})\}_{i=1}^{n}$ be a persistence diagram. The respective Betti curve is a function $\beta_{\mathcal{D}} : \mathbb{R} \longrightarrow \mathbb{N}$ whose values $\beta_{\mathcal{D}}, s \in \mathbb{R}$, are the amount of points (b_i, d_i) in $\mathcal{D}$ such that $b_i \leq s < d_i$ [Tau+20]. Here, we count with multiplicity [Tau+20], meaning that we consider how many times each point contributes to this amount.*

Stability of the Betti Curve

Johnson et al. showed that the (original) Betti curve is unstable with respect to the 1-Wasserstein metric [JJ21]. Similarly, it is unstable with respect to the Bottleneck distance [JJ21].

Recall that we noted in Section 2.5 that unstable functional summaries are particularly sensitive to small-scale holes (noise), and we might profit from topological noise in our analysis. Including unstable functional summaries like the Betti curve might therefore turn out valuable.

3.4 Heatkernel

We can apply "kernel-density estimation techniques" to persistence diagrams to estimate the distribution of the points within a persistence diagram. A kernel is defined as follows.

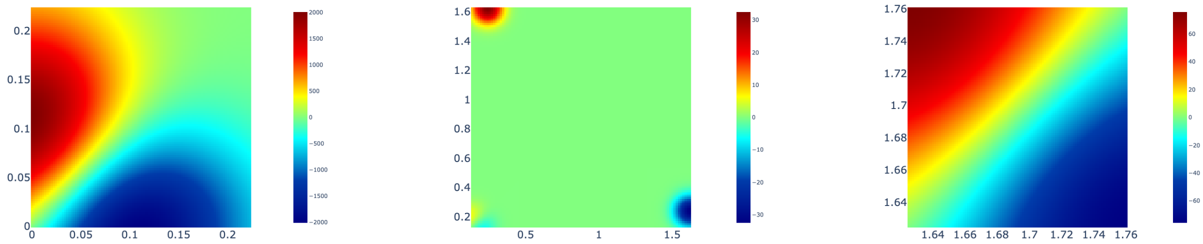

Figure 3.4 Heatkernel computed from data points sampled from a circle with noise. Homology dimension 0 (left), 1 (middle) and 2 (right).

Definition 3.4.1 (Kernel) *A kernel on a set A is a function [Tau+20]*

$$k : A \times A \longrightarrow \mathbb{R} \tag{3.5}$$

for which there exists a map ϕ from A to a vector space V, such that for $x, y \in A$,

$$k(x, y) = \langle \phi(x), \phi(y) \rangle. \tag{3.6}$$

[Rei+14] introduces a kernel-based functional summary that works by convolving persistence diagrams with a Gaussian kernel [Tau+20]. We will call their kernel-based functional summary "heatkernel". The idea is to use a kernel to assess the dissimilarity between two persistence diagrams $\mathcal{D}, \mathcal{D}'$ [Rei+14] .

The kernel presented [Rei+14] is a "Gaussian kernel" given by

$$k\left(\mathcal{D}, \mathcal{D}'\right) := \frac{1}{8\pi\sigma} \sum_{p \in \mathcal{D}, q \in \mathcal{D}'} \exp\left(-8^{-1}\sigma^{-1}\|p - q\|^2\right) - \exp\left(-8^{-1}\sigma^{-1}\|p - \bar{q}\|^2\right) \tag{3.7}$$

$$\phi(x) := \frac{1}{4\pi\sigma} \sum_{p \in \mathcal{D}} \exp\left(-4^{-1}\sigma^{-1}\|x - p\|^2\right) - \exp\left(-4^{-1}\sigma^{-1}\|x - \bar{p}\|^2\right), \tag{3.8}$$

where $\sigma > 0$ is called the scale, and $\bar{p}$ is the point you get when mirroring p on the diagonal [Rei+14].

Therefore, our Gaussian kernel is the sum of exponentially weighted distances between two points in two persistence diagrams $p \in \mathcal{D}$ and $q \in \mathcal{D}'$ [Rei+14].

To obtain a Gaussian kernel for *one* persistence diagram $\mathcal{D}$, we set $\mathcal{D}'$ to be the *diagonal diagram* [Rei+14], consisting of only zero-persistence holes, and measure the dissimilarity of $\mathcal{D}$ to this diagonal diagram.

Computing the "heatkernel" functional summary involves convolving the Gaussian kernel of a persistence diagram with the original persistence diagram. To be exact, we convolve the Gaussian kernel with the "Dirac delta distribution" of the persistence diagram (for each homology dimension separately), which results from mapping each point p in $\mathcal{D}$ to a *Dirac delta function* $\delta(x - p)$ [Tau+20]. Dirac delta functions are zero everywhere except at the point p, where they are infinite, such that they integrate to 1 over their domain [Tau+20].

After performing such a convolution, we perform the same convolution with mirrored subdiagrams reflected about the diagonal [Tau+20].

These convolutions smooth out the discrete points p in the persistence diagram, leading to an impressive visualization of the diagram and a summary that is quite robust as it tends to smooth over noisy data, reducing the impact of outliers or small, non-robust features [Rei+14]. Figure 3.4 shows an example of a heatkernel, computed from data sampled from a circle with noise, which is shown in Figure 2.12 (The corresponding persistence diagram is shown in Figure 2.16).

We used the software Giotto TDA to compute the heatkernels. For computational efficiency, Giotto TDA does not use the complete persistence diagrams, but only "locations evenly sampled from appropriate ranges of the filtration parameter" [Tau+20].

Stability of the Heatkernel

The authors of [Rei+14] show that their Gaussian kernel is stable with respect to the 1−Wasserstein distance.

3.5 Persistence Images

Persistence images are another kernel-based functional summary, which was introduced by [Ada+17]. Figure 3.5 shows the construction of a persistence image. After rotating a persistence diagram $\mathcal{D}$ clockwise by 45 degrees, we map it to an integrable function $\rho_{\mathcal{D}} : \mathbb{R}^2 \to \mathbb{R}$, which we call *persistence surface* [Ada+17]. We construct this surface as the sum of weighted Gaussian kernels centered at every point of the persistence diagram [Ada+17].

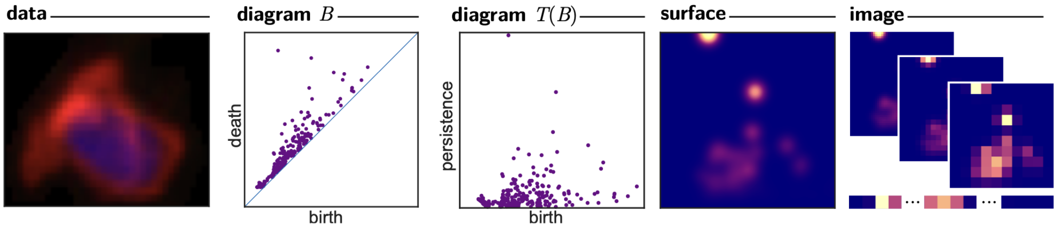

Figure 3.5 Construction of the persistence image [Ada+17].

We then construct a grid on the persistence surface. Finally, we compute the integral of ρ_B within each grid box to compute the persistence image cell [Ada+17]. The resulting persistence image is a “vectorized” representation of the persistence diagram [Ada+17].

The Gaussian kernels we center around each point (b_i, d_i) in a persistence diagram $\mathcal{D}$ are given by [Ada+17]

$$k(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left[(x-b_i)^2+(y-d_i)^2\right]/2\sigma^2}. \tag{3.9}$$

The persistence surface is as a weighted combination of these kernels [Ada+17]:

$$\rho_{\mathcal{D}}(x, y) = \sum_{u \in T(\mathcal{D})} f(u) k(x, y), \tag{3.10}$$

where $T(\mathcal{D})$ is the rotated persistence diagram, and the $f(u)$ are weighting functions defined as [Ada+17]

$$f(u) = \begin{cases} 0 & \text{if } u \leq 0 \\ \frac{u}{b} & \text{if } 0 < u < b, \text{ and } \\ 1 & \text{if } u \geq b \end{cases}. \tag{3.11}$$

Here, b is the persistence value of the most persistent feature [Ada+17].

After creating a grid on the surface, we compute the persistence image value of each pixel p (which are squares in the grid) as the persistence surface over the p) [Ada+17]:

Definition 3.5.1 (Persistence Image) *For a persistence diagram $\mathcal{D}$, a persistence image is the collection of pixel values $I\left(\rho_{\mathcal{D}}\right)_p$,*

$$I\left(\rho_{\mathcal{D}}\right)_p = \iint_p \rho_{\mathcal{D}}(x, y) dy dx. \tag{3.12}$$

Stability of Persistence Images

Adams et al. prove the stability of the persistence image with respect to the 1-Wasserstein distance [Ada+17].

3.6 Amplitude

Amplitude is a basic functional summary introduced by [Giu+24].

Let A be the set of persistence subdiagrams of each homology dimension. We can vectorize A using a vectorization $\phi : A \longrightarrow V$ with V being a normed space [Tau+20].

Generally, an amplitude is a function $f : A \longrightarrow \mathbb{R}$ on A, for which $f(x) = \|\phi(x)\|$ for all $x \in X$ [Tau+20].

In persistent homology, this norm $\|\phi(x)\|$ is often constructed by the Wasserstein, Bottleneck, or Landscape metric [Tau+20].

For instance, for a persistence diagram $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^n$, the amplitude based on the $p-$Wasserstein distance, the *Wasserstein amplitude* of order p, is the L^p norm of the vector of point distances to the diagonal [Tau+20]:

$$A_W = \frac{\sqrt{2}}{2} \left(\sum_i (d_i - bi)^p \right)^{\frac{1}{p}}. \tag{3.13}$$

The *Landscape amplitude* of $\mathcal{D}$ is essentially the landscape distance between D and the diagonal diagram, which contains only the diagonal points [Tau+20].

Stability of the Amplitude

Persistence diagrams are stable with respect to the Wasserstein and Bottleneck distances, meaning that the Wasserstein and the Bottleneck amplitudes are also stable. The Landscape amplitude is stable, as well [Bub15].

3.7 Number of Points

The *Number of Points* in a persistence diagram is also a quantity that can turn out helpful in data analysis [Tau+20]. Note that they are not the same as Betti numbers. While Betti numbers refer to the number of holes in the underlying manifold of the data, the number of points in a persistence diagram is the number of distinct holes in a filtration, including very small-scale holes.

Stability of the Number of Points

As persistence diagrams are stable under small perturbations [CSEH07] to the Bottleneck and other Wasserstein distances, so is the number of points [KK21].

3.8 Persistence of the Most Prominent Point

We additionally included the persistence of the point with the largest persistence as a feature in our analysis.

Stability of the Most Prominent Point

The persistence of the most prominent point is stable with respect to the Bottleneck and other Wasserstein distances[KK21].

3.9 Persistence Entropy

We can quantify the degree of diversity of points in a persistence diagram by a functional summary called *persistence entropy*, introduced by [AGDR19]. The persistent entropy PE of a persistence diagram $\mathcal{D}$ is computed using the formula [KK21]:

$$PE(\mathcal{D}) := \sum_{x \in \mathcal{D}} -\frac{l_x}{L_{\mathcal{D}}} \ln\left(\frac{l_x}{L_D}\right). \tag{3.14}$$

Here, $L_{\mathcal{D}} := \sum_{x \in \mathcal{D}} l_x$ is the sum of the lifetimes of all points in the diagram. If the diagram only consists of the diagonal, the persistent entropy is considered zero [KK21].

Stability of the Persistence Entropy

The authors of [AGDST18] find that persistence entropy is stable with respect to the Bottleneck and Wasserstein distance.

3.10 Complex Polynomials

Complex polynomials are a feature in Giotto TDA [Tau+20] based on [DFF15], which in turn is based on an idea of Landi [Fer99] [DFF15]. Her idea was to encode a persistence diagram using coefficients of complex polynomials that have the points in the persistence diagrams as roots [DFF15].

Definition 3.10.1 (Complex Polynomial Vectorization) *Let $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^{n}$ be a persistence diagram, and let m_i denote the multiplicity for a respective birth-death pair (b_i, d_i), meaning that there are m features that are all born at b_i and die at d_i [KMN17]. For a given function $R : \mathbb{R}^2 \longrightarrow \mathbb{C}$, the complex polynomial for is given by [Ali+23] [DFF15]*

$$C(z) := \prod_{(b_i,d_i)\in D} [z - R(b_i, d_i)]^{m_i}. \tag{3.15}$$

Giotto TDA gives three options for the function R, one of which is the continuous function $R : \mathbb{R}^2 \longrightarrow \mathbb{C}$ [Tau+20],

$$R(x, y) = x + iy. \tag{3.16}$$

The final vector we can use for classification consists of the first few highest degree coefficients of $C(z)$ [Tau+20].

Stability of Complex Polynomials

The authors of [DFF15] do not prove stability for their complex polynomials.

3.11 ATOL vectorization

The adaptive topology-oriented learning (ATOL) vectorization is a feature that Giotto TDA does not provide [Tau+20], but an alternative Python package, GUDHI [The15], does. The vectorization is based on [Roy+20].

Constructing an ATOL vectorization involves two steps. Let $\mathcal{D}_1$, $\mathcal{D}_2$, ..., $\mathcal{D}_n$ be a set of persistence diagrams. The first step of the ATOL algorithm involves clustering $\mathcal{D}_1$, $\mathcal{D}_2$, ..., $\mathcal{D}_n$ into b clusters (for instance, with KMeans) [The15]. We then compute the empirical mean measure $\bar{\mathcal{D}}_i$ for each cluster c_i, $i = 1, ..., b$. $\bar{\mathcal{D}}_i$ is the representative center for cluster c_i obtained through the clustering process [The15]. We refer to this step as the "quantization step" [Roy+20].

Next, we define b *contrast functions*, which compute the proximity of a persistence diagram to the cluster centers [The15].

Definition 3.11.1 (Contrast Functions) *Let $\mathcal{D}$ be a persistence diagram. For cluster centers c_i, $i = 1, ..., b$, we define a scaling factor [Roy+20]*

$$\sigma_i := \frac{1}{2} \min_{j\in[b], j\neq i} \left\|c_i - c_j\right\|_2 . \tag{3.17}$$

The contrast functions

$$\left\{\Omega_i : \mathbb{R}^2 \to \mathbb{R} \mid 1 \leq i \leq b\right\} \tag{3.18}$$

are defined as [Roy+20]

$$\Omega_i(D) = \exp\left(-\frac{\|\mathcal{D} - z_i\|_2}{\sigma_i}\right). \tag{3.19}$$

Our b-dimensional ATOL vectorization of persistence diagram $\mathcal{D}$ is now given by [Roy+20]

$$(\Omega_1(\mathcal{D}), \ldots, \Omega_b(\mathcal{D})) . \tag{3.20}$$

Stability of the ATOL vectorization

In [Roy+20], Chazal et al. show that ATOL vectorizations are stable with respect to the 1-Wasserstein metric.

3.12 Adcock-Carlsson Coordinates

In their influential paper "The ring of algebraic functions on persistence barcodes" [ACC16], Adcock, Carlsson, and Carlsson defined a summary of persistence barcodes which will prove to have a pretty significant impact on our classification accuracy.

In [BPP21], the authors used the same features on persistence diagrams, calling them "Adcock-Carlsson coordinates". In their paper, Adcock-Carlsson coordinates are among the most accurate and efficient features for a variety of tasks on a set of five datasets.

The specific Adcock-Carlsson coordinates $(c_1, c_2, c_3, c_4)^T$ suggested by [ACC16] and chosen by [BPP21] are defined as follows.

Definition 3.12.1 (Adcock-Carlsson Coordinates) *Given a persistence diagram $\mathcal{D} = \{(b_i, d_i)\}_{i=1}^n$, the vector $(c_1, c_2, c_3, c_4)^T$ is defined by*

$$c_1 = \sum_{b_i} b_i \left(d_i - b_i\right) \tag{3.21}$$

$$c_2 = \sum_i \left(y_{\max} - d_i\right)\left(d_i - b_i\right) \tag{3.22}$$

$$c_3 = \sum_i b_i^2 \left(d_i - b_i\right)^4 \tag{3.23}$$

$$c_4 = \sum_i \left(d_{\max} - d_i\right)^2 \left(d_i - b_i\right)^4, \tag{3.24}$$

where $d_{\max}$ denotes the latest death [BPP21].

Stability of Adcock-Carlsson coordinates

According to [Ali+23], changes in a persistence diagram (in terms of Bottleneck distance) can lead to large fluctuations in the corresponding Adcock-Carlsson coordinates. This makes Adcock-Carlsson a relatively unstable functional summary, which does not mean they cannot be helpful for our purposes. If one wanted to find an alternative to Adcock-Carlsson coordinates that addresses the problem of Bottleneck instability, one could use *tropical coordinates* [Ali+23].

4 Data Exploration

Before feeding our functional summaries into a machine learning model, we want to explore them visually and see if we can detect patterns without the help of machine learning. Our method for computing the persistence diagrams, which is a necessary step before computing functional summaries, is described in Section 5.

4.1 Data

We will conduct all our experiments on mouse brain data, but the same or similar methods could be applied to human brain data. Our mouse brain data consists of *EEG/EMG data* and *Calcium brain imaging data.*

The data we use for this master thesis consists of two datasets, and we perform two different classification tasks on them. The first dataset, the "Anesthesia Data", is multimodal brain data consisting of EEG/EMG and brain imaging data. The data was recorded on five different mice, put under five different levels of anesthesia, and recorded for five minutes during each of these levels.

We encoded the anesthesia levels as numerical *labels* for our analysis. Label 0 means 1% of anesthesia gas, label 1 means 1.2%, label 2 means 1.4%, label 3 means 1.6%, and label 4 means 1.8%.

The second dataset, the "Sleep Data", solely consists of EEG/EMG data, recorded on three subjects, which were recorded for about 2 hours each and observed regarding their sleeping behavior. Label 1 stands for "Awake", label 3 for "Non-REM Sleep", label 5 for "REM Sleep" and label 2, 5 and 7 are different types of artifacts.

Electroencephalography (EEG) and Electromyography (EMG) Data

Electroencephalography (EEG) is a commonly used technique in neuroscience, offering insights into the brain by capturing and analyzing its electrical activity in a multivariate or univariate time series [CA08].

For measuring electrical activity in the brain, multiple electrodes are placed on the scalp. EEG detects brain activity by measuring the voltage between pairs of electrodes [Hat+23]. It detects the activity of large groups of neurons primarily in areas surrounding the electrodes [CA08].

In our analysis, we positioned only two electrodes on the mouse scalps. One electrode is positioned at the top of the left hemisphere of the mouse scalp, while we placed another at the bottom of the right hemisphere. The potential difference between these two electrodes is recorded, amplified, digitized, and stored [CA08].

The resulting EEG data is a univariate time series with a very high temporal resolution, showing the brain activity in the order of milliseconds.

In addition to EEG data, we record Electromyography (EMG) data. EMG records the activity of muscles [BD06], which can help to distinguish sleep and anesthesia states, as breathing activity is lower in deeper anesthesia or sleep stages.

Figure 4.1 shows parts of the raw EEG Sleep data of different labels. As you can see, Non-REM sleep and the Awake state look relatively quite similar, while the amplitude during Non-REM Sleep is more diverse.

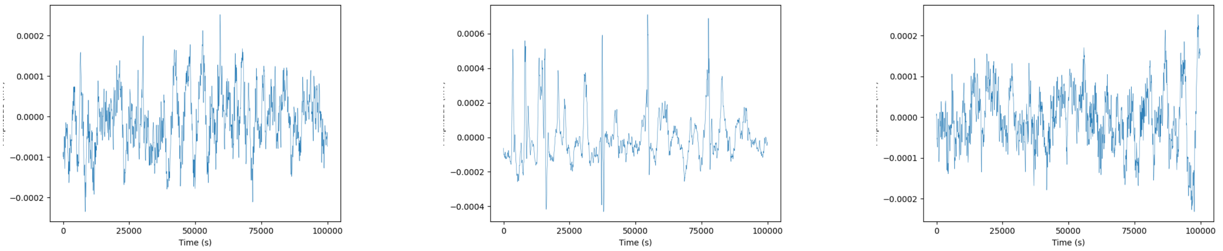

Figure 4.1 EEG data from the Sleep dataset. Awake (left), Non-REM (middle) and REM Sleep (left).

Figure 4.2 shows EEG and EMG Anesthesia data recordings for one subject (subject “m292”), while the subject was put under a low level of anesthesia for five minutes, while Figure 4.3 shows similar recordings for a high level of anesthesia.

The most apparent remark that can be made from Figure 4.2 and Figure 4.3 is that the EEG recordings seem to have (at least partly) higher peaks under a higher level of anesthesia. However, the amplitude also seems to vary more over time under a higher level of anesthesia.

As we can see, the EMG data in Figures 4.2 and 4.3 has a low resolution. On the figures, we observe that muscle movement is smaller in higher levels of anesthesia.

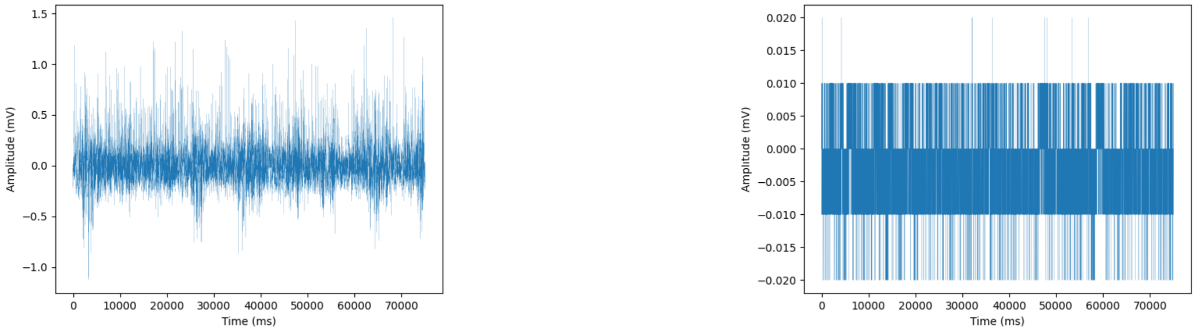

Figure 4.2 EEG (left) and EMG (right) data for subject m292 under the weakest level of anesthesia.

Brain Imaging Data

EEG and EMG recordings have a high temporal resolution. However, using only two electrodes does not capture any spatial information. Therefore, we additionally want to record the brain by filming it, even though videos usually have a lower temporal resolution than EEG recordings.

We call our method to record video data of the mouse brains *brain imaging*. When speaking of brain imaging, researchers can mean different methods, including fMRI in the three-dimensional domain [Glo11]. In our analysis, brain imaging recordings refer to (two-dimensional) videos of the mouse brain from above. Figure 4.4 shows an exemplary frame of such a recording.

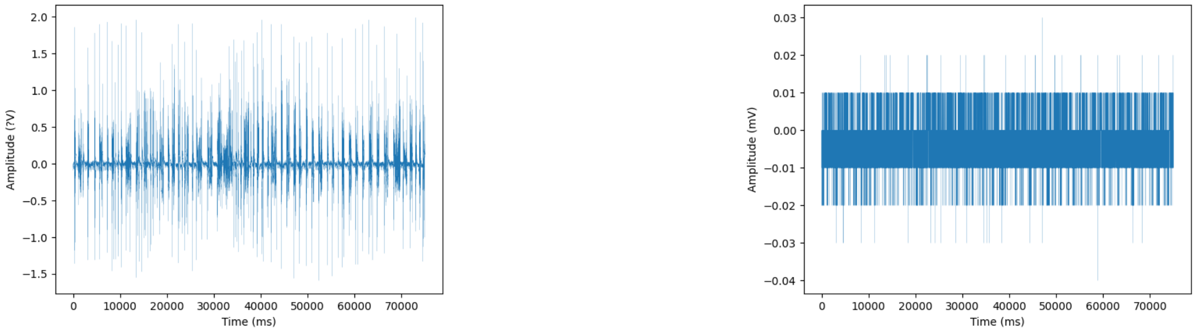

Figure 4.3 EEG (left) and EMG (right) data for subject m292 under the highest level of anesthesia.

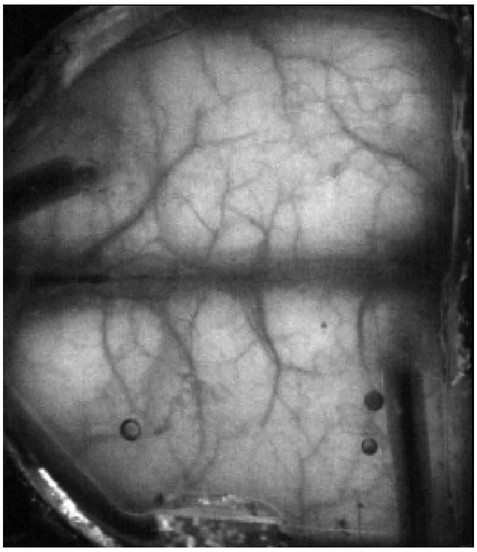

Figure 4.4 Example frame of brain imaging recordings for subject m292.

Our method involves tracking calcium levels in neurons. After neuronal activity, there is a strong response in calcium in the brain [Che+13], such that tracking calcium levels is, with some delay, a measure of neuronal activity. We make calcium levels visible to the bare eye with fluorescent calcium sensors in the mouse brain.

The method we use is based on a procedure described in [Che+13], which uses *GCaMPs* as calcium sensors. GCaMPs refer to a family of ultrasensitive protein calcium sensors [Che+13]; we use GCaMP*6m*.

GCaMPs are fluorescent proteins that indicate the calcium level in neurons and, therefore, their activity. A chemical reaction between GCaMPs and calcium causes GCaMPs to fluoresce with green light when irradiated with blue light [Che+13]. By observing the green irradiated green light, we can observe calcium levels with our bare eyes and, therefore, also indirectly observe neuronal activity.

GCaMPs are a type of genetically encoded calcium indicators (GECIs) [Che+13], which means that we do not have to inject the proteins directly into the mouse brain; instead, the mouse's genes are manipulated to contain the genetic code for the fluorescent proteins [OLK19].

We genetically manipulated the mice using structure-based mutagenesis. This technique involves introducing the DNA encoding GCaMP6 into the mice via a virus that infects neurons [Che+13].

4.2 Visualizing Persistence Diagrams

Wasserstein Barycenters

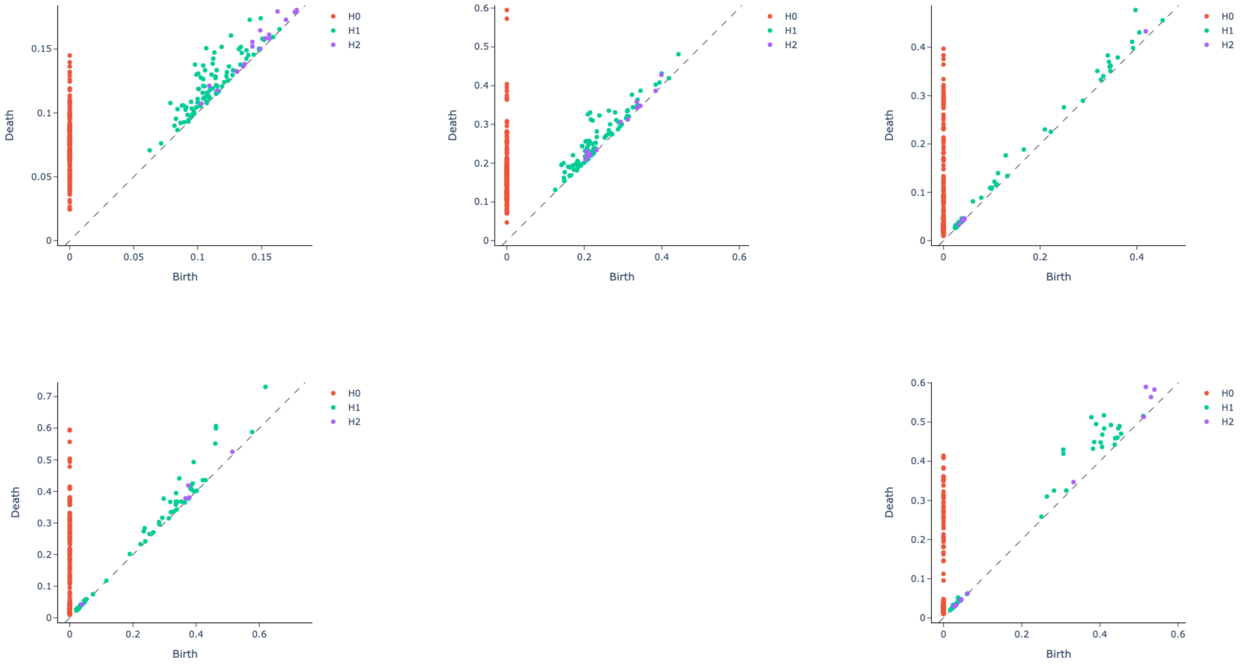

Figure 4.5 Wasserstein Barycenters of EEG persistence diagrams, subject m292. Label 0 on the top left, followed by label 1, 2, 3 reading from left to right, and label 4 at the bottom right.

For the Anesthesia data, we created persistence diagrams for *each 4-second interval* of the EEG/EMG data and for each frame of the brain imaging data. Instead of visualizing all of them here, we computed the *Wasserstein barycenters* of all persistence diagrams for each modality and label. The Wasserstein barycenter of a collection of persistence diagrams can be seen as the most representative diagram, as is the persistence diagram with the smallest average Wasserstein distance to all other persistence diagrams [ABA22].

Figure 4.5 shows the Wasserstein barycenters for the EEG data recorded on subject m292. As can be seen, none of the persistence diagrams has holes of a dimension higher than 2. For higher labels, there seem to be less holes in general.

Figure 4.6 shows the Wasserstein barycenters for the brain imaging data recorded on subject m292. The differences between the labels are less apparent here, but again we observe that there are no holes of dimension higher than 2.

To compute the Wasserstein barycenters for the brain imaging data, we only used a randomyl selected portion of each segment in the data instead of the complete data due to computational efficiency, such that the brain imaging Wasserstein barycenters are only partly representative. For each segment, we then computed the Wasserstein barycenter in each segment first, and the overall Wasserstein barycenter as the most representative one out of these segment Wasserstein barycenters.

Dissimilarity Matrices

Figure 4.7 shows a dissimilarity matrix depicting the Wasserstein distances between the Wasserstein barycenters of the different labels, using the EEG Anesthesia data of one subject. Figure 4.8 shows the

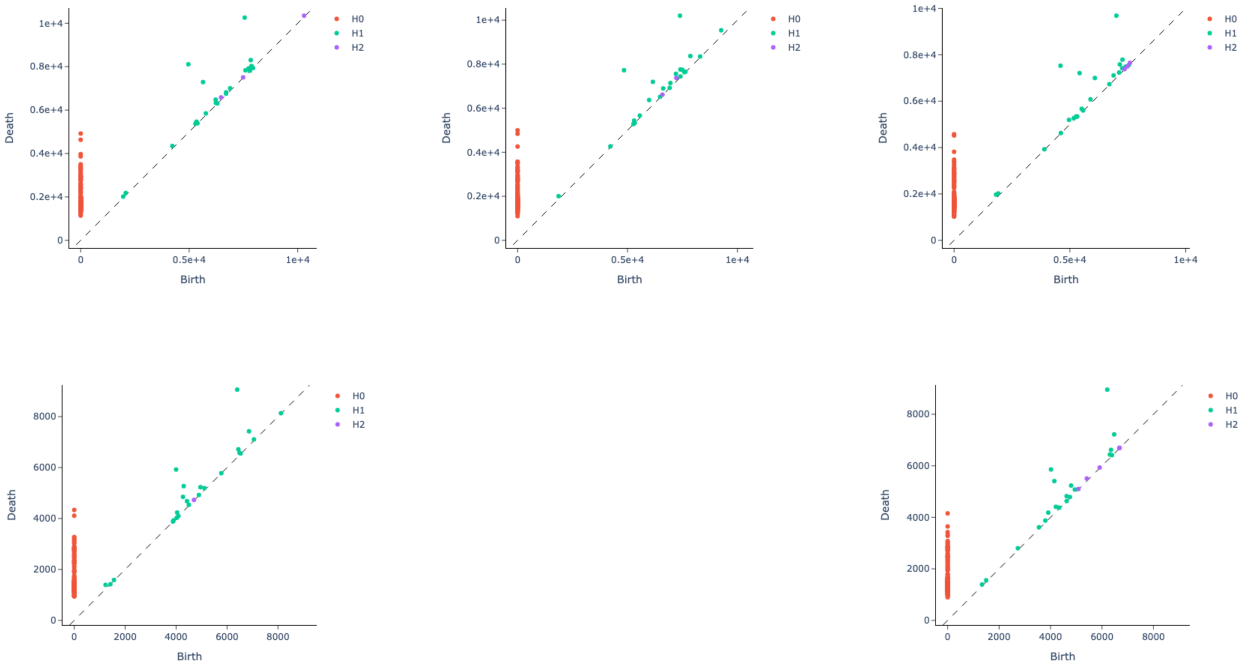

Figure 4.6 Wasserstein Barycenters of brain imaging persistence diagrams, subject m292. Label 0 on the top left, followed by label 1, 2, 3 reading from left to right, and label 4 at the bottom right.

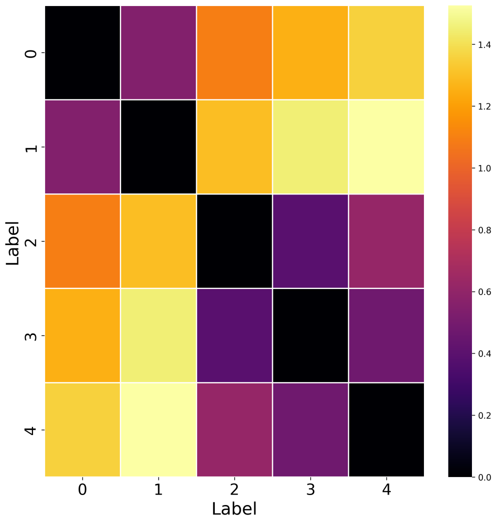

Figure 4.7 Dissimilarity matrix of the Wasserstein barycenters of the different labels for the EEG data, subject m292. Dissimilarity is measured by the Wasserstein metric.

same for brain imaging data. It is not surprising that the respective Wasserstein barycenters tend to differ more for higher differences in the depth of anesthesia.

4.3 Visualizing Basic Summary Statistics

Figure 4.9-4.12 show box plots of the distributions of the basic summary statistics *amplitude* and *number of points* for EEG and brain imaging data.

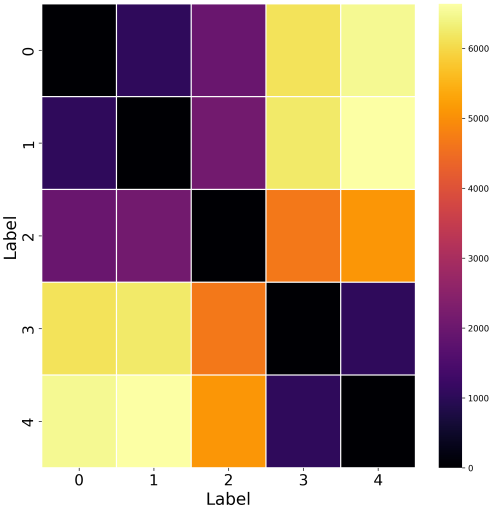

Figure 4.8 Dissimilarity matrix of the Wasserstein barycenters of the different labels for the brain imaging data, subject m292. Dissimilarity is measured by the Wasserstein metric.

For both data modalities, amplitude increases for some homology dimensions with an increasing depth of anesthesia and decreases for other homology dimensions. The number of points mostly decreases with an increasing depth of anesthesia for EEG data and increases when the depth of anesthesia increases for brain imaging data.

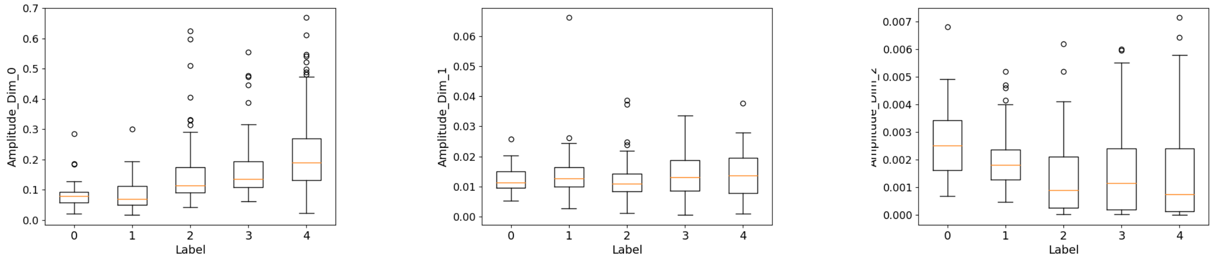

Figure 4.9 Box plots for the amplitude over different levels (EEG).

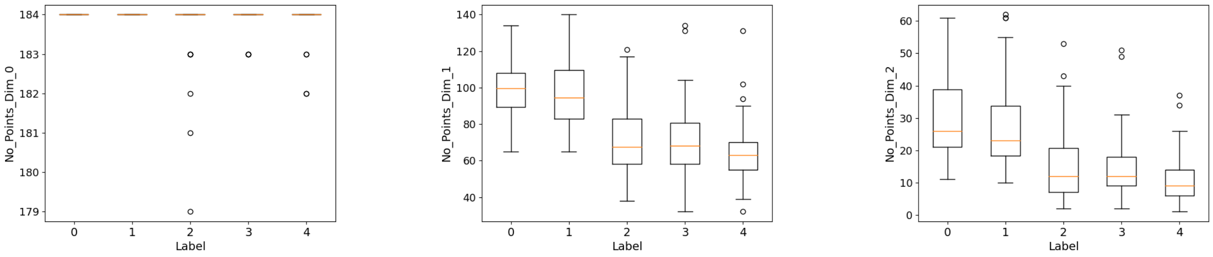

Figure 4.10 Box plots for the number of points over different levels (EEG).

4.4 Visualizing Signatures

Giotto TDA contains in-built functions to visualize signatures [Tau+20], based on methods in the Python package plotly [Inc15]. With these functions, we visualized signatures of the Wasserstein barycenters of the anesthesia data, trying to visually discover patterns that distinguish low anesthesia (label 0) from high

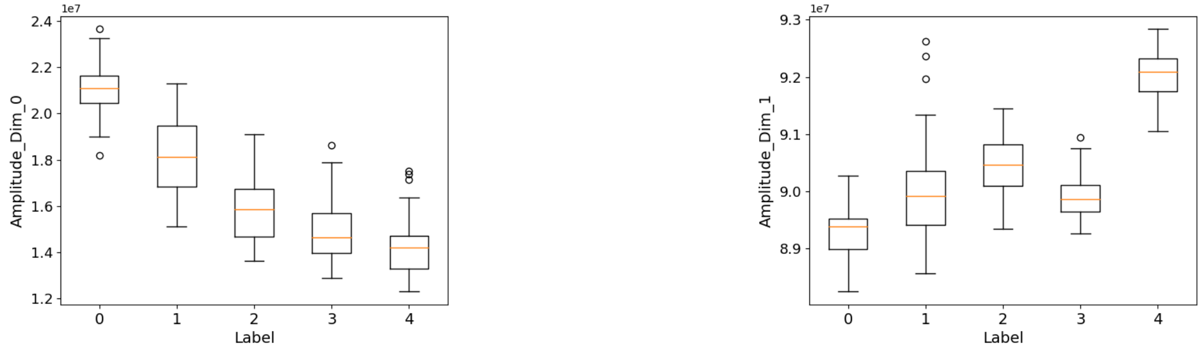

Figure 4.11 Box plots for the amplitude over different levels (brain imaging).

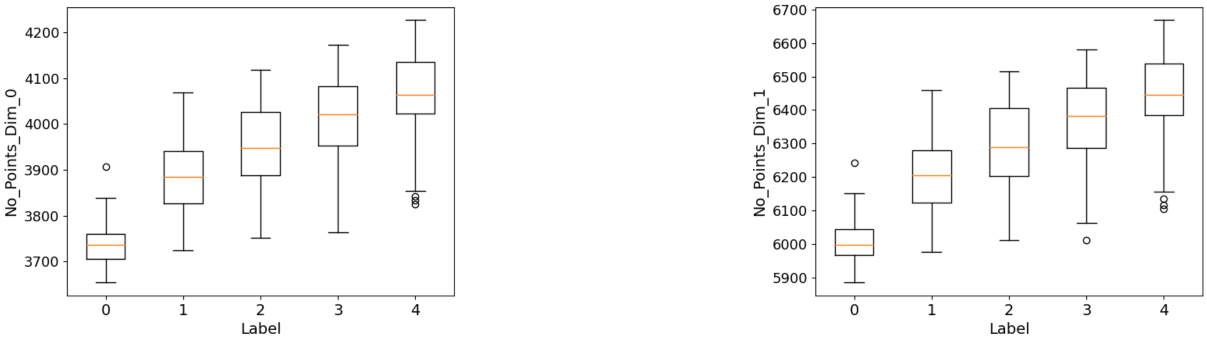

Figure 4.12 Box plots for the number of points over different levels (brain imaging).

anesthesia (label 4). Signatures of persistence diagrams can be more suitable for distinguishing than the persistence diagrams themselves, as we will see, as persistence diagrams can not be easy to grasp visually.

This chapter shows a selection of signatures, while signatures that are not shown in this chapter can be found in the Appendix. We adapted some plotly parameters in the Giotto TDA functions for creating the visualizations.

Persistence Landscapes

Figure 4.13 compares the persistence landscapes of the Wasserstein barycenter of label 0 to that of label 4 for the EEG Anesthesia data.

The persistence in dimension 0 peaks at around $\epsilon = 0.15$ for label 0 and at around $\epsilon = 0.4$ for label 4. The higher and broader peak in label 4 suggests that the data under deep anesthesia has lasting connected components, which reflected in the persistence diagrams in Figure 4.5. Also for homology dimension 1 and 2 (but especially for dimension 1), deeper anesthesia leads to more significant topological loops.

Figure 4.14 shows the persistence landscape of the brain imaging Wasserstein barycenters of the lowest and highest level of anesthesia.

The difference of the persistence landscape in homology dimension 0 is rather subtle. In homology dimension 1, persistence peaks at smaller filtration parameter values in the lowest level of anesthesia than in the highest level of anesthesia ($\epsilon \approx 6500$ and $\epsilon \approx 8700$ vs. $\epsilon \approx 5000$ and $\epsilon \approx 8000$).

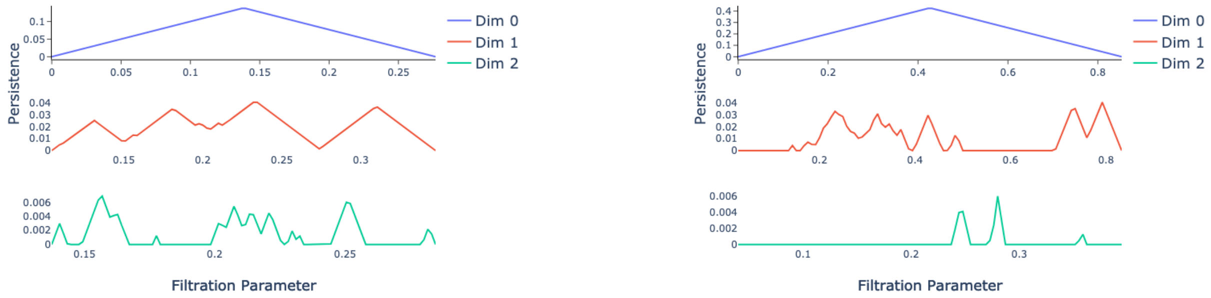

Figure 4.13 Persistence landscapes computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Persistence landscape for lowest level of anesthesia on the left, persistence landscape for highest level of anesthesia on the right.

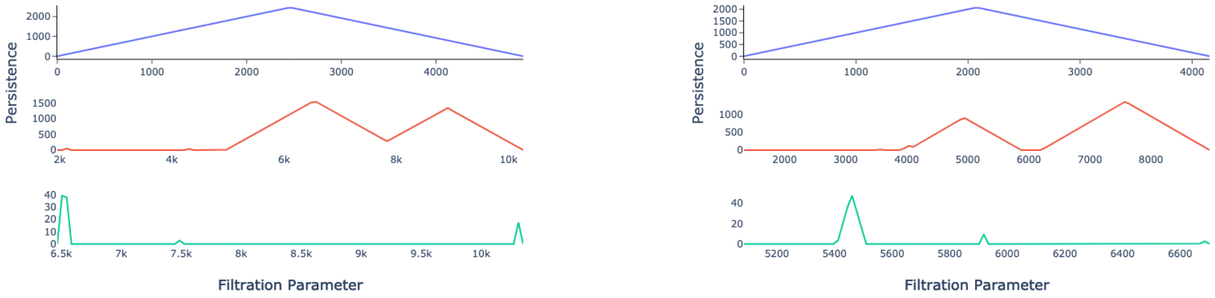

Figure 4.14 Persistence landscapes computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Persistence landscape for lowest level of anesthesia on the left, persistence landscape for highest level of anesthesia on the right.

Betti Curve

Figure 4.15 compares the Betti curves of the Wasserstein barycenters of label 0 to label 4 for the EEG Anesthesia data.

In homology dimension 0, the Betti curve for both labels starts high and then decreases when the filtration parameter ϵ increases. It decreases slower in label 4, however, indicating that the 0-dimensional holes of label 4 are, on average, more persistent. Homology dimension 1 and 2 are hard to visually compare in this depiction due to the scale.

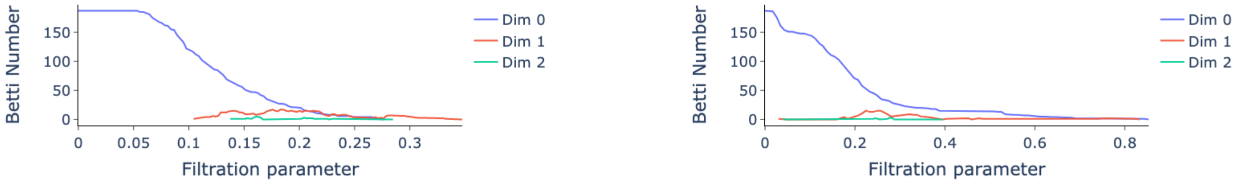

Figure 4.15 Betti curve computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Betti curve for lowest level of anesthesia on the left, Betti curve for highest level of anesthesia on the right.

Figure 4.16 compares the Betti curves of the Wasserstein barycenters of label 0 to label 4 for the brain imaging Anesthesia data. In contrast to what we observed for the EEG data, the Betti curve of homology dimension decreases slightly earlier for label 4 than for label 0. This indicates that there are more persistent 0-dimensional holes for label 0 than for label 4.

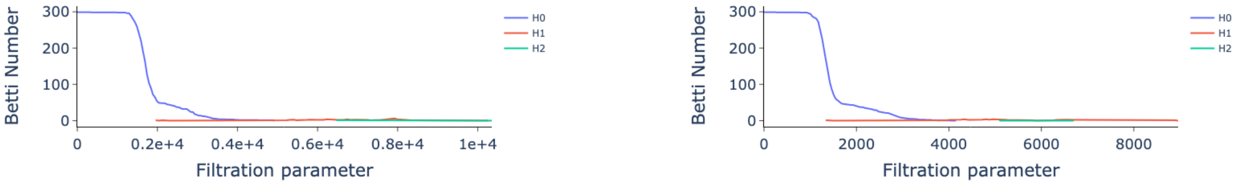

Figure 4.16 Betti curve computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Betti curve for lowest level of anesthesia on the left, Betti curve for highest level of anesthesia on the right.

Heatkernels

Figure 4.17 shows the heatkernels of homology dimension of the Wasserstein barycenters of label 0 and 4 for the EEG Anesthesia data. Homology dimensions 1 and 2 are shown in the Appendix in Figure A.3 and A.4.

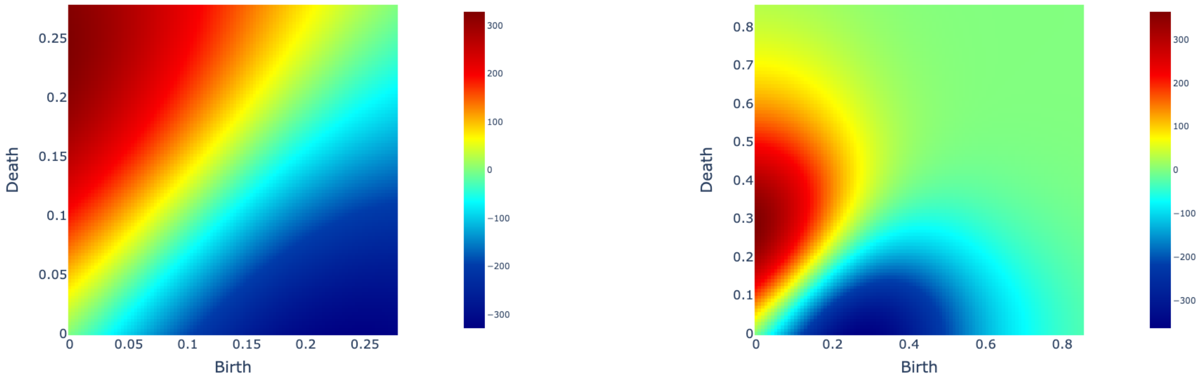

Figure 4.17 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

Figure 4.18 shows the heatkernels of homology dimension of the Wasserstein barycenters of label 0 and 4 for the brain imaging Anesthesia data. Homology dimensions 1 and 2 are shown in the Appendix in Figure A.5 and A.6.

4.5 Searching Clusters in Functional Summaries

It is an interesting question to ask whether the direct embedding vectors representing the signatures show clear clusters which represent the different label. To be able to answer this question visually, we reduce the dimension of each embedding to 3, such that we can depict all signature embedding in a 3-dimensional vector space.

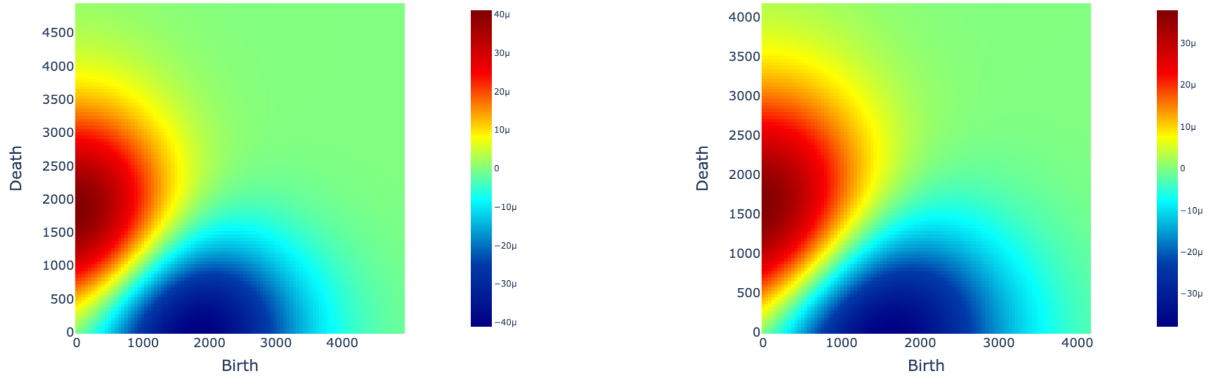

Figure 4.18 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

Depending on the nature of the signature, we either apply UMAP [MHM20] or Kernel PCA [SSM05], two common dimensionality reduction algorithms, to reduce the dimension of the signature; we apply Kernel PCA to kernel-based signatures, and UMAP to the remaining signatures.

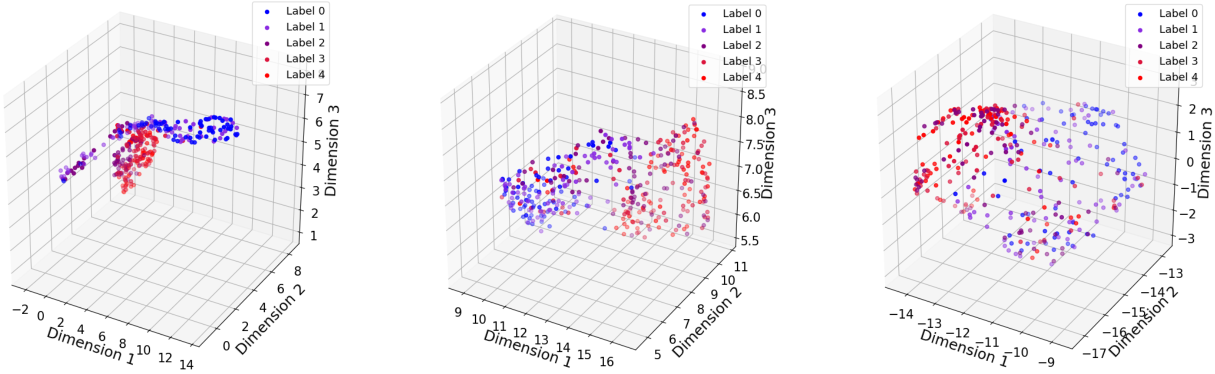

Figure 4.19 Dimensionality reduced Betti curve embeddings for EEG data, subject m292. Homology dimension 0 (left), 1 (middle) and 2 (right).

Some exemplary results for subject m292 are shown in Figure 4.19, which shows the dimensionality reduced Betti curve of each homology dimension for EEG data, and Figure 4.20, which shows the dimensionality reduced Betti curve of each dimension for brain imaging data.

The figures are representative of other signatures, as well. All (dimensionality-reduced) signatures do not show clear clusters. Instead, they often show a gradual change along the labels. The points appear more separable for brain imaging data, whereas some EEG signatures show almost no pattern.

We additionally search for clusters in further functional summaries that do not require dimensionality reduction to be visible, such as Adcock-Carlsson coordinates. Figure 4.21 depicts the distribution of the four Adcock-Carlsson coordinates for the brain imaging data of subject m292 pairwise in two scatter plots. At least for this subject, we can see clear clusters, which is an interesting result. Other summary statistics, as exemplary shown in Figure 4.22, show no clusters but a gradual change instead.

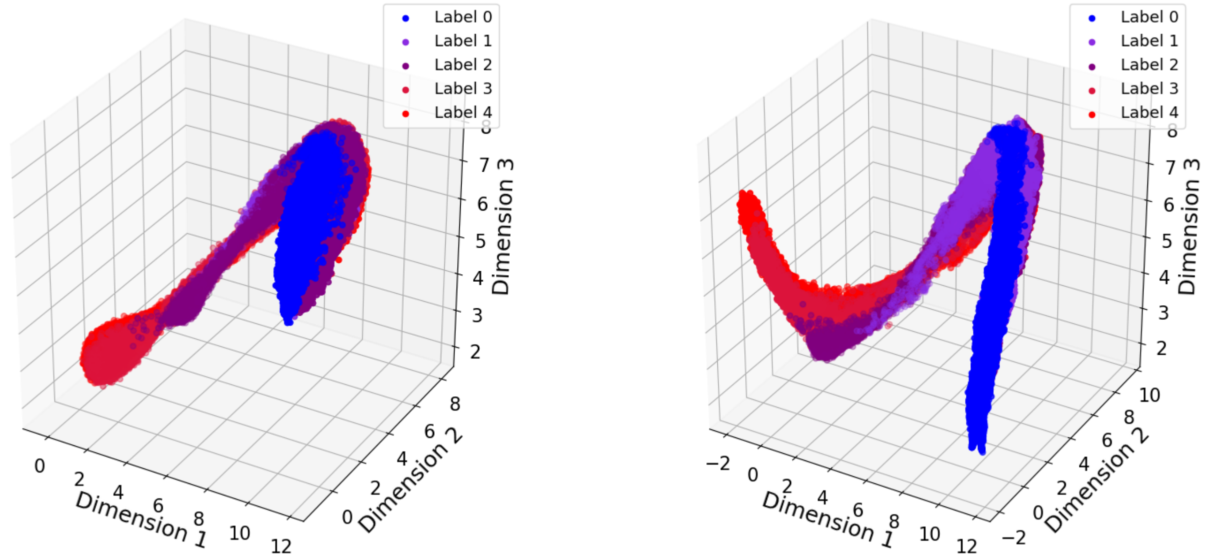

Figure 4.20 Dimensionality reduced Betti curve embeddings for brain imaging data, subject m292. Homology dimension 0 (left) and 1 (right).

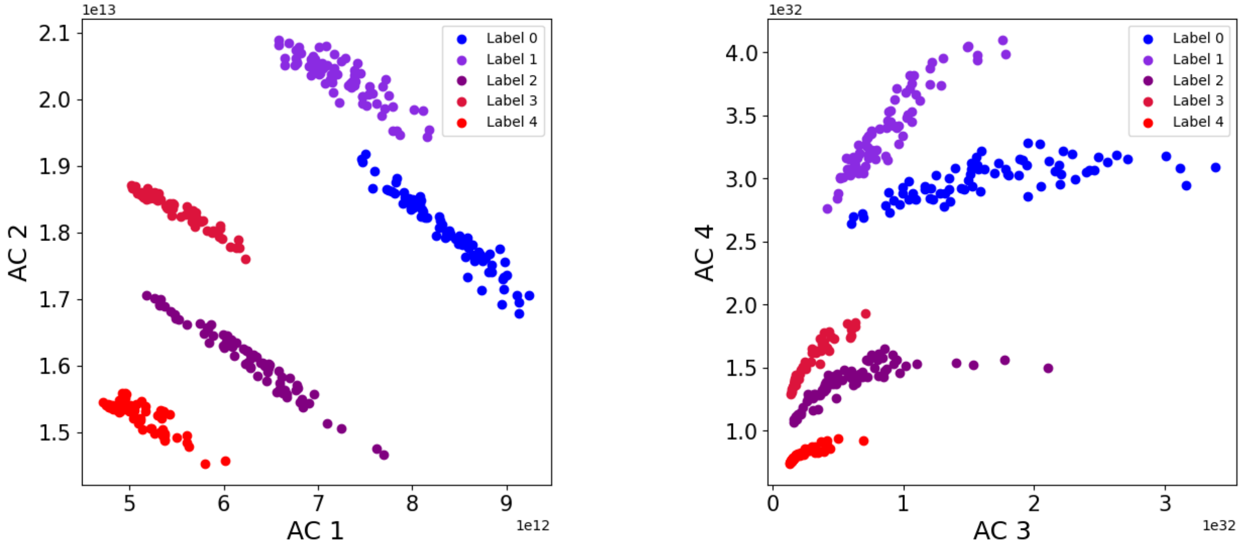

Figure 4.21 Adcock-Carlsson coordinates for brain imaging data, subject m292. Coordinates 1 and 2 on the left, 3 and 4 on the right.

4.6 Quantifying the Correlation Between Topological Complexity and Time Series Complexity

We conducted a simple approach for finding correlations between time series statistics indicating time series complexity with topological statistics indicating topological complexity. The approach consisted of correlating time series amplitude and the variance of the amplitude with the topological basic summary statistics.

As you can see in Table 4.1, the correlation is sometimes significant, and sometimes quite low.

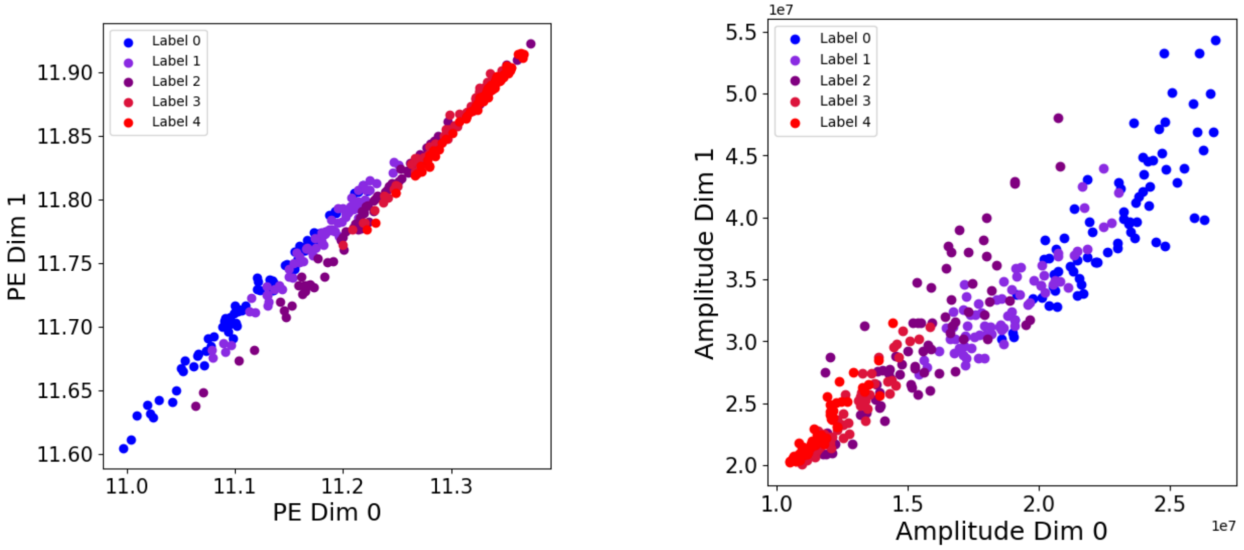

Figure 4.22 Persistent Entropy (left) and Amplitude (right) for brain imaging data, subject m292, each for both homology dimensions.

Table 4.1 Correlation of time series statistics indicating time series complexity with topological statistics indicating topological complexity.

	Mean EEG Amplitude	**EEG Amplitude Variance**
No Points, Dim 0	0.22348645	0.04401384
No Points, Dim 1	0.3812383	-0.45042328
No Points, Dim 2	0.33099998	-0.41640633
Amplitude, Dim 0	-0.15437558	0.49142078
Amplitude, Dim 1	0.53908987	-0.45419209
Amplitude, Dim 2	0.64841197	0.08495041
Persistence Entropy, Dim 0	0.66713319	-0.31385204
Persistence Entropy, Dim 1	0.45073513	-0.46159222
Persistence Entropy, Dim 2	0.33882067	-0.43717896

4.7 Quantifying the Correlation Between the Data Modalities

In the Anesthesia data, the EEG/EMG and brain imaging data was recorded simultaneously. Technically, they are both time series; the EEG and EMG data are a series of data points with a high temporal resolution, while the brain imaging data is a series of images with a lower temporal resolution. As they evolve over time, we assume the two data modalities show some shared patterns in their homology.

When we later feed the data to machine learning models, we hope the classifiers will recognize such shared patterns in a way that improves classification accuracy. Beforehand, we analyze such shared patterns in this chapter by computing correlations between summary statistics of both data modalities.

We therefore correlate the basic summary statistics of the segments of the EEG Anesthesia data with the basic summary statistics of the segments of brain imaging Anesthesia data.

As shown in table 4.2, the correlation between the basic summary statistics of the two modalities is negative in all cases. The absolute correlation is not significant in some cases, and very significant in other cases, like for persistence entropy.

Table 4.2 Correlation of the basic summary statistics of EEG and brain imaging data.

	Homology Dimension 0	**Homology Dimension 1**
Number of Points	-0.16449993	-0.72676456
Amplitude	-0.45882338	-0.00214351
Persistence Entropy	-0.80898952	-0.74508192

5 Methods for Classification

The following chapter will describe our methods for creating persistence diagrams from the data modalities, aligning the data, and finally classifying it into anesthesia and sleep stages.

5.1 Preprocessing

Before computing the persistence diagram, we apply some simple preprocessing steps.

Common Preprocessing Steps

For EEG data, common preprocessing steps for generating persistence diagrams include band-pass filtering to eliminate noise, downsampling the data, and removing artifacts [XDR21]. While transforming EEG data into another space is not widely applied, some studies have explored this approach. Those studies mostly use Independent Component Analysis (ICA) to extract source components or Fourier transform for denoising [XDR21].

For image data, blurring approaches like Gaussian blurring are commonly used preprocessing steps [Pul19].

Our Preprocessing Steps

As persistence diagrams are particularly sensitive to outliers [CVJ21], removing artifacts from the EEG/EMG data is essential. Our method to to detect artifacts consisted of labeling data points with high "z scores" as artifacts, which are the numbers of standard deviations the data point is different from the mean [BA08].There were artifacts in the sleep data but none in the anesthesia data (which was also reflected in direct plots of the data).

We conducted initial experiments with ICA on the EEG/EMG data. However, they yielded worse results than experiments with raw data, so we restricted our preprocessing for EEG/EMG data to removing outliers.

The brain imaging images are left as they were initially, as we do not want to lose local information with Gaussian blurring.

5.2 Creating EEG/EMG Data Persistence Diagrams

Related work

As mentioned, we want to create persistence diagrams for each 4-second interval in the EEG and EMG time series. However, we did not restrict our literature review to topological approaches analyzing EEG time series *windows* (such as [Zen+21]) but also reviewed literature capturing the homology of an *entire* EEG time series.

Topological Data Analysis has been applied to EEG data only recently [XDR21]. There are two main approaches to analyzing univariate EEG data with TDA. The first approach is to apply sublevel set filtration directly onto the EEG data, while the second approach is to apply TDA onto a vector space retrieved by *time delay embeddings* [XDR21].

Table 5.1, an extension to a survey in [XDR21], provides an overview of existing approaches for creating persistence diagrams from EEG data.

Table 5.1 Related literature for creating persistence diagrams from EEG data.

Paper	**Goal**	Preprocessing	**Method**
[Zen+21]	Time Series Forecasting	Segmenting	Time delay embedding
[Alt+21]	MI – BCI	-	Time delay embedding
[BB21]	Eyes-open/eyes-closed classification	-	Time delay embedding
[Yam+21]	Improving Detection of Delirium	-	Time delay embedding
[MD20]	Autism Spectrum Disorder (ASD)	-	Sublevel filtration
[Wan+20b]	Aphasia	ICA	Sublevel filtration
[Wan+20a]	Aphasia	ICA	Gradient filtration
[Mar+19]	Hypnotizability	ICA	Connectivity
[NM19]	Brain state classification	-	Sublevel filtration
[WC19]	Seizure localization	Fourier transform	Sublevel filtration
[AYI18]	MI – BCI	-	Time delay embedding
[Pia+18]	Detecting Epileptic seizure	-	Sublevel filtration
[WC18]	Epilepsy	Fourier transform	Sublevel filtration

Time Delay Embeddings

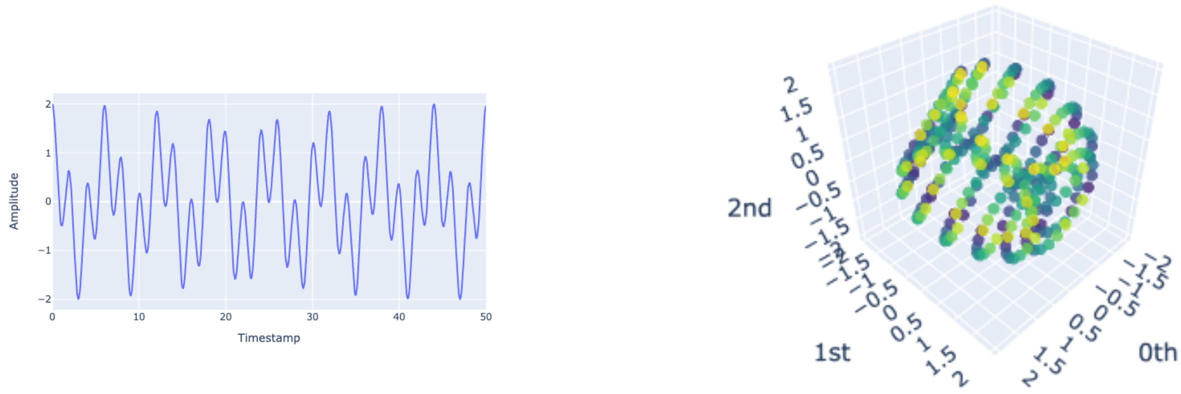

Figure 5.1 Example of a nonperiod time series (left) converted to a time delay embeddings (right) with embedding dimension $m = 3$ and time delay $\tau = 16$ [Tau+20].

With the time delay embeddings method, we can convert time series to point clouds, from which we can subsequently create a Vietoris-Rips or Čech filtration. Figure 5.1 shows an example of a time series converted to a point cloud with a time delay embedding. A time delay embedding is created by constructing vectors in a vector space using past observations.

For a chosen embedding dimension m and time delay τ, we form a new vector $\mathbf{y}(t)$ from the time series $x(t)$ as [Tau+20]

$$\mathbf{y}(t) = [x(t), x(t-\tau), x(t-2\tau), \ldots, x(t-(m-1)\tau)]. \tag{5.1}$$

We refer to this vector as a *time delay embedding* of $x(t)$.

We can easily reconstruct the time series $x(t)$ from the time delay embedding $\mathbf{y}(t)$. This is ensured by *Taken's Theorem*, which can be found in [Roh09].

Optimal Parameters

If we decide to use time delay embeddings, we must also decide on the delay and embedding dimension. *Mutual Information* and *False Nearest Neighbours* are commonly used methods to determine the optimal parameters of a time delay embedding [Alt+21].

To determine the optimal time delay τ^*, we can use a method called Mutual Information, provided by Giotto TDA [Tau+20].

Mutual information determines the "mutual information" between the original time series and its delayed version, so the amount of information the two time series share, for several possible values of the time delay τ [Tau+20].

Subsequently, the optimal delay τ^* is then determined as a value of τ that minimizes the mutual information [Tau+20]. Therefore, the time series delayed by τ^* is the least redundant to the original series.

False Nearest Neighbours is a method for determining the optimal embedding dimension m. The underlying assumption of the algorithm is that two points that are close in one dimension of the embedding should also be close in other dimensions of the embedding [Tau+20]. We compute the distances between two points in different embedding dimensions, and if the difference between these distances is above a certain threshold, we define the two points as "false nearest neighbours" [Tau+20]. We then determine the optimal embedding dimension as the embedding minimizing the number of false nearest neighbours [Tau+20].

Our Method in a Nutshell

After preprocessing, we convert all 4-second time series segments to a point cloud with a time delay embedding approach. We automatically determine the optimal time delay and embedding dimension with Mutual Information and False Nearest Neighbours. We then create a Vietoris-Rips filtration from each time delay embedding and create persistence diagrams with Giotto TDA [Tau+20], which we can use for further analysis.

Outlook for Multivariate EEG Data

When recording EEG and EMG data on humans, we are usually dealing with multivariate time series as we primarily use more than two electrodes. While our approach is typical for univariate time series, multivariate EEG data is usually approached differently.

Multivariate EEG data is often modeled in a *dependence network* [EYCO23], which also captures spatial information. Dependence networks depict the brain as a network of brain regions, which is a natural choice due to the network character of the brain, and model the dependence between the brain regions [EYCO23].

There are many ways to measure this dependence. Common metric choices are cross-correlation, coherence, partial coherence, and partially directed coherence [EYCO23].

[EYCO23] shows an example of how to approach a dependence network to analyze its homology. The authors depict each time series in the multivariate EEG as a vector in a high-dimensional vector space, in which the distance between the single vectors is based on their dependence [EYCO23]. They then build a Vietoris-Rips filtration in this vector space, using the vectors as vertices [EYCO23].

5.3 Creating Brain Imaging Data Persistence Diagrams

Related Work

Approaches using persistent homology on Calcium imaging are relatively rare, with one example being [Ble+22]. More approaches are applying persistent homology on fMRI data [Sto+21], which has an additional spatial dimension but still permits similar approaches to ours.

Our approach to creating persistence diagrams from the brain imaging data was heavily inspired by [Rie+20], which transforms fMRI data to cubical complexes using a sublevel set filtration. Like [Rie+20], we use cubical complexes but used a different kind of filtration, namely *radial filtration*, as radial filtration was the kind of filtration suggested for images by the software package Giotto TDA [Tau+20].

Our Method

We build cubical complexes and persistence diagrams directly from the brain imaging images in a construction where each pixel corresponds to a vertex in the cubical complex, using radial filtration.

5.4 Noise in Persistence Diagrams

We consciously decided not to remove noise from persistence diagrams, neither for the EEG/EMG nor the brain imaging data. As mentioned in Section 2.4, we do not want to remove local geometric information from our data. In a further evaluation, one could test how removing noise performs in comparison.

5.5 Data Fusion of Multimodal Brain Data

We are dealing with two fundamentally different data modalities: Video data, which contains temporal and spatial information, and time series data, which contains no spatial information but has a higher temporal resolution than the video data.

So far, we have converted 4-second segments of the time series (EEG) data into persistence diagrams and equivalently converted each frame in our brain imaging videos into a persistence diagram.

The two data modalities were recorded simultaneously; we want to combine them to extract knowledge from both.

There are several ways to combine data. “Early data fusion” means finding a common representation of two raw data modalities [VCP22]. Early data fusion can be in a new domain or in the domain of one of the data modalities, as in [DCK19], where fMRI signals are converted to time series and could potentially be used the same way as EEG signals subsequently.

“Intermediate data fusion” means first processing the data modalities individually and then merging intermediate results [VCP22]. In our case, this could happen either by combining persistence diagrams or "on feature level", e.g. by combining signatures in representations as combined persistence images before feeding them into one machine learning model [MKE23].

"Late data fusion", in our case, means training two separate machine learning on the modalities and subsequently combining their results [VCP22].

While late fusion could be a viable option, there might be cross-modal relationships between the two data modalities that we can exploit to improve the results of our analysis [Wil+]. We can conclude from high correlations between the feature distributions of the two modalities, described in Section 4.7, that such interrelations are quite likely. Early and intermediate fusion might be better suited to find such interdependencies.

Related Work of Fusing Brain Data

Table 5.2 shows a small survey of existing approaches for fusing brain data. The only topology-based data fusion approach listed is marked by a star, even though more fusion approaches exist based on topology, such as joint manifold learning [Dav+10]. Among those approaches in which early data fusion was applied, there were particularly many approaches using *Joint ICA*.

Table 5.2 Survey of data fusion approaches. Topology-based data fusion approach marked with a *.

Paper	Data	Type	Method
[DCK19]	fMRI and EEG	Early Data Fusion	Generating timeseries from fMRI data
[Del+21]	fNIRs and EEG	Intermediate Data Fusion	Choose features based on redundancy etc.
[CA08]	fMRI and EEG	Early Data Fusion	Joint ICA
[MKE23]*	WESAD and AV-MNIST	Intermediate Data Fusion	Fuse Persistence Image
[Lee+16]	PET and MRI	Intermediate Data Fusion	One-dimensional Projection
[Ste+13]	MEG and DTI	Early Data Fusion	Joint ICA
[MTF14]	EEG and fMRI?	Early Data Fusion	Joint ICA

Our Approach

We decided to fuse the two data modalities on the feature level, so at an intermediate step. One thought behind this is that unlike in many early data fusion approaches, in which the modalities are often fused into a dimensionality-reduced new space, we will not lose any information in a feature-based fusion process.

We want to capture the homology of intervals of the recorded data, and we chose the length of these intervals to be 4 seconds. For the time series data, we computed persistence diagrams directly for these 4-second intervals, so we retrieved one persistence diagram for each interval. We then computed statistics for each 4-second interval persistence diagram as described in section 3 and fed these statistics into a machine learning model.

For the brain imaging data, we first computed a persistence diagram and the respective statistics for each frame in a video. There are 80 frames in each 4-second interval. Instead of using the statistics of each diagram directly as machine learning features, we first averaged the statistics over sets of 80 persistence diagrams. This approach provides us with mean statistics representing the homology of each 4-second interval in the brain imaging data.

Feeding the statistics for 4-second intervals of both time series and brain imaging data into the same model is a type of intermediate, feature-level data fusion. With this type of data fusion, a machine learning model receiving statistics of both modalities for each 4-second interval will likely capture interrelations between the modalities.

Experiments with Early Data Fusion

As an alternative, or extension, to our intermediate data fusion approach, we considered early data fusion approaches, such as Joint ICA [Moo+08].

Many of the early data fusion approaches we reviewed are "blind source separation methods" [PS20]. An example is JIVE (Joint and Individual Variation Explained), which separates the data variability originating from individual characteristics of the data modalities and the variability originating from cross-modal interrelations [PS20].

The early data fusion approaches we reviewed all involve dimensionality reduction. Despite losing information in such approaches, exploring them further could be a valuable extension to intermediate data fusion. For instance, one could try to compute persistence diagrams and statistics from a space representing shared variability and shared "hidden factors", and then use these statistics as additional features in our pipeline. Including topological features computed from hidden factors could emphasize interdependencies between the modalities and improve the classification result.

The problem is that the interrelations we want to use must be hidden factors that both modalities share *when observed over time.* There is presumably no use in fusing a 4-second EEG interval with one frame of the brain imaging data (with no temporal information), as the shared variance in such a depiction might be only random noise.

After conducting a literature review and first practical experiments with two early data fusion approaches, Joint ICA and AJIVE (a variation of the JIVE algorithm) [Fen+18], it became clear that many early data fusion approaches struggle with combining data modalities with entirely different dimensions, such as our video and time series segments.

Even though AJIVE, unlike Joint ICA, does not require both modalities to have the exact same shape, video data is too high-dimensional for AJIVE. Thus, we would only be able to fuse a time series interval with one single frame, which would be limited in its ability to capture any time-dependent variance.

A method to include early data fusion approaches in our classification could be explored in future work.

5.6 Classification of Multimodal Brain Data into Anesthesia and Sleep Stages

Once we have fused our multimodal statistics to one feature dataframe, we can classify it and evaluate the classification performance. Here, we want to compare our topological approach to a baseline approach, which only uses statistical features. We will list all topological (machine learning) features and all statistical features that we use later in this chapter.

Classifiers Used for the Topological Features

On tabular data, tree-based machine learning models tend to outperform Deep Learning models [GOV24]. Additionally, deep learning models commonly require immense datasets for good results, such that we refrained from using a deep learning model and decided to use tree-based machine learning classifiers instead.

We decided to use *XGBoost* and *Random Forest*, which we implemented using the Python packages scikit-learn [Ped+11] and xgboost [TC16]. Both models are ensemble models based on Decision Trees [TC16] [Ped+11]. While Random Forest uses Bagging as an ensemble method [Ped+11], XGBoost uses Boosting as an ensemble method [TC16]. An explanation of Bagging and Boosting can be found in [Bï2].

Splitting into Train and Test Sets

To ensure the variance of our model is not too high, we perform 5-fold Cross-Validation during the classification. We do not split the data according to subjects, leaving, for instance, just one subject's data as the test set, as with a small amount of subjects, such a splitting strategy could make us suffer from variation between the subjects and randomness. Instead, we reserve 20% of each subject's data for the final test set and use 80% of the remaining data as a training set and 20% as a validation set in each fold.

As some features, particularly ATOL vectorizations, depend on training data, we cannot perform the splitting in the very end but define train and validation sets for each fold before computing the features. We compute 5 different versions of the ATOL vectorizations for each of the 5 folds (each with different training data), but compute all other functional summaries only once.

Features

Table 5.3 Topological features used for both modalities and both Anesthesia and Sleep data.

Functional Summary	**Statistics Used for Classification**
Landscape Amplitude	Direct Usage
Number of Points	Direct Usage
Persistence Entropy	Direct Usage
Largest Persistence	Direct Usage
Adcock-Carlsson Coordinates	Direct Usage
ATOL vectorizations	Direct Usage
Persistence Landscape	L1 norm Mean Standard Deviation Skewness Kurtosis Direct Usage of Embedding Coordinates
Heatkernel	Intensity Maximum Minimum
Silhouette	L1 norm Mean Standard Deviation Skewness Kurtosis Direct Usage of Embedding Coordinates
Persistence Image	Intensity Maximum Minimum
Betti Curve	L1 norm Mean Standard Deviation Skewness Kurtosis Direct Usage of Embedding Coordinates

Table 5.3 shows a list of machine learning features we computed from each functional summary listed in Chapter 3. First of all, we can use each functional summary directly; in the case of signatures, this would mean taking each coordinate of the signature embeddings as one feature, which we did for all signatures except the heatkernel and persistence image, as they are higher dimensional.

However, we saw in Section 4.5 that these direct coordinates do not show a clear separability. Additionally, they are so high-dimensional that a machine learning model might have trouble extracting important information from them. Therefore, they might not be the most meaningful way to use signatures for classification, and we additionally compute statistics from the signatures to capture them in just a few numbers, hoping that such statistics have more meaningfulness than the direct embeddings.

From the Betti curve, for example, we computed the mean, standard deviation, skewness, and kurtosis of the curve as statistics, which is an approach we saw in [Fel+19]. From the heatkernel, we computed intensity, and the maximum and minimum of the (absolute of) the embedding, which was inspired from [ASS23].

We computed the L1 norm of some signatures inspired by [Gid+20]. We initially also computed the L1 Norm of the persistence landscape but later noted that it is the same as the landscape amplitude, which we already use as a feature. An amplitude of the heatkernel could be similarly computed with the amplitude function in Giotto TDA [Tau+20] in future work, using the metric “heat”.

In theory, we could directly feed the kernel density estimates into a kernel-based machine learning model like Kernel SVM [Ped+11]. However, we want to have a holistic solution in which we can combine heatkernel and persistence image with other functional summaries, which is why we refrain from using Kernel SVM.

By using a rather large amount of features, we want to reach two main goals: First, we want to maximize accuracy by later selecting the features that maximize accuracy. Second, we want to find out which features contribute particularly much to our classification goal.

Something like an “overall best feature”, performing the best across all existing datasets, does not exist. The predictive power of features like ours must be assessed over many datasets. If some summary statistics have a higher predictive power across many datasets than others, this gives researchers valuable heuristics to decide which features to choose in a classification task. We want to add to this assessment by evaluating different features over two datasets.

We evaluate the predictive power of the features with two different metrics. One is the built-in feature importance method that decision-tree-based models in scikit-learn [Ped+11] automatically return; the other is the accuracy that XGBoost reaches when just using one feature. We show both metrics in Section 6.2.

To maximize the accuracy, we manually tested which feature combinations work best. We also tested automatic feature selection algorithms, which, however, did not improve the results.

Baseline Models

When choosing baseline methods to compare our topological approach, we first refrained from using pre-trained models as a baseline model. The fact that pre-trained models were trained on a larger amount of data might lead to an unfair advantage and make them unsuitable for a direct comparison.

After not yielding good results with LSTMs taking time series segments directly as an input, which we explained with our small amount of data, we decided to use the same traditional classifiers instead of deep learning models as baseline models, with different features than our topological features. We did not test deep learning models for computer vision, as we came to the conclusion that using the same classifiers but with different features makes the features more directly comparable.

Table 5.4 Traditional statistical features for the time series data.

Data Types	Statistics Used for Classification
KATS Statistics	Mean Variance Entropy Lumpiness Stability Flat Spots Hurst Standard Deviation First Derivative Crossing Points Binarized Means The Test Statistic Based on the KPSS Test Heterogenity Histogram mode Linearity
KATS Level Shift Features	Level Shift Index Level Shift Size
KATS Autocorrelation Features	First ACF Value (Original TS) Sum of Squares of First five ACF Values (Original TS) First ACF Value (Differenced TS) Sum of Squares of First five ACF Values (Differenced TS) First ACF Value (Twice-Differenced TS) Sum of Squares of First five ACF Values (Twice-Differenced TS) AC coefficient First Seasonal Lag
Additional Features	Longest Sequence of Values Above Mean Longest Sequence of Values Below Mean Median Minimum Maximum Range between Minimum and Maximum 25th Percentile 75th Percentile 90th Percentile Number of Peaks (Local Maxima) Number of Valleys (Local Minima) Kurtosis Skewness

The baseline features are all statistical and are shown in Table 5.4 and 5.5. For time series data (Table 5.4), we used features proposed in the Python module “Kats” [Jia+22]. Additionally, we used some common self-chosen features such as skewness and kurtosis.

Table 5.5 Traditional statistical features for the brain imaging data.

Data Types	Statistics Used for Classification
Corner Features (FAST Corner Features and Most Prominent Corner Features)	Direct Coordinates Corner Count Mean Corner Position Corner Density Convex Hull Area Bounding Box Area SD of Positions Mean Distance Standard Deviation of Distances Mean and SD of Angles
Intensity Histogram Features	Direct Coordinates Mean Intensity SD Skewness Kurtosis Entropy Energy Contrast Mode Median 25th, 75th, 90th Percentile
Dense Optical Flow Features	Direct Coordinates Mean and SD of Flow Magnitudes Mean and SD of Flow Angles Mean and SD of Flow Gradient Magnitude Histogram of Flow Magnitudes Histogram of Flow Angles
Histograms of Canny Edges	Direct Coordinates Number of Edges Edge Density Mean and SD Edge Length Mean and SD of Edge Orientation
Local Binary Pattern	Direct Coordinates Mean Intensity SD Skewness Kurtosis Entropy Energy Contrast Mode Median 25th, 75th, 90th Percentile

For image data, [ST94] suggests using colour features as well as texture features and dissimilarity measures when using traditional Computer Vision. Table 5.5 shows the brain imaging data features we used. The images are black and white, so we used intensity histograms as colour features, local binary patterns, edges, and two different kinds of corners as texture features, and optical flow as a dissimilarity measure.

Parameter Tuning

We tune hyperparameters in a mix of manual tuning, where we track the best hyperparameters and features with the open-source software MLFLow [Zah+18], and automatic tuning with Grid Search. We tuned the models separately for each type of features (topological and statistical) and each of the Sleep and Anesthesia datasets.

Tests with pytest

We performed several tests with pytest [Kre+04] to avoid errors like duplicate rows. The tests fail if there are duplicate rows or if there are columns that contain labels or indices.

Visualizing the Classification Method for Topological Features

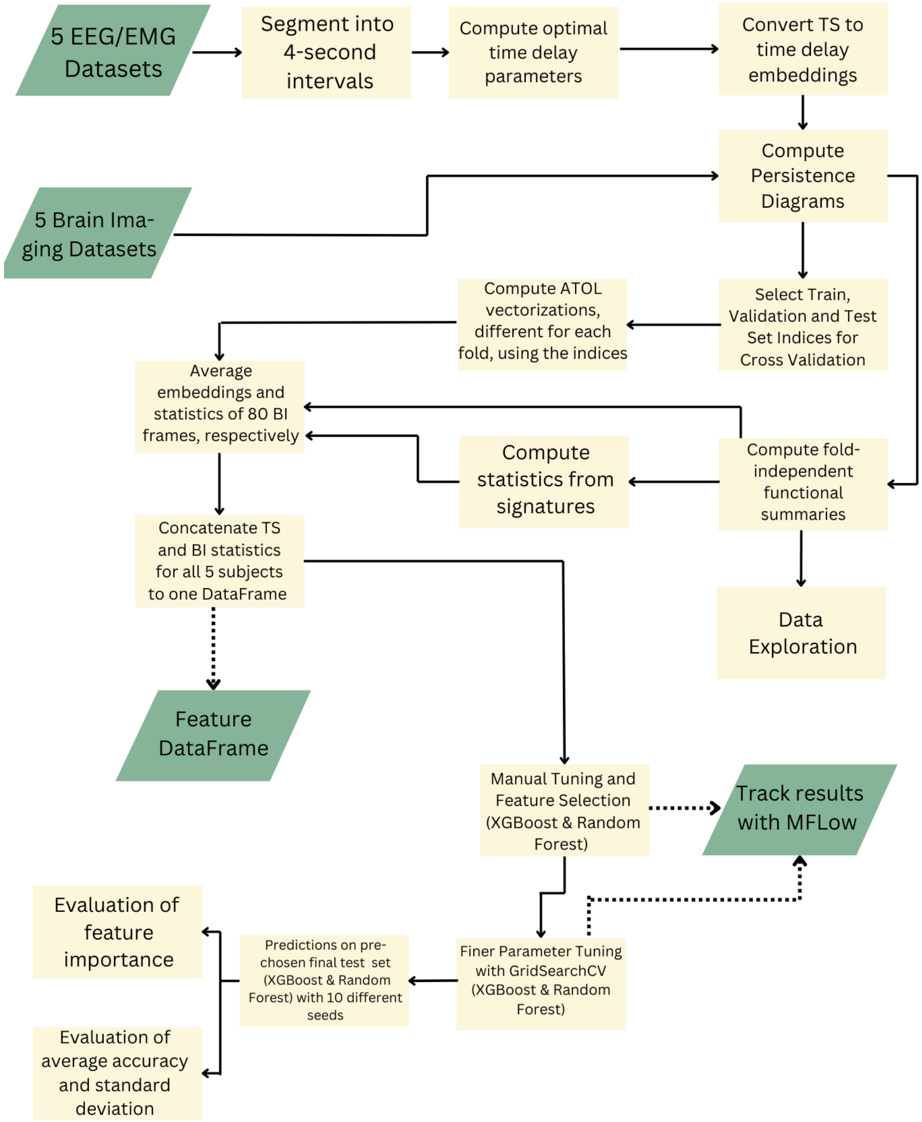
5 EEG/EMG Datasets
Segment into 4-second intervals
Compute optimal time delay parameters
Convert TS to time delay embeddings
Compute Persistence Diagrams
5 Brain Imaging Datasets
Compute ATOL vectorizations, different for each fold, using the indices
Select Train, Validation and Test Set Indices for Cross Validation
Average embeddings and statistics of 80 BI frames, respectively
Compute statistics from signatures
Compute fold-independent functional summaries
Concatenate TS and BI statistics for all 5 subjects to one DataFrame
Data Exploration
Feature DataFrame
Manual Tuning and Feature Selection (XGBoost & Random Forest)
Track results with MFLow
Evaluation of feature importance
Predictions on pre-chosen final test set (XGBoost & Random Forest) with 10 different seeds
Finer Parameter Tuning with GridSearchCV (XGBoost & Random Forest)
Evaluation of average accuracy and standard deviation

Figure 5.2 Flowchart of the classification algorithm using the topological features of the Anesthesia data. Instruction nodes are yellow rectangles, input/output nodes are green parallelograms.

Figure 5.2 shows a flowchart of the algorithm for the classification of the Anesthesia data with the topological features, aiming at giving an overview over the algorithm at a glance.

6 Results

Given the promising success of TDA in recent years, we aim to yield good results with our approach of computing machine learning features with persistent homology and subsequently classifying the data into consciousness stages with traditional machine learning models. The following chapter will show the classification results and compare the performance of our topological features with that of more traditional statistical features.

6.1 Classification Results

Anesthesia Data

Table 6.1 compares the results of both models when using only topological features against only using statistical features on the Anesthesia data, which consists of brain imaging data as well as EEG/EMG data. As XGBoost and Random Forest depend on random seeds as a parameter, we compute the results over ten different random seeds, such that we get a mean accuracy on the final test set and a standard deviation.

Generally, topological features outperform the statistical features by far, and when using topological features, a mix of both data modalities performs best with an accuracy of 99.86% with XGBoost, even though the accuracy is almost as high (99.76%) when only using brain imaging data.

Table 6.1 Performance comparison of topological features vs. statistical features on the Anesthesia dataset, using Random Forest and XGBoost.

Data Types	Model	Topological Features	Statistical Features
Brain Imaging	XGBoost	0.9976 ± 0.0014	0.8541 ± 0.0145
	Random Forest	0.9957 ± 0.0013	0.8059 ± 0.0049
EEG/EMG	XGBoost	0.9059 ± 0.0072	0.7384 ± 0.0140
	Random Forest	0.5995 ± 0.0098	0.6688 ± 0.0085
Both Modalities	XGBoost	0.9986 ± 0.0013	0.8899 ± 0.0075
	Random Forest	0.9906 ± 0.0021	0.8357 ± 0.0030

Even when only using the most important topological feature alone (Adcock-Carlsson coordinates), this leads to an accuracy of 0.9827 ± 0.0018 with XGBoost, such that the model still performs better than with all of the statistical features (with an accuracy of 0.8899 ± 0.0075).

After Cross Validation experiments with manual feature selection and parameter tuning, using all features except the persistence image and the persistence landscape statistics worked best.

Note that combining topological and statistical features leads to worse results than only using topological features.

Sleep Data

The Sleep dataset consists only of EEG and EMG data. Table 6.2 compares the results of both models when using only topological features against only using statistical features. Again, the topological features perform better than the statistical features with both models. For XGBoost, the mean accuracy is 0.9816 for the topological features and 0.9177 for the statistical features.

Table 6.2 Performance comparison of topological features vs. statistical features on the Sleep dataset, using Random Forest and XGBoost.

Model	Topological Features	Statistical Features
XGBoost	0.9816 ± 0.0014	0.9177 ± 0.0019
Random Forest	0.9571 ± 0.0011	0.9152 ± 0.0016

6.2 Feature Importance

Anesthesia Data

Table 6.3 depicts the XGBoost feature importances of the topological features on the Anesthesia data. It shows both the “Decision Tree Feature Importance”, which is the significance of the feature in the XGBoost model and which can be computed with a predefined Python function, as well as the “Single-Feature Accuracy”, which is the accuracy on the test set when only using the feature single-handedly.

Table 6.3 Feature importance on the Anesthesia Data, using XGBoost and both modalities for classification.

Feature	Single-Feature Accuracy	Decision Tree Importance
Landscape Amplitude	0.8421 ± 0.0067	0.0023 ± 0.0016
Number of Points	0.5699 ± 0.0107	0.0043 ± 0.0018
Persistence Entropy	0.5992 ± 0.0083	0.0024 ± 0.0013
Largest Persistence	0.3240 ± 0.0136	0.0060 ± 0.0027
Adcock-Carlsson Coordinates	**0.9827 ± 0.0018**	0.0187 ± 0.0074
ATOL vectorizations	0.8293 ± 0.0058	0.0374 ± 0.0081
Persistence Landscape Mean	0.9429 ± 0.0062	0.0167 ± 0.0020
Persistence Landscape Standard Deviation	0.9640 ± 0.0032	0.0440 ± 0.0054
Persistence Landscape Skewness	0.5568 ± 0.0202	0.0096 ± 0.0025
Persistence Landscape Kurtosis	0.5251 ± 0.0124	0.0072 ± 0.0033
Persistence Landscape Embedding Coordinates	0.9816 ± 0.0025	0.2220 ± 0.0611
heatkernel Intensity	0.8189 ± 0.0071	0.0050 ± 0.0030
heatkernel Maximum	0.6832 ± 0.0100	0.0030 ± 0.0016
heatkernel Minimum	0.7792 ± 0.0086	0.0019 ± 0.0014
Silhouette L1 norm	0.7429 ± 0.0064	0.0002 ± 0.0002
Silhouette Mean	0.7557 ± 0.0074	0.0018 ± 0.0017
Silhouette Standard Deviation	0.7459 ± 0.0063	0.0066 ± 0.0072
Silhouette Skewness	0.7304 ± 0.0061	0.0030 ± 0.0025
Silhouette Kurtosis	0.6946 ± 0.0098	0.0056 ± 0.0091
Silhouette Embedding Coordinates	0.9816 ± 0.0025	0.1272 ± 0.0372
Persistence Image Intensity	0.2661 ± 0.0070	0.0005 ± 0.0004
Persistence Image Maximum	0.2661 ± 0.0070	0.0003 ± 0.0003
Persistence Image Minimum	0.2936 ± 0.0101	0.0001 ± 0.0001
Betti Curve L1 norm	0.6392 ± 0.0088	0.0010 ± 0.0009
Betti Curve Mean	0.8339 ± 0.0066	0.0014 ± 0.0017
Betti Curve Standard Deviation	0.7120 ± 0.0088	0.0042 ± 0.0038
Betti Curve Skewness	0.7005 ± 0.0092	0.0027 ± 0.0016
Betti Curve Kurtosis	0.6461 ± 0.0084	0.0011 ± 0.0014
Betti Curve Embedding Coordinates	0.9728 ± 0.0056	**0.4510 ± 0.0835**

The feature that performs best in the “Single-Feature Accuracy” is Adcock-Carlsson coordinates, even though some of the direct coordinates of the signatures perform almost as well. Note that we observed in Section 4.5 that the Adcock-Carlsson coordinates of our data are particularly separable, which could be one reason they are so significant.

Interestingly, the feature that performs best in Single-Feature Accuracy is an unstable functional summary. Recall that we decided to include small-scale holes in our analysis instead of labeling them as noise and removing them. As unstable functional summaries can potentially profit from such small-scale holes well, the decision to include topological noise apparently turned out to be good, enabling our TDA-based method to profit from local geometric information. However, it could be a further step to test noise removal against our method.

“Decision Tree Feature Importance” is calculated in xgboost using the default metric “gain”, which measures the average improvement in loss that a feature brings in the total loss [TC16]. Decision Tree Feature Importance is by far the highest for the direct embeddings. However, note that to compute the Decision Tree Feature Importance of a feature consisting of several coordinates, the feature importances of the single coordinates were summed up. The high Decision Tree Feature Importance of the embeddings does not indicate the importance of the single embedding entries, which might be quite low, but still shows that the overall contribution of the direct embedding vectors is quite high.

Out of the direct embeddings, the embeddings of the Betti curve perform the best in Decision Tree Feature Importance. As described in Section 3.3, the Betti curve is also an unstable functional summary. This makes the best-performing features in our classification two unstable features.

In general, we observe that our initial assumption that statistics computed from signatures are more meaningful than their direct embeddings does not hold. Using statistics instead of direct embeddings, however, still comes with the advantage of better computational efficiency during the classification.

Sleep Data

Table 6.4 depicts the same feature importances for the Sleep data. The observations that we made were quite similar to the Anesthesia dataset. Again, Adcock-Carlsson coordinates perform best in Single-Feature Accuracy, and a direct embedding vector performs best in Decision Tree Importance, in this case, the stable silhouette embeddings.

The fact that Adcock-Carlsson coordinates perform the best in Single-Feature Accuracy in both datasets is worth noting. As there is no overall best (TDA) feature that universally performs best across all datasets, evaluating the performance of different signatures as machine learning features across many datasets can turn out helpful.

Table 6.4 Feature importance on the Sleep Data, using XGBoost and both modalities for classification.

Feature	Single-Feature Accuracy	Decision Tree Importance
Landscape Amplitude	0.8812 ± 0.0027	0.0089 ± 0.0029
Number of Points	0.7421 ± 0.0044	0.0178 ± 0.0068
Persistence Entropy	0.7769 ± 0.007	0.0072 ± 0.0018
Largest Persistence	0.8740 ± 0.0033	0.0022 ± 0.0008
Adcock-Carlsson Coordinates	**0.9843 ± 0.0009**	0.0584 ± 0.0032
ATOL vectorizations	0.8899 ± 0.0028	0.0118 ± 0.0022
Persistence Landscape Mean	0.8500 ± 0.0024	0.0016 ± 0.0008
Persistence Landscape Standard Deviation	0.8566 ± 0.0031	0.0020 ± 0.0008
Persistence Landscape Skewness	0.6585 ± 0.0059	0.0013 ± 0.0002
Persistence Landscape Kurtosis	0.6578 ± 0.0037	0.0006 ± 0.0003
Persistence Landscape Embedding Coordinates	0.8907 ± 0.0014	0.2286 ± 0.0089
heatkernel Intensity	0.8625 ± 0.0048	0.0040 ± 0.0008
heatkernel Maximum	0.8858 ± 0.002	0.0056 ± 0.0023
heatkernel Minimum	0.8651 ± 0.0038	0.0053 ± 0.0015
Silhouette L1 norm	0.8838 ± 0.0018	0.0068 ± 0.0033
Silhouette Mean	0.8817 ± 0.0021	0.0034 ± 0.0006
Silhouette Standard Deviation	0.8781 ± 0.0028	0.0045 ± 0.0012
Silhouette Skewness	0.6741 ± 0.0056	0.0039 ± 0.0015
Silhouette Kurtosis	0.6712 ± 0.0059	0.0039 ± 0.0008
Silhouette Embedding Coordinates	0.8939 ± 0.0017	**0.3092 ± 0.0163**
Persistence Image Intensity	0.8792 ± 0.0022	0.0121 ± 0.0049
Persistence Image Maximum	0.8892 ± 0.0018	0.0139 ± 0.0053
Persistence Image Minimum	0.8669 ± 0.0027	0.0051 ± 0.0016
Betti Curve L1 norm	0.7437 ± 0.0031	0.0037 ± 0.0032
Betti Curve Mean	0.7620 ± 0.0031	0.0016 ± 0.0011
Betti Curve Standard Deviation	0.7631 ± 0.0033	0.0108 ± 0.0089
Betti Curve Skewness	0.7124 ± 0.0069	0.0031 ± 0.0013
Betti Curve Kurtosis	0.7187 ± 0.0055	0.0036 ± 0.0018
Betti Curve Embedding Coordinates	0.7773 ± 0.0056	0.2439 ± 0.0146

7 Discussion and Outlook

In this master thesis, we analyzed consciousness stages in brain data with the help of persistent homology, a method in TDA for computing topological holes in data at different spatial resolutions.

While TDA has limitations, such as its inability to distinguish data of different categories [XDR21], it has had promising successes in recent years. In contrast to purely geometric approaches, TDA detects topological, qualitative data characteristics. It can capture global and high dimensional features, which are robust and invariant to transformations such as translation and amplitude scaling, where other methods such as statistical analyses or also graph theory fail [XDR21].

Our goals were to explore the effectiveness of persistent homology in classifying brain states and to compare the performance of different functional summaries as machine learning features.

Using persistent homology, we created features for machine learning models to classify brain data into consciousness stages, namely sleep stages on one dataset and anesthesia stages on another. We then evaluated two classifiers with topological features and compared the performance of the topological features with the performance of traditional statistical features.

On both of our datasets, using traditional features performed significantly better than using statistical features, with an accuracy of 99.86% versus 88.99% on the anesthesia dataset and an accuracy of 98.16% vs 91.77% on the sleep dataset. TDA therefore appears to be an effective tool for classifying brain states.

On our primary dataset, the multimodal "Anesthesia data", classifiers performed very well with topological features computed from the brain imaging data alone but still slightly profited from additional topological features from the EEG/EMG data.

When comparing the feature importances, the Adcock-Carlsson coordinates, in particular, proved to be among the best-performing features. Adock-Carlsson coordinates are an unstable functional summary and might, therefore, profit from our decision to include small-scale holes in the analysis particularly much. During the data exploration phase, we observed that the Adcock-Carlsson coordinates formed distinct clusters, which supports their effectiveness for classifying different consciousness stages.

Persistent homology captures both local geometric and global topological information. In this master thesis, we profited from its ability to use both fields by explicitly including small-scale holes in our analysis, which are traditionally labeled as noise. Unstable functional summaries like Adcock-Carlsson coordinates can make use of such noise much better.

Detecting sleep and anesthesia stages can have significant applications in medical and scientific settings. For instance, accurate detection of sleep stages can help in diagnosing sleep disorders, while identifying anesthesia stages could improve patient care during surgery. However, there are ethical concerns when relying on data science methods, like topological data analysis and machine learning, for detecting consciousness stages. For example, testing the brain function of a patient in a coma could raise questions about the accuracy and reliability of these methods.

There are limitations to the present study, as we worked with a mouse brain dataset, which may not easily generalize to human brain data. For computational efficiency in real-time applications, not all features can be used, so selecting a subset, like just the Adcock-Carlsson coordinates, might be necessary. Future work could enhance the current approach by testing it on human datasets, including fMRI data. Additionally, testing on more diverse datasets and comparing how excluding noise performs could be interesting.

This master thesis contributes to the field of Topological Data Analysis by demonstrating the utility of persistent homology for classifying brain states. We showed that topological features can outperform traditional statistical methods in certain settings, and we identified the Adcock-Carlsson coordinates as particularly effective features. These findings open up exciting possibilities for future research, such as applying this approach to human data and further exploring its potential in medical diagnostics and real-time monitoring of consciousness.

A Appendix

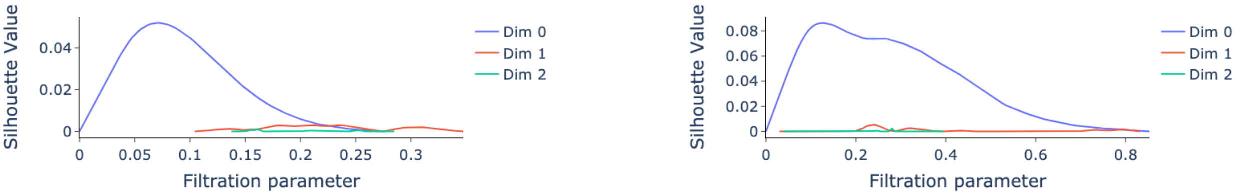

Figure A.1 Silhouettes computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Silhouette for lowest level of anesthesia on the left, silhouette for highest level of anesthesia on the right.

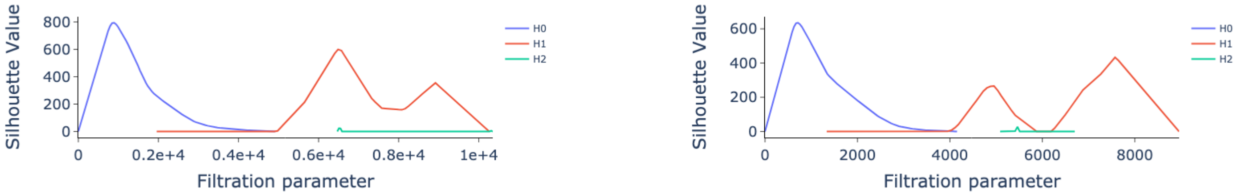

Figure A.2 Silhouettes computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Silhouette for lowest level of anesthesia on the left, silhouette for highest level of anesthesia on the right.

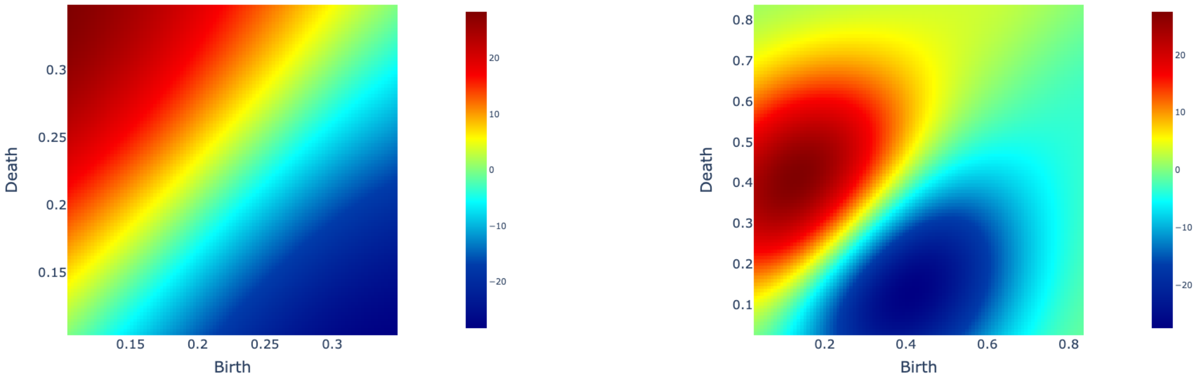

Figure A.3 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

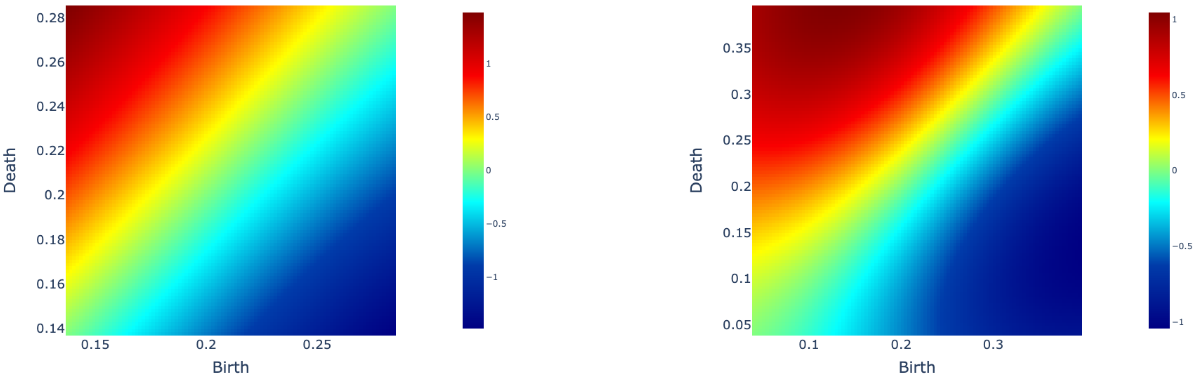

Figure A.4 Heatkernels for homology dimension 2 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

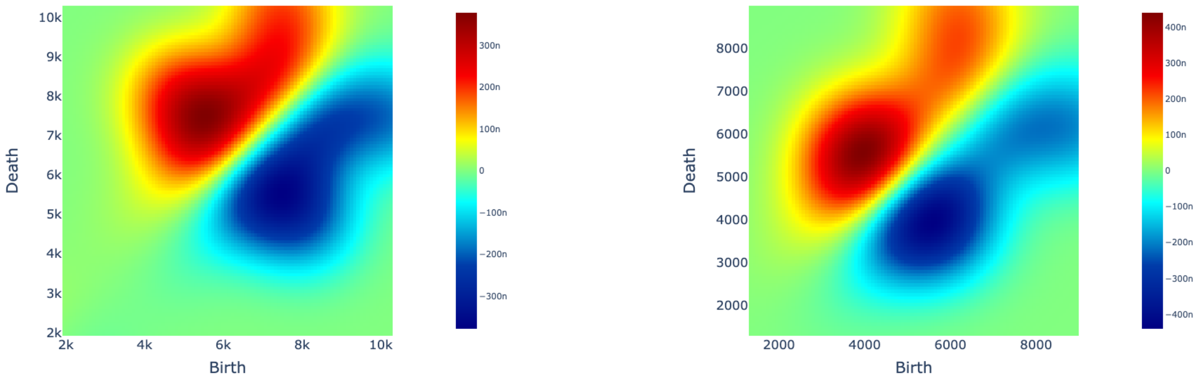

Figure A.5 Heatkernels for homology dimension 1 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

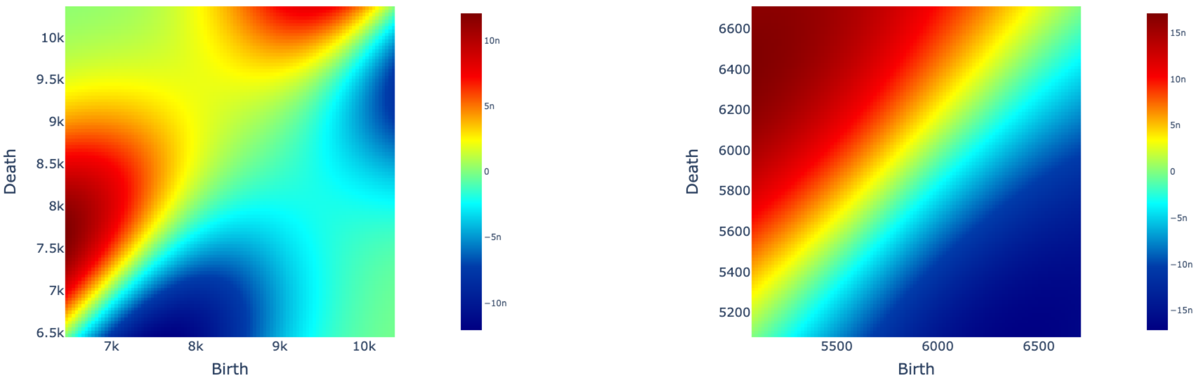

Figure A.6 Heatkernels for homology dimension 2 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right.

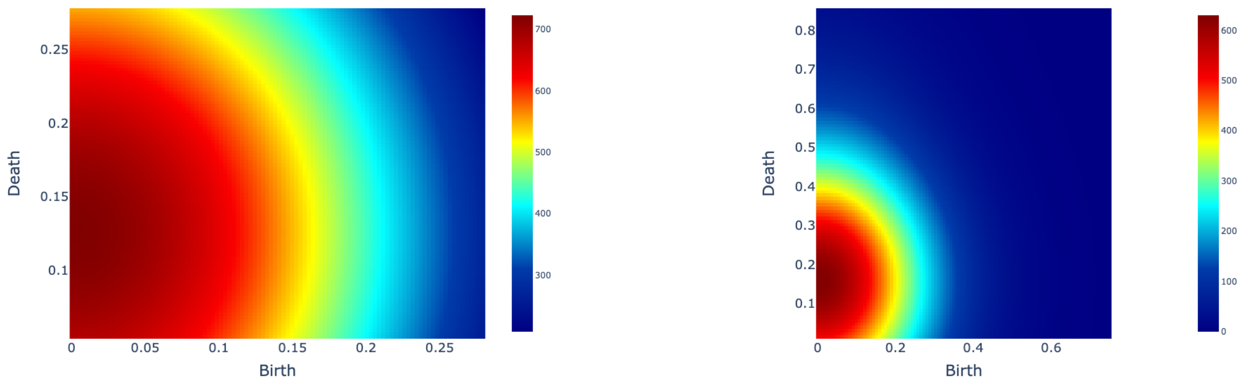

Figure A.7 Persistence images computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292, homology dimension 0. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right.

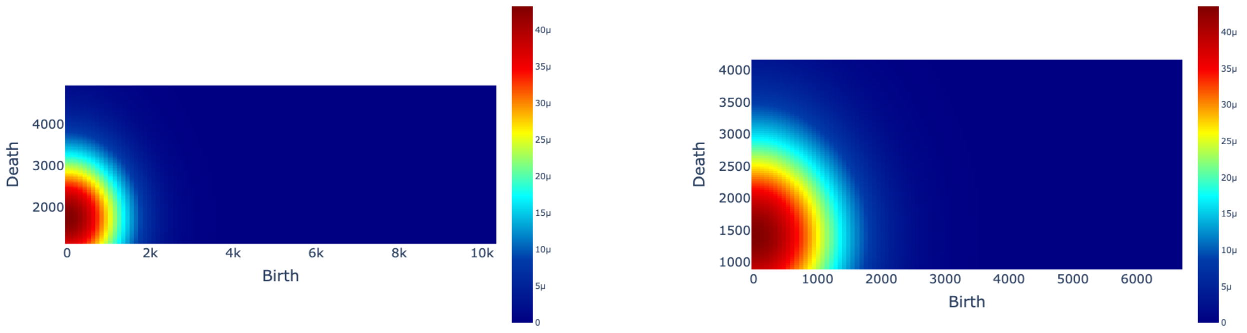

Figure A.8 Persistence images computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292, homology dimension 0. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right.

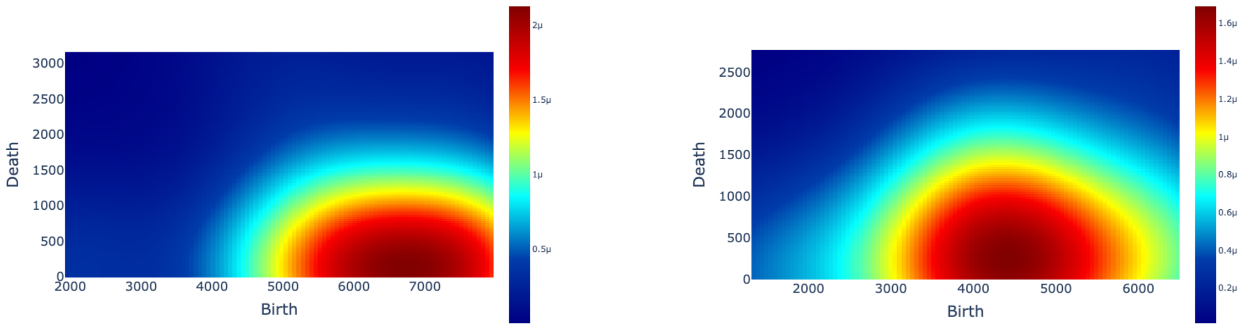

Figure A.9 Persistence images computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292, homology dimension 1. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right.

List of Figures

1.1 Data points showing a donut as an “intrinsic shape” [YM21]. 1
1.2 Coffee mug transformed to a donut [Sch18]. 2
1.3 Data points looking like circles each enclosing a tunnel (a type of hole). The tunnel in between the left upper data points is larger and more persistent than the tunnel in between the right lower data points [LR21]. 2

2.1 Euler’s representation of the Königsberg bridge problem with a graph, in which the nodes represent the islands and the edges represent the bridges [Ven+21]. 4
2.2 Examples of manifolds [Ken18] 6
2.3 Mapping a sphere to a plane [LA10]. 6
2.4 Numbers of holes of a point, a circle, a torus and a coffee mug [Dey14]. 8
2.5 Two simplicial complexes. Simple simplicial complex (left) [cfl09] and triangulation of a torus (right) [Ag214]. 9
2.6 Simplices of dimension 0, 1, 2 and 3 [EH10]. We call these simplices vertex, interval, triangle and tetrahedron. 9
2.7 Structure of simplices that is not a simplicial complex [Smi09]. 10
2.8 1-dimensional non-bounding cycles of a torus [Pel+07]. 11
2.9 Examples of the boundary operator applied on an oriented interval and an oriented triangle [Buc18]. 12
2.10 Non-bounding 1-cycles (black) and bounding 1-cycle (pink) of a torus [Pel+07]. 13
2.11 Exemplary cubical complex [GDJM11]. 15
2.12 Data sampled from a circle with little noise (left) and much noise (right). 17
2.13 Čech complex $C^{\text{Čech}}(D, \alpha)$ (left) and Vietoris-Rips complex $C^{\text{VR}}(D, \alpha)$ (right) constructed on a point cloud D [CM21]. 18
2.14 Vietoris-Rips filtration of a point cloud [Mun17]. 18
2.15 Persistence diagram of the filtration in Figure 2.14 [Mun17]. 19
2.16 Persistence diagrams of a point cloud sampled from a circle with little (left) and much (right) noise. 20
2.17 Flock of birds [AM21]. 22

3.1 Construction of the persistence landscape [HMR21]. 25
3.2 Exemplary power-weighted silhouettes for different choices of p, namely $p = 0.1$ and $p = 3$ [Cha+14]. 27
3.3 Construction of the Betti curve [HMR21]. Labels of the barcode axis are exluded. 27
3.4 Heatkernel computed from data points sampled from a circle with noise. Homology dimension 0 (left), 1 (middle) and 2 (right). 28
3.5 Construction of the persistence image [Ada+17]. 29

4.1 EEG data from the Sleep dataset. Awake (left), Non-REM (middle) and REM Sleep (left). . . 35
4.2 EEG (left) and EMG (right) data for subject m292 under the weakest level of anesthesia. . . 35
4.3 EEG (left) and EMG (right) data for subject m292 under the highest level of anesthesia. . . 36
4.4 Example frame of brain imaging recordings for subject m292. 36
4.5 Wasserstein Barycenters of EEG persistence diagrams, subject m292. Label 0 on the top left, followed by label 1, 2, 3 reading from left to right, and label 4 at the bottom right. . . . 37

4.6 Wasserstein Barycenters of brain imaging persistence diagrams, subject m292. Label 0 on the top left, followed by label 1, 2, 3 reading from left to right, and label 4 at the bottom right. 38
4.7 Dissimilarity matrix of the Wasserstein barycenters of the different labels for the EEG data, subject m292. Dissimilarity is measured by the Wasserstein metric. 38
4.8 Dissimilarity matrix of the Wasserstein barycenters of the different labels for the brain imaging data, subject m292. Dissimilarity is measured by the Wasserstein metric. 39
4.9 Box plots for the amplitude over different levels (EEG). 39
4.10 Box plots for the number of points over different levels (EEG). 39
4.11 Box plots for the amplitude over different levels (brain imaging). 40
4.12 Box plots for the number of points over different levels (brain imaging). 40
4.13 Persistence landscapes computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Persistence landscape for lowest level of anesthesia on the left, persistence landscape for highest level of anesthesia on the right. 41
4.14 Persistence landscapes computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Persistence landscape for lowest level of anesthesia on the left, persistence landscape for highest level of anesthesia on the right. 41
4.15 Betti curve computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Betti curve for lowest level of anesthesia on the left, Betti curve for highest level of anesthesia on the right. 41
4.16 Betti curve computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Betti curve for lowest level of anesthesia on the left, Betti curve for highest level of anesthesia on the right. 42
4.17 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 42
4.18 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 43
4.19 Dimensionality reduced Betti curve embeddings for EEG data, subject m292. Homology dimension 0 (left), 1 (middle) and 2 (right). 43
4.20 Dimensionality reduced Betti curve embeddings for brain imaging data, subject m292. Homology dimension 0 (left) and 1 (right). 44
4.21 Adcock-Carlsson coordinates for brain imaging data, subject m292. Coordinates 1 and 2 on the left, 3 and 4 on the right. 44
4.22 Persistent Entropy (left) and Amplitude (right) for brain imaging data, subject m292, each for both homology dimensions. 45

5.1 Example of a nonperiod time series (left) converted to a time delay embeddings (right) with embedding dimension $m = 3$ and time delay $\tau = 16$ [Tau+20]. 48
5.2 Flowchart of the classification algorithm using the topological features of the Anesthesia data. Instruction nodes are yellow rectangles, input/output nodes are green parallelograms. 57

A.1 Silhouettes computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292. Silhouette for lowest level of anesthesia on the left, silhouette for highest level of anesthesia on the right. 64
A.2 Silhouettes computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292. Silhouette for lowest level of anesthesia on the left, silhouette for highest level of anesthesia on the right. 64
A.3 Heatkernels for homology dimension 0 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 64

A.4 Heatkernels for homology dimension 2 from the Wasserstein barycenter of the EEG Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 65
A.5 Heatkernels for homology dimension 1 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 65
A.6 Heatkernels for homology dimension 2 from the Wasserstein barycenter of the brain imaging Anesthesia data, subject m292. Heatkernel for lowest level of anesthesia on the left, heatkernel for highest level of anesthesia on the right. 65
A.7 Persistence images computed from the Wasserstein barycenter of the EEG anesthesia data, subject m292, homology dimension 0. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right. 66
A.8 Persistence images computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292, homology dimension 0. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right. . . . 66
A.9 Persistence images computed from the Wasserstein barycenter of the brain imaging anesthesia data, subject m292, homology dimension 1. Persistence images for lowest level of anesthesia on the left, Persistence images for highest level of anesthesia on the right. . . . 66

List of Tables

4.1 Correlation of time series statistics indicating time series complexity with topological statistics indicating topological complexity. 45
4.2 Correlation of the basic summary statistics of EEG and brain imaging data. 46

5.1 Related literature for creating persistence diagrams from EEG data. 48
5.2 Survey of data fusion approaches. Topology-based data fusion approach marked with a *. . 51
5.3 Topological features used for both modalities and both Anesthesia and Sleep data. 53
5.4 Traditional statistical features for the time series data. 55
5.5 Traditional statistical features for the brain imaging data. 56

6.1 Performance comparison of topological features vs. statistical features on the Anesthesia dataset, using Random Forest and XGBoost. 58
6.2 Performance comparison of topological features vs. statistical features on the Sleep dataset, using Random Forest and XGBoost. 59
6.3 Feature importance on the Anesthesia Data, using XGBoost and both modalities for classification. 59
6.4 Feature importance on the Sleep Data, using XGBoost and both modalities for classification. 61

Bibliography

[AM21] Adams, H. and Moy, M. (2021). "Topology Applied to Machine Learning: From Global to Local". *Frontiers of Artificial Intelligence* 4.

[Ada+17] Adams, H. et al. (2017). "Persistence Images: A Stable Vector Representation of Persistent Homology". *Journal of Machine Learning Research* 18. 1–35.

[ACC16] Adcock, A., Carlsson, E., Carlsson, G. (2016). "The Ring of Algebraic Functions on Persistence Bar Codes". *Homology, Homotopy and Applications* 18.1. 381–402.

[Ag214] Ag2gaeh (2014). "Torus: triangulated". *Wikimedia Commons.*

[Ali+23] Ali, D. et al. (2023). "A Survey of Vectorization Methods in Topological Data Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45. 14069–14080.

[AYI18] Altındis, F., Yılmaz B., B. S., Icoz, K. (2018). "Use of topological data analysis in motor intention based brain-computer interfaces". *European Signal Processing Conference (EUSIPCO)* 26. 1695–1699.

[Alt+21] Altındis, F. et al. (2021). "Parameter investigation of topological data analysis for eeg signals". *Biomed. Signal. Process. Control.* 63.

[ABA22] Altschuler, J. M. and Boix-Adserà, E. (2022). "Wasserstein Barycenters Are NP-Hard to Compute". *SIAM Journal on Mathematics of Data Science* 4.1. 179–203.

[Am20] Amézquita, E. J. et al. (2020). "The shape of things to come: Topological data analysis and biology, from molecules to organisms". *Developmental Dynamics* 249. 816–833.

[ASS23] Andreeva, R., Sarkar, A., Sarkar, R. (2023). "Machine learning and topological data analysis identify unique features of human papillae in 3D scans". *Sci Rep* 13.

[AGDR19] Atienza, N., Gonzalez-Diaz, R., Rucco, M. (2019). "Persistent Entropy for Separating Topological Features from Noise in Vietoris-Rips Complexes". *Journal of Intelligent Information Systems* 52. 637–655.

[AGDST18] Atienza, N., Gonzalez-Diaz, R., Soriano-Trigueros, M. (2018). "On the stability of persistent entropy and new summary functions for topological data analysis".

[ALS13] Attali, D., Lieutier, A., Salinas, D. (2013). "Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shape". *Computational Geometry* 46.4. 448–465.

[ALS19] Attali, D., Lieutier, A., Salinas, D. (2019). "When Convexity Helps Collapsing Complexes". *International Symposium on Computational Geometry.*

[BPP21] Barnes, D., Polanco, L., Perea, J. A. (2021). "A Comparative Study of Machine Learning for Persistence Diagrams". *Front. Artif. Intell.* 4.

[BHG22] Beers, D., Harrington, H., Goriely, A. (2022). "Stability of topological descriptors for neuronal morphology". *arXiv*: 2211.09058.

[BA08] Behboodian, J. and Asgharzadeh, A. (2008). "On the distribution of Z-scores". *Iranian Journal of Science and Technology, Transaction A* 32.

[Ber+20] Berry, E. et al. (2020). "Functional Summaries of Persistence Diagrams". *Journal of Applied and Computational Topology* 4. 211–262.

[Bil+21] Billings, J. et al. (2021). "Simplicial and topological descriptions of human brain dynamics". *Network Neuroscience* 5.2. 549–568.

[BB21] Bischof, B. and Bunch, E. (2021). “Geometric feature performance under downsampling for eeg classification tasks”. *arXiv*: 2102.07669.

[Ble+22] Blevins, A. S. et al. (2022). “From calcium imaging to graph topology”. *Network neuroscience* 6.4. 1125–1147.

[BD06] Blinowska, K. and Durka, P. (2006). “Electroencephalography (EEG)”. *Wiley Encyclopedia of Bio-medical Engineering.* John Wiley and Sons, Ltd.

[Bub15] Bubenik, P. (2015). “Statistical Topological Data Analysis using Persistence Landscapes”. *Journal of Machine Learning Research* 16. 77–102.

[Buc18] Buchet M., H. Y. O. I. (2018). “Persistent Homology and Materials Informatics”. *Tanaka, I. (eds) Nanoinformatics.*

[Bï2] Bühlmann, P. (2012). “Bagging, Boosting and Ensemble Methods”. *Handbook of Computational Statistics: Concepts and Methods.* Springer. 985–1022.

[CA08] Calhoun, V. D. and Adalı, T. (2008). “Feature-Based Fusion of Medical Imaging Data”. *IEEE Trans Inf Technol Biomed.* 13.5. 711–720.

[Car13] Carandang, R. (2013). “Introduction to the Homology Groups of Complexes”. *University of Chicago REU.*

[Car19] Carlson, G. (2019). “Persistent Homology and Applied Homotopy Theory”. *Handbook of Homotopy Theory.* CRC Press.

[Car09] Carlsson, G. (2009). “Topology and Data”. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC* 46. 255–308.

[CVJ21] Carlsson, G. and Vejdemo-Johansson, M. (2021). “Topological Data Analysis with Applications”. Cambridge University Press.

[Car+05] Carlsson, G. et al. (2005). “Persistence Barcodes for Shapes.” *International Journal of Shape Modeling* 11. 149–188.

[Cas06] Casati, R. (2006). “Topology and Cognition”. *Encyclopedia of Cognitive Science.* John Wiley and Sons, Ltd.

[cfl09] cflm (2009). “An example of a simplicial complex”. *Wikimedia Commons.*

[CM21] Chazal, F. and Michel, M. (2021). “An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists”. *Frontiers of Artificial Intelligence* 4.

[Cha+09] Chazal, F. et al. (2009). “Proximity of Persistence Modules and their Diagrams”. *SCG ’09: Proceedings of the twenty-fifth annual symposium on Computational geometry.* 237 –246.

[Cha+14] Chazal, F. et al. (2014). “Stochastic Convergence of Persistence Landscapes and Silhouettes”. *SOCG’14: Proceedings of the thirtieth annual symposium on Computational geometry.* 474 – 483.

[CZS03] Chen, L., Zhang, S., Srinivasan, M. V. (2003). “Global perception in small brains: topological pattern recognition in honey bees”. *Proceedings of the National Academy of Sciences of the United States of America* 100.11. 6884–6889.

[Che+13] Chen, T. et al. (2013). “Ultrasensitive fluorescent proteins for imaging neuronal activity”. *Nature.* 295–300.

[CSEH07] Cohen-Steiner, D., Edelsbrunner, H., Harer, J. (2007). “Stability of Persistence Diagrams”. *Discrete Comput Geometry* 37. 103–120.

[Dav+10] Davenport, M. A. et al. (2010). “High-Dimensional Data Fusion via Joint Manifold Learning”. *AAAI Fall Symposium.*

[DCK19] Deco, G., Cruzat, J., Kringelbach, M. L (2019). “Brain songs framework used for discovering the relevant timescale of the human brain”. *Nature Communications* 10.583.

[Del+21] Deligani, R. J. et al. (2021). “Multimodal fusion of EEG-fNIRS: a mutual information-based hybrid classification framework”. *Biomedical optics express* 12.3.

[Dey14] Dey, A. (2014). “Topological Autoencoders”. *Weights and Biases.*

[Dey22] Dey, T. K. (2022). “Computational Topology for Data Analysis”. Cambridge University Press.

[DFF15] Di Fabio, B. and Ferri, M. (2015). “Comparing Persistence Diagrams Through Complex Vectors”. *Image Analysis and Processing* 9279. 294–305.

[Doc21] Doctrow, B. (2021). “Device allows paralyzed man to communicate with words”. *NIH Research Matters.*

[Ede14] Edelsbrunner, H. (2014). “A Short Course in Computational Geometry and Topology”. Springer International Publishing.

[EH10] Edelsbrunner, H. and Harer, J. (2010). “Computational Topology: An Introduction”. American Mathematical Soc.

[ELZ02] Edelsbrunner, H., Letscher, D., Zomorodian, A. (2002). “Topological Persistence and Simplification”. *Discrete and Computational Geometry* 28. 511–533.

[EYCO23] El-Yaagoubi, A. B., Chung, M. K., Ombao, H. (2023). “Topological Data Analysis for Multivariate Time Series Data”. *Entropy* 25.11. 1509.

[FMN16] Fefferman, C., Mitter, S., Narayanan, H. (2016). “Testing the Manifold Hypothesis”. *Journal of the American Mathematical Society* 29. 983–1049.

[Fel+19] Feldbrugge, J. et al. (2019). “Stochastic Homology of Gaussian vs. non-Gaussian Random Fields: Graphs towards Betti Numbers and Persistence Diagrams”. *Journal of Cosmology and Astroparticle Physics.*09.

[Fen+18] Feng, Q. et al. (2018). “Angle-based joint and individual variation explained”. *Journal of Multivariate Analysis* 166. 241–265.

[Fer99] Ferri M., L. C. (1999). “Representing size functions by complex polynomials”. *Proc. Math. Met. in Pattern Recognition* 9.

[Gal23] Gallier, J. H. (2023). “Basics of Affine Geometry”. *Advanced Geometric Methods in Computer Science, University of Pennsylvania.*

[Gid+20] Gidea, M. et al. (2020). “Topological recognition of critical transitions in time series of cryptocurrencies”. *Physica A: Statistical Mechanics and its Applications* 548.

[Giu+24] Giunti, B. et al. (2024). “Amplitudes in persistence theory”. *Journal of Pure and Applied Algebra* 228.

[Glo11] Glover, G. H. (2011). “Overview of Functional Magnetic Resonance Imaging”. *Neurosurg Clin N Am.* 22.2. 133–9.

[GDJM11] Gonzalez-Diaz, R., Jiménez, M., Medrano, B. (2011). “Well-Composed Cell Complexes”. *Discrete Geometry for Computer Imagery* 6607. 153–162.

[GOV24] Grinsztajn, L., Oyallon, E., Varoquaux, G. (2024). “Why do tree-based models still outperform deep learning on tabular data?” *NIPS’22: Proceedings of the 36th International Conference on Neural Information Processing Systems.* 507 –520.

[Har08] Harpaz, Y. (2008). “The Classification Problem of Smooth Manifolds”. *Université Sorbonne Paris Nord, HUJI student seminar.*

[Hat02] Hatcher, A. (2002). “Algebraic Topology”. Cambridge University Press.

[Hat+23] Hatton, S. L. et al. (2023). “Quantitative and Qualitative Representation of Introductory and Advanced EEG Concepts: An Exploration of Different EEG Setups”. *J Undergrad Neurosci Educ.* 21.2.

[HMR21] Hensel, F., Moor, M., Rieck, B. (2021). "A Survey of Topological Machine Learning Methods". *Front. Artif. Intell* 4.

[Inc15] Inc., P. T. (2015). "Collaborative data science". Plotly Technologies Inc.

[Ize12] Izenman, A. (2012). "Introduction to manifold learning". *Wiley Interdisciplinary Reviews: Computational Statistics* 4.

[Jia+22] Jiang, X. et al. (2022). "Kats" version 0.2.0.

[JJ21] Johnson, M. and Jung, J. (2021). "Instability of the Betti sequence for persistent homology and a stabilized version of the Betti sequence". *J. Korean Soc. Ind. Appl. Math.* 25.4. 296–311.

[KMM04] Kaczynski, T., Mischaikow, K., Mrozek, M. (2004). "Computational Homology". Springer Science and Business Media.

[KK21] Karana, A. and Kayguna, A. (2021). "Time Series Classification via Topological Data Analysis". *Expert Systems with Applications* 183. 1–12.

[Ken18] Keng, B. (2018). "Manifolds: A Gentle Introduction". *Bounded Rationality.*

[KMN17] Kerber, M., Morozov, D., Nigmetov, A. (2017). "Geometry Helps to Compare Persistence Diagrams". *Journal of Experimental Algorithmics (JEA)* 22. 1–20.

[Klo10] Kloke, J. N. *Methods and Applications of Topological Data Analyis.* 2010.

[Kre+04] Krekel, H. et al. (2004). "pytest".

[Lal14] Lalwani, K. (2014). "Triangulation and Classification of 2-Manifolds". *arXiv*: 1407. 8478.

[LR21] Lazar, N. and Ryu, H. (2021). "The shape of things: Topological Data Analysis". *CHANCE : New Directions for Statistics and Computing.* 59 –64.

[LS21] Lecha Sánchez, M. (2021). "Persistent Homology: Functional Summaries of Persistence Diagrams for Time Series Analysis". *Treballs Finals de Grau de Matemàtiques, Facultat de Matemàtiques, Universitat de Barcelona.*

[Lee00] Lee, J. M. (2000). "Introduction to Topological Manifolds". Springer Science and Business Media.

[Lee+16] Leea, H. et al. (2016). "Integrated multimodal network approach to PET and MRI based on multidimensional persistent homology". *Human brain mapping* 38.3. 1387–1402.

[LL06] Lerner, K. L. and Lerner, B. W. (2006). "Real-Life Math". Thomson Gale.

[LA10] Lespinats, S. and Aupetit, M. (2010). "Mapping without visualizing local default is nonsense." *The European Symposium on Artificial Neural Networks.*

[MD20] Majumder S., A. F. M. F. and Das, K. (2020). "Detecting autism spectrum disorder using topological data analysis". *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 1210–1214.

[Mak+18] Makarenko, I. et al. (2018). "Topological data analysis and diagnostics of compressible magnetohydrodynamic turbulence". *Journal of Plasma Physics* 84.

[Mar+19] Marcelo, E. Ibáñez et al. (2019). "Topology highlights mesoscopic functional equivalence between imagery and perception: the case of hypnotizability". *Neuroimage* 200. 437–449.

[Mar17] Marchese, A. (2017). "Data Analysis Methods using Persistence Diagrams". *University of Tennessee.*

[MHM20] McInnes, L., Healy, J., Melville, J. (2020). "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". *arXiv*: 1802.03426.

[M0́8] Mémoli, F. (2008). "Gromov-Hausdorff distances in Euclidean spaces". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (Anchorage, 2008).* 1–8.

[Moo08] Moore, G. H. (2008). "The emergence of open sets, closed sets, and limit points in analysis and topology". *Historia Mathematica* 35.3. 220–241.

[Moo+08] Moosmann, M. et al. (2008). "Joint independent component analysis for simultaneous EEG-fMRI: Principle and Simulation". *International Journal of Psychophysiology* 67.3. 212–221.

[MTF14] Moustafa, A. N., Taha, T. E., Faragallah, O. S. (2014). "A Comparison of Joint ICA and Parallel ICA Multimodal Fusion Methods in Schizophrenia". *International Journal of Computer Applications* 95.9. 25–28.

[MHJ22] Mulas, R., Horak, D., Jost, J. (2022). "Graphs, Simplicial Complexes and Hypergraphs: Spectral Theory and Topology". *Higher-Order Systems.* 1–58.

[Mun17] Munch, E. (2017). "A User's Guide to Topological Data Analysis". *Journal of Learning Analytics* 4. 47–61.

[MKE23] Myers, A., Kvinge, H., Emerson, T. (2023). "TopFusion: Using Topological Feature Space for Fusion and Imputation in Multi-Modal Data". *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 600–609.

[NM19] Nasrin F., O. C. B. D. and Maroulas, V. (2019). "Bayesian topological learning for brain state classification". *IEEE International Conference On Machine Learning And Applications (ICMLA)* 18. 1247–1252.

[Nic+11] Nicolau, M. et al. (2011). "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival". *Proceedings of the National Academy of Sciences of the United States of America* 108.17. 7265–7270.

[NSW08] Niyogi, P., Smale, S., Weinberger, S. (2008). "Finding the Homology of Submanifolds with High Confidence from Random Samples". *Discrete and Computational Geometry* 39.

[OLK19] Oh, J., Lee, C., Kaang, B. (2019). "Imaging and analysis of genetically encoded calcium indicators linking neural circuits and behaviors". *Korean J Physiol Pharmacol.* 23.4. 237–249.

[PS20] Pakravan, M. and Shamsollahi, M. B. (2020). "Joint, Partially-joint, and Individual Independent Component Analysis in Multi-Subject fMRI Data". *IEEE Trans Biomed Eng* 67.7.

[Pat+18] Patrangenaru, V. et al. (2018). "Challenges in Topological Object Data Analysis". *Sankhya A* 81.1. 244–271.

[Ped+11] Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12. 2825–2830.

[Pel+07] Peltier, S. et al. (2007). "Controlling Geometry of Homology Generators". *Computer Vision Winter Workshop.* 115–121.

[Pia+18] Piangerelli, M. et al. (2018). "Topological classifier for detecting the emergence of epileptic seizures". *BMC Research Notes* 11.392.

[Pom03] Pomerantz, J. R. (2003). "Wholes, holes, and basic features in vision". *Trends in Cognitive Sciences* 7.11. 471–473.

[Pul19] Pulfer, E. (2019). "Different approaches to blurring digital images and their effect on facial detection". *University of Arkansas.*

[Qui13] Quinn, F. (2013). "The Triangulation of Manifolds". *arXiv*: 1310.7644.

[Rei+14] Reininghaus, J. et al. (2014). "A Stable Multi-Scale Kernel for Topological Machine Learning". *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 4741–4748.

[Rie+20] Rieck, B. et al. (2020). "Uncovering the Topology of Time-Varying fMRI Data using Cubical Persistence". *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems.* Curran Associates, Inc. 6900 –6912.

[Rob99] Robins, V. (1999). "Towards computing homology from finite approximations". *Topology Proceedings* 24.

[Roh09] Rohrbacker, N. (2009). “Analysis of Electroencephologram Data Using Time-Delay Embeddings to Reconstruct Phase Space”. *Dynamics at the Horsetooth* 1.

[RV07] Rote, G. and Vegter, G. (2007). “Effective Computational Geometry for Curves and Surfaces”. Springer.

[Roy+20] Royer, R. et al. (2020). “ATOL: Measure Vectorization for Automatic Topologically-Oriented Learning”. *International Conference on Artificial Intelligence and Statistics.*

[Sch18] Schleimer, S. (2018). “MA3F1:Introduction to Topology”. *Warwick.*

[SSM05] Schölkopf, B., Smola, A., Müller, K. R. (2005). “Kernel principal component analysis”. *Artificial Neural Networks.* Springer.

[ST80] Seifert, H. and Threlfall, W. (1980). “A Textbook of Topology”. Academic Press.

[ST94] Shi, J. and Tomasi, C. (1994). “Good Features to Track”. *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* 593–600.

[ST23] Skraba, P. and Turner, K. (2023). “Wasserstein Stability for Persistence Diagrams”. *arXiv*: 2006.16824.

[Smi09] Smithers888 (2009). “Simplicial complex nonexample”. *Wikimedia Commons.*

[Ste+13] Stephen, J. et al. (2013). “Using joint ICA to link function and structure using MEG and DTI in schizophrenia”. *Neuroimage* 83. 418–30.

[Sti12] Stillwell, J. (2012). “Poincaré and the early history of 3-manifolds”. *Bulletin of the American Mathematical Society* 49. 555–576.

[Sto+21] Stolz, B. J. et al. (2021). “Topological data analysis of task-based fMRI data from experiments on schizophrenia”. *Journal of Physics: Complexity* 2.3.

[Tan17] Tantivasadakarn, N. (2017). “Dimensional Reduction and Topological Invariants of Symmetry Protected-Topological Phases”. *Physical Review B* 96.19.

[Tau+20] Tauzin, G. et al. (2020). “giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration”. *arXiv*: 2004.02551.

[The15] The GUDHI Project (2015). “GUDHI User and Reference Manual”. GUDHI Editorial Board.

[TC16] Tianqi, C. and Carlos, G. (2016). “XGBoost: A Scalable Tree Boosting System”. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM. 785–794.

[TMO22] Turkeš, R., Montúfar, G., Otter, N. (2022). “On the Effectiveness of Persistent Homology”. *Neural Information Processing Systems.*

[Tur+21] Turkeš, R. et al. (2021). “Noise robustness of persistent homology on greyscale images, across filtrations and signatures”. *PLOS ONE* 16.9. 1–26.

[Ume17] Umeda, Y. (2017). “Time Series Classification via Topological Data Analysis”. *Transactions of the Japanese Society for Artificial Intelligence* 32.3.

[VM05] Van Meter, J. R. (2005). “The world problem: on the computability of the topology of 4-manifolds”. *General Relativity and Quantum Cosmology.*

[Ven+21] Venkat, I. et al. (2021). “Graph Analytics to Reason Citations of Prophets in the Holy Quran”. *Computational Intelligence in Information Systems.* Springer International Publishing. 192–201.

[VCP22] Vortmann, L., Ceh, S., Putze, F. (2022). “Multimodal EEG and Eye Tracking Feature Fusion Approaches for Attention Classification in Hybrid BCIs”. *Frontiers in Computer Science* 4.

[WC19] Wang Y., O. H. and Chung, M. K. (2019). “Statistical persistent homology of brain signals”. *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 1125–1129.

[WC18] Wang Y., O. H. and Chung, M. K. (2018). “Topological data analysis of single-trial electroencephalographic signals”. *Ann. Appl. Stat.* 12. 1506–1534.

[Wan+20a] Wang, Y. et al. (2020). “Topological signal processing in neuroimaging studies”. *IEEE International Symposium on Biomedical Imaging Workshops (ISBI Workshops)* 17. 1–4.

[Wan+20b] Wang, Y. et al. (2020). “Topology highlights neural deficits of post-stroke aphasia patients”. *IEEE International Symposium on Biomedical Imaging (ISBI)* 17. 754–757.

[Wil10] Willis, P. N. (2010). “Local Homeomorphisms that *-coomute with the shift”. *arXiv*: 1010.5739.

[Wil+] Wilson, H. et al. “Performance of data-driven inner speech decoding with same-task EEG-fMRI data fusion and bimodal models”. *arXiv*: 2306.10854.

[Wu+21] Wu, R. et al. (2021). “Topological dominance in peripheral vision”. *Journal of Vision* 21.10.

[XDR21] Xu, X., Drougard, N., Roy, R. N. (2021). “Topological Data Analysis as a New Tool for EEG Processing”. *Frontiers in Neuroscience* 15.

[Yam+21] Yamanashi, T. et al. (2021). “Topological data analysis (TDA) enhances bispectral EEG (BSEEG) algorithm for detection of delirium”. *Scientific Reports* 11.304.

[YM21] Yang, T. and Meng, J. (2021). “Manifold fitting algorithm of noisy manifold data based on variable-scale spectral graph”. *Soft Computing* 27.1. 471 –482.

[Zah+18] Zaharia, M. A. et al. (2018). “Accelerating the Machine Learning Lifecycle with MLflow”. *IEEE Data Eng. Bull.* 41. 39–45.

[Zen+21] Zeng, S. et al. (2021). “Topological Attention for Time Series Forecasting”. *Neural Information Processing Systems.*

[Zom05] Zomorodian, A. J. (2005). “Topology for Computing”. Cambridge University Press.