


A Survey of Topological Machine Learning Methods

Review Article**Author(s):**

Hensel, Felix; Moor, Michael; Rieck, Bastian Alexander 

Publication date:

2021-05-26

Permanent link:

<https://doi.org/10.3929/ethz-b-000513747>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Originally published in:

Frontiers in Artificial Intelligence 4, <https://doi.org/10.3389/frai.2021.681108>

Funding acknowledgement:

190466 - TOPAZ: Topology of Alzheimers (SNF)



A Survey of Topological Machine Learning Methods

Felix Hensel^{1,2}, Michael Moor^{1,2} and Bastian Rieck^{1,2*}

¹ Machine Learning and Computational Biology Laboratory, ETH Zurich, Zurich, Switzerland, ² Swiss Institute of Bioinformatics, Lausanne, Switzerland

OPEN ACCESS

Edited by:

Kathryn Hess,
École Polytechnique Fédérale de
Lausanne, Switzerland

Reviewed by:

Raphael Reinauer,
École Polytechnique Fédérale de
Lausanne, Switzerland
Matteo Caorsi,
L2F SA, Switzerland

*Correspondence:

Bastian Rieck
bastian.rieck@bsse.ethz.ch

Specialty section:

This article was submitted to
Machine Learning and Artificial
Intelligence,
a section of the journal
Frontiers in Artificial Intelligence

Received: 15 March 2021

Accepted: 13 April 2021

Published: 26 May 2021

Citation:

Hensel F, Moor M and Rieck B (2021)
A Survey of Topological Machine
Learning Methods.
Front. Artif. Intell. 4:681108.
doi: 10.3389/frai.2021.681108

The last decade saw an enormous boost in the field of computational topology: methods and concepts from algebraic and differential topology, formerly confined to the realm of pure mathematics, have demonstrated their utility in numerous areas such as computational biology personalised medicine, and time-dependent data analysis, to name a few. The newly-emerging domain comprising topology-based techniques is often referred to as topological data analysis (TDA). Next to their applications in the aforementioned areas, TDA methods have also proven to be effective in supporting, enhancing, and augmenting both classical machine learning and deep learning models. In this paper, we review the state of the art of a nascent field we refer to as “topological machine learning,” i.e., the successful symbiosis of topology-based methods and machine learning algorithms, such as deep neural networks. We identify common threads, current applications, and future challenges.

Keywords: computational topology, persistent homology, machine learning, topology, survey, topological machine learning

1. INTRODUCTION

Topological machine learning recently started to emerge as a field at the interface of topological data analysis (TDA) and machine learning. It is driven by improvements of computational methods, which make the calculation of topological features (via persistent homology, for instance) increasingly flexible and scalable to more complex and larger data sets.

Topology is colloquially often referred to as encoding the overall shape of data. Hence, as a complement to localised and generally more rigid geometric features, topological features are suitable to capture multi-scale, global, and intrinsic properties of data sets. This utility has been recognised with the rise of TDA, and topological information is now generally accepted to be relevant in the context of data analysis. Numerous works aim to leverage such information to gain a fundamentally different perspective on their data sets. We want to focus on a recent “outgrowth” of TDA, i.e., the integration of topological methods to *enhance* or *augment* both classical machine learning methods and deep learning models.

Our survey therefore discusses this ongoing synthesis of topology and machine learning, giving an overview of recent developments in the field. As an emerging research topic, topological machine learning is highly active and rapidly developing. Our survey is therefore explicitly not intended as a formal and complete review of the field. We rather want to identify, present, and discuss some of the main directions of developments, applications, and challenges in topological machine learning as we perceive it based on our own research background. Our aim is to provide newcomers to the field with a high-level overview of some of the central developments and techniques that have been developed, highlighting some “nuggets,” and outlining common threads and future challenges. We

focus on publications in major machine learning conferences (such as AISTATS, ICLR, ICML, and NeurIPS) and journals (such as JMLR) but want to note that the selection of topics and papers presented here reflects our own preferences and knowledge. In particular, we decided against the inclusion of unpublished work in this area.

The survey is broadly structured as follows: we first provide a brief mathematical background on persistent homology, one of the core concepts of topological data analysis, in section 2. Following the introduction, the main part of the survey is in section 3. Section 3.2 focuses on what we term *extrinsic topological features* in machine learning. These methods are mainly concerned with the transformation of topological descriptors of data into feature vectors of fixed dimensionality, permitting their use as features in machine learning frameworks. This is in contrast to *intrinsic topological features*, portrayed in section 3.3, which employ topological features to analyse or influence the machine learning model itself, for instance by architectural choices or regularisation. Finally, section 4 discusses future directions and challenges in topological machine learning.

2. BACKGROUND ON ALGEBRAIC TOPOLOGY AND PERSISTENT HOMOLOGY

This section provides some background on basic concepts from algebraic topology and persistent homology. For in-depth treatments of the subject matter, we refer to standard literature (Bredon, 1993; Hatcher, 2000; Edelsbrunner and Harer, 2010). Readers familiar with algebraic topology and the concept of persistent homology may safely skip this section.

A basic hypothesis in data analysis which drives current research is that data has *shape*, or put differently, that data is sampled from an underlying manifold—the so-called “manifold hypothesis” (Fefferman et al., 2013). Instead of restricting the analysis to statistical descriptors, *topological data analysis* (TDA) aims to analyse data from a fundamentally different perspective by investigating this underlying manifold structure in an algebraic fashion. Namely, one computes descriptors of data sets which are *stable* under perturbation and encode intrinsic *multi-scale* information on the their shape. TDA is a rapidly developing field of mathematics aiming to leverage concepts of the well-established field of (algebraic) topology toward applications for real-world data sets and machine learning.

Topology studies invariant properties of (topological) spaces under homeomorphisms (i.e., continuous transformations); in the following, we restrict ourselves to topological manifolds, so as to simplify the exposition. A fundamental problem in topology is about classification: *How can two manifolds be distinguished from each other?* Algebraic topology (Bredon, 1993; Hatcher, 2000) provides sophisticated and powerful tools to study this question. The basic idea being to associate computable *algebraic structures* (e.g., groups or vector spaces) to a manifold that remain *invariant* under homeomorphisms. A very important class of algebraic invariants are the *homology groups*, which encode a great deal of information while still being efficiently computable

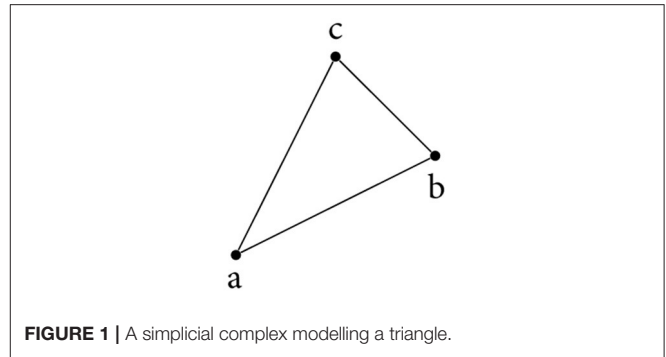


FIGURE 1 | A simplicial complex modelling a triangle.

in many cases. Homology groups arise from combinatorial representations of the manifold, the *chain complexes*.

2.1. Chain Complexes and Homology

The *standard k -simplex* Δ^k is defined as the convex hull of the standard basis vectors in \mathbb{R}^{k+1} , i.e.,

$$\Delta^k := \left\{ (x_0, \dots, x_k) \in \mathbb{R}^{k+1} \mid \sum_{i=0}^k x_i = 1, \quad x_i \geq 0 \quad \forall i \right\}.$$

Similarly, a general *k -simplex* $[v_0, \dots, v_k]$ is the convex hull of $k + 1$ affinely independent points v_0, \dots, v_k in a Euclidean space. Note that deleting one of the vertices v_i from a k -simplex $[v_0, \dots, v_k]$ yields a $(k - 1)$ -simplex $[v_0, \dots, \hat{v}_i, \dots, v_k]$ which is determined by the remaining vertices and called the *i -th face* of $[v_0, \dots, v_k]$. Simplices are the basic building blocks of chain complexes that are used in algebraic topology for the computation of homological invariants. Any *topological manifold* X can be topologically modelled using simplices (see Figure 1). A *singular k -simplex* in X is a continuous map $\sigma : \Delta^k \rightarrow X$. It is not required that σ is an embedding, for instance any constant map, mapping so a single point in X is a valid singular simplex. The inclusion of the i -th face of Δ^k is an important singular simplex in Δ^k , which we will denote by $F_i^k : \Delta^{k-1} \rightarrow \Delta^k$. To keep the exposition simple we will restrict ourselves to working over the two element field $\mathbb{F}_2 := \mathbb{Z}/2\mathbb{Z}$ in what follows. Given any space X , its *singular k -chains* are elements of the \mathbb{F}_2 -vector space $C_k(X)$ generated by the set of all singular k -simplices in X . Elements in $C_k(X)$ are thus “formal sums” of simplices. The *singular chain complex* $(C(X), \partial)$ of X is the sequence of spaces

$$\begin{aligned} \dots &\xrightarrow{\partial_{d+1}} C_d(X) \xrightarrow{\partial_d} C_{d-1}(X) \xrightarrow{\partial_{d-1}} \\ &\dots \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} 0, \end{aligned}$$

together with the *boundary maps* $\partial_k : C_k(X) \rightarrow C_{k-1}(X)$ given by

$$\partial_k(\sigma) := \sum_i \sigma \circ F_i^k$$

on the basis elements and extended linearly. A crucial property of the boundary maps is that they compose to 0, that is

$\partial_k \circ \partial_{k-1} = 0$. Elements of $Z_k(X) := \ker(\partial_k)$ are called k -cycles and those of $B_k(X) := \text{im}(\partial_{k+1})$ are called k -boundaries and their well-defined quotient

$$H_k(X) := Z_k(X)/B_k(X)$$

is the k -th singular homology group of X (despite the name, this is still technically a quotient vector space; however, the group-theoretical viewpoint is more convenient and prevalent in algebraic topology). The homology groups are *topological invariants*, i.e., they remain invariant under homeomorphisms and therefore encode intrinsic information on the topology of X . Thus, homology groups and simpler invariants derived from them, such as the *Betti-numbers* $\beta_k := \dim H_k(X)$, are useful in studying the classification question raised above. For example, the 0-th Betti number β_0 is a count of the connected components of a space, while β_1 is a count of the number of cycles.

2.1.1. Brief Example

Using the simplicial complex in **Figure 1**, we briefly illustrate some of the aforementioned concepts. Let $X = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$ be the representation of the simplicial complex. The *boundary* of the triangle is non-trivial, i.e., $\partial_2\{a, b, c\} = \{b, c\} + \{a, c\} + \{a, b\}$. The boundary of this chain of edges is trivial, though, because duplicate simplices cancel each other out. We get $\partial_1(\{b, c\} + \{a, c\} + \{a, b\}) = \{c\} + \{b\} + \{c\} + \{a\} + \{b\} + \{a\} = 0$, which is consistent with the property of compatible boundary maps to compose to 0. To compute $H_1(X) := Z_1(X)/B_1(X)$, we only have to calculate $Z_1(X)$; the boundary group $B_1(X)$ does not contain any non-trivial simplices because X does not contain any 2-simplices. By definition, $Z_1(X) = \ker(\partial_1) = \text{span}(\{a, b\} + \{b, c\} + \{a, c\})$. This is the *only* cycle in X (which we can easily verify either by inspection or based on combinatorics). Hence $H_1(X) = Z_1(X) = \mathbb{F}_2$ and $\beta_1 = 1$; the triangle therefore exhibits a single cycle, which aligns with our intuition.

2.2. Persistent Homology

Persistent homology (Edelsbrunner et al., 2000; Zomorodian and Carlsson, 2005) is the flagship tool of TDA. In the analysis of real-world data, it is typically not a priori clear at what *scale* interesting topological features occur. By using a filtration (connected to the scale parameter) persistent homology is able to capture topological changes across the whole range of scales and store this information in so-called persistence diagrams.

Persistent homology is an extension of homology to the setting of filtered chain complexes. A *filtered chain complex* is a (not-necessarily strictly) ascending sequence of chain complexes $C^{e_0} \subset C^{e_1} \subset C^{e_2} \subset \dots$ with inclusion maps $\iota^i: C^{e_i} \hookrightarrow C^{e_{i+1}}$ and $\iota^{ij} := \iota^j \circ \iota^{j-1} \circ \dots \circ \iota^i: C^{e_i} \hookrightarrow C^{e_j}$ for $i < j$. Filtered chain complexes naturally arise in situations where we have a sequence of inclusions of spaces $X^{e_0} \subset X^{e_1} \subset X^{e_2} \subset \dots$. Such cases, for instance, occur if we consider the sublevel sets $X^\epsilon := f^{-1}(\mathbb{R}_{\leq \epsilon})$ of a so-called *filtration* function $f: X \rightarrow \mathbb{R}$, or if we consider a point cloud Y in a metric space (M, d) and set

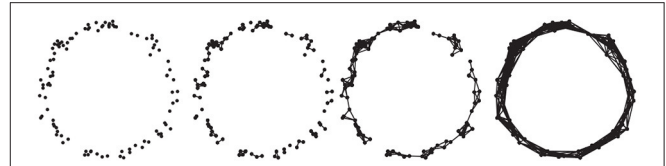


FIGURE 2 | Different stages of a Vietoris–Rips filtration for a simple “circle” point cloud. From left to right, connectivity of the underlying simplicial complex increases as ϵ increases.

$$Y^\epsilon := \bigcup_{y \in Y} B_\epsilon(y) = g^{-1}(\mathbb{R}_{\leq \epsilon})$$

with filtration function $g: M \rightarrow \mathbb{R}$ given by $g(m) := \inf_{y \in Y} d(m, y)$. Here $B_\epsilon(y)$ denotes the open ball of radius ϵ centred at y and we implicitly identify $\epsilon \simeq \epsilon'$ if X^ϵ (resp. Y^ϵ) is canonically homeomorphic to $X^{\epsilon'}$ (resp. $Y^{\epsilon'}$) for all $\delta \in [\epsilon, \epsilon']$. An important property of (singular) homology is that it is *functorial* (see e.g., Bredon, 1993), which implies that the inclusion maps ι^{ij} induce maps on the respective homology groups $H_k(\iota^{ij}): H_k(C^{e_i}) \rightarrow H_k(C^{e_j})$. **Figure 2** depicts the Vietoris–Rips complex construction based on a distance filtration, a standard construction in TDA. The k -th persistent homology groups are the images of these inclusions, that is

$$H_k^{ij} := \text{im } H_k(\iota^{ij}) = Z_k(C^{e_i}) / (B_k(C^{e_j}) \cap Z_k(C^{e_i})),$$

and thus precisely consist of the k -th homology classes of C^{e_i} that still exist after taking the inclusion $H_k(\iota^{ij})$. A homology class $\alpha \in H_k(C^{e_i})$ is said to be *born* at C^{e_i} if $\alpha \notin H_k^{i-1, i}$, i.e., if it is not in the image of $H_k(\iota^{i-1, i})$. If α is born at C^{e_i} , it is said to *die* at C^{e_j} if $H_k(\iota^{ij-1})(\alpha) \notin H_k^{i-1, j-1}$ and $H_k(\iota^{ij})(\alpha) \in H_k^{i-1, j}$. The *persistence* of α is given by $\epsilon_j - \epsilon_i$ and set to infinity if it never dies. The *persistent Betti-numbers*, defined by $\beta_k^{ij} := \dim H_k^{ij}$, carry information on how the homology (and thus the topology) changes across the filtration.

This information can be captured in a so-called *persistence diagram*, a multiset in $\mathbb{R}^2 := \mathbb{R}^2 \cup \mathbb{R} \times \{\infty\}$. Specifically, the persistence diagram of (homological) *dimension* k is given by the points $(\epsilon_i, \epsilon_j) \in \mathbb{R}^2$ with multiplicity

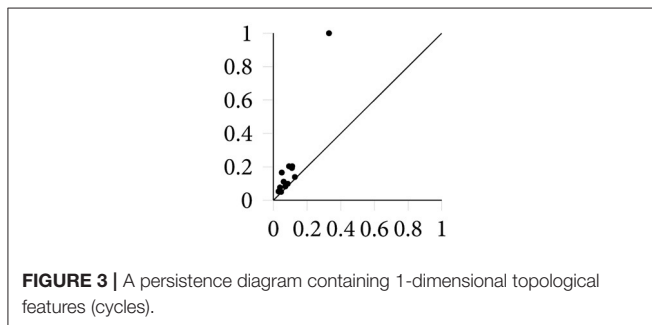
$$\mu_k^{ij} := (\beta_k^{ij-1} - \beta_k^{ij}) - (\beta_k^{i-1, j-1} - \beta_k^{i-1, j})$$

for all $i < j$. The multiplicity μ_k^{ij} counts the number of k -th homology classes that are born at C^{e_i} and die at C^{e_j} . **Figure 3** depicts a simple persistence diagram, calculated from the Vietoris–Rips complex in **Figure 2**. The axes of this diagram correspond to the ϵ values at which topological features are created and destroyed, respectively. The single point of high persistence corresponds to the primary topological feature of the point cloud, namely its circular shape. Other topological features occur at smaller scales—lower values of ϵ —and hence form a small dense cluster in the lower-left corner of the persistence diagram. The persistent Betti-numbers can be recovered from the persistence diagram itself; see Edelsbrunner and Harer, 2010.

A crucial fact that makes persistent homology valuable for application in data analysis is its *stability with respect to perturbations* of the filtration function. This means that persistent homology is robust to noise and constitutes an encoding of intrinsic topological properties of the data. More precisely, the space of persistence diagrams can be endowed with a metric induced by the *bottleneck distance* (or the *Wasserstein distances*) Edelsbrunner and Harer, 2010. A celebrated stability theorem (Cohen-Steiner et al., 2007) states that the L_∞ -distance of two real-valued functions f and g is an upper bound for the bottleneck distance W_∞ of their respective persistence diagrams \mathcal{D}_f and \mathcal{D}_g , i.e., $W_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq \|f - g\|_\infty$. The stability theorem and its variants (Skraba and Turner, 2020) are highly relevant for applications because they imply that the behaviour of persistent homology under noise is known; descriptors such as persistence diagrams change continuously as the input function is varied, and the “amplitude” of their change is bounded from above via the stability theorem.

3. SURVEY

This section comprises the main part of the paper, where we gather and discuss pertinent methods and tools in topological machine learning. We broadly group the methods into the following categories. First, in section 3.2, we discuss methods that deal with *extrinsic topological features*. By the qualification *extrinsic*, we mean that no analysis of the topology of the machine learning model or the neural network itself is incorporated.



These methods are instead mainly concerned with enabling the use of topological features, extracted from a given data set, in downstream machine learning models. This can be achieved through *vectorisation* of topological features or by designing specialised layers of neural networks that are capable of handling such features. Next, section 3.3 discusses *intrinsic topological features*. Those are methods that incorporate the topological analysis of aspects of the machine learning model itself. Whenever applicable, we further classify methods into *observational* and *interventional* methods. This sub-classification specifies *how* the methods are applied in a machine learning framework. Observational methods “observe” the topology of the data or model but they do not *directly* influence the model training or architecture. Interventional methods, by contrast, apply topological properties of the data, as well as *post-hoc* analysis of topological features of machine learning models, in order to inform the architectural design and/or model training. See **Figure 4** for an overview of the methods and their categories, as well as **Table 1** for the classification of all papers mentioned in this survey.

3.1. Limitations

Our paper selection is a cross-section over major machine learning conferences and machine learning journals. We refrain from comparing methods on certain tasks—such as classification—because there is considerable heterogeneity in the experimental setup, precluding a *fair* assessment of such methods.

3.2. Extrinsic Topological Features in Machine Learning

This section gives an overview of methods that aim at suitably representing topological features in order to use them as input features for machine learning models. We will refer to this class of methods as *extrinsic topological features in machine learning*, as they take topological information of the data sets into account, as opposed to intrinsic topological information of the machine learning framework itself (see section 3.3). A large class of such methods is comprised of *vectorisation* methods, that aim to transform persistent homology information into a feature vector form in order to make use of it in machine learning models.

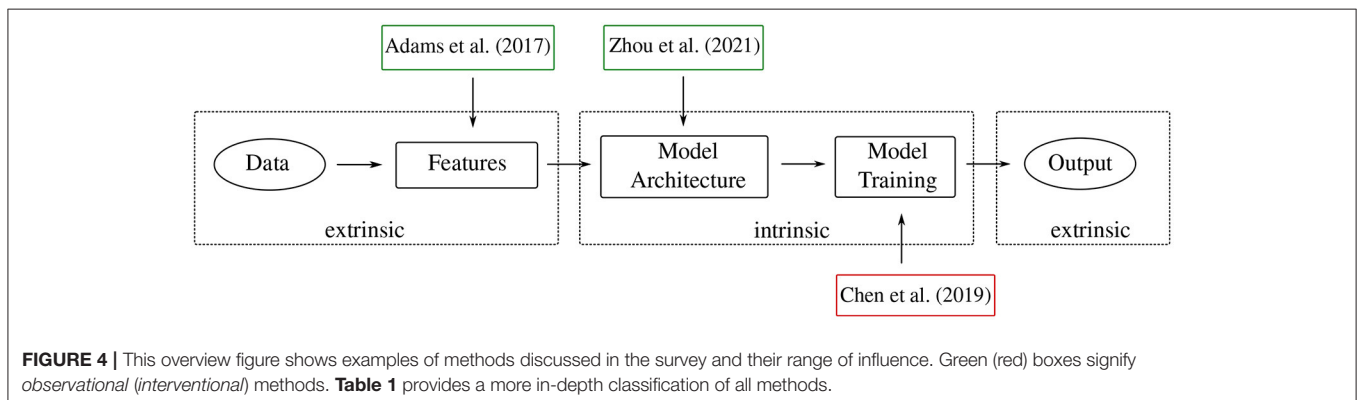


TABLE 1 | The categorisation of the approaches discussed in the present survey.

Extrinsic		Intrinsic	
Observational	Interventional	Observational	Interventional
Adams et al., 2017	Carrière et al., 2020	Gabrielsson and Carlsson, 2019	Chen et al., 2019
Bubenik, 2015	Kim et al., 2020	Khrulkov and Oseledets, 2018	Hofer et al., 2017
Carrière et al., 2015	Zhao and Wang, 2019	Zhou et al., 2021	Hofer C. et al., 2019
Carrière et al., 2017			Hofer et al., 2020a
Kusano et al., 2018			Hofer et al., 2020b
Reininghaus et al., 2015			Moor et al., 2020
Rieck et al., 2020a			Ramamurthy et al., 2019
Rieck et al., 2020b			Rieck et al., 2019b
Umeda, 2017			Zhao et al., 2020

It is interesting to note that intrinsic features tend to be used more in interventional settings, whereas extrinsic features remain observational for the most part.

However, alternative representations of topological descriptors, such as kernels or function-based representations, are also discussed in this section.

3.2.1. Vector-Based and Function-Based Representations

Persistence diagrams (see section 2) constitute useful descriptors of homological information of data. However, being multisets, they cannot be used *directly* as input data for machine learning models in the usual sense (recent paradigm shifts in machine learning, namely the introduction of *deep sets* (Zaheer et al., 2017), challenge this assumption somewhat, as we will later see in section 3.2.3). One first needs to suitably represent—or *vectorise*—persistence diagrams (PDs) in order to use them for downstream machine learning tasks. There are two predominant strategies for facilitating the integration of topological features into machine learning algorithms, namely (i) different representations that ideally give rise to feature vectors, and (ii) kernel-based methods that permit the integration into certain classifiers. Notice that these two strategies are not necessarily exclusionary; some representations, for example, also give rise to a kernel-based method.

Representations and kernel-based methods should ideally be efficiently computable, satisfy similar stability properties as the persistence diagrams themselves—hence exhibiting robustness properties with respect to noise—as well as provide some interpretable features. The stability of such representations is based on the fundamental stability theorem by Cohen-Steiner et al. (2007). In recent years, a multitude of suitable representation methods have been introduced; we present a selection thereof, focusing on representations that have already been used in machine learning contexts. As a somewhat broad categorisation, we observe that persistence diagrams are often mapped into an auxiliary *vector space*, e.g., by discretisation (Anirudh et al., 2016; Adams et al., 2017), or by mapping into a (Banach- or Hilbert-) *function space* (Chazal et al., 2014; Bubenik, 2015; Di Fabio and Ferri, 2015). Alternatively, there are several *kernel methods* (Reininghaus et al., 2015; Carrière et al., 2017; Kusano et al., 2018) that enable the efficient calculation of a similarity measure between persistence diagrams. Representations and kernel-based methods fall into the category

of what we denote “observational” methods. The only exception is given by PersLay (Carrière et al., 2020), which informs the layers of the model and thus is an “interventional” method.

Arguably the most simple form of employing topological descriptors in machine learning tasks uses *summary statistics*, such as the total persistence of a persistence diagram (Cohen-Steiner et al., 2010), its p -norm (Chen and Edelsbrunner, 2011), or its persistent entropy (Atienza et al., 2019), i.e., the Shannon entropy of the individual persistence values in a diagram. While all of these approaches result in scalar-valued summary statistics, they are often not directly applicable to complex machine learning tasks, which require more expressive representations. We note, however, that such statistics give rise to hypothesis testing (Blumberg et al., 2014) based on topological information and we envision that this field will become more prominent as topological features find their use case for data analysis. A simple and stable representation of persistence diagrams, suitable for machine learning tasks, is provided by what are commonly called *Betti curves*. Given a persistence diagram \mathcal{D} , and a weight function $w: \mathbb{R}^2 \rightarrow \mathbb{R}$, its Betti curve is the function $\beta: \mathbb{R} \rightarrow \mathbb{R}$ defined by

$$\beta(t) := \sum_{(b,d) \in \mathcal{D}} w(b,d) \cdot \mathbb{1}_{[b,d]}(t), \quad (1)$$

where

$$\mathbb{1}_{[b,d]}(t) := \begin{cases} 1, & \text{if } t \in [b,d] \\ 0, & \text{else} \end{cases} \quad (2)$$

is the indicator function. The Betti curve was often informally used to analyse data (Umeda, 2017); recently, Rieck et al. (2020a) provided a summarising description of their features. **Figure 5** depicts a simple illustration of the calculation of Betti curves. Betti curves are advantageous because they permit the calculation of a *mean curve*, next to providing an easy-to-evaluate distance and kernel method. Chevyrev et al. (2018) used this representation—and related “paths” derived from a persistence diagram and its representations—to solve classification tasks, using random forests and support vector machine classifiers. One

drawback of the Betti curves is their limited expressive power. Being a summary statistic of a persistence diagram, the mapping from a diagram to a curve is not injective; moreover, the curve only contains *counts* of topological features and does not permit tracking single features, for example.

A more fundamental technique, developed by Carrière et al. (2015), *directly* generates a high-dimensional feature vector from a persistence diagram. The main idea is to obtain a vector representation of some persistence diagram \mathcal{D} based on the distribution of pairwise distances of its elements, including points on the diagonal $\Delta := \{(x, x) \mid x \in \mathbb{R}\} \subset \mathbb{R}^2$. More precisely, for each pair (p, q) of points in \mathcal{D} , they compute $m(p, q) := \min\{d_\infty(p, q), d_\infty(p, \Delta), d_\infty(q, \Delta)\}$ and associate to \mathcal{D} the vector of these values, sorted in descending order. As persistence diagrams may be of different sizes, they enlarge each of these vectors by zeros so that its length matches the length of the longest vector in the set. Hence, the set of persistence diagrams one considers needs to be fixed a priori. This vectorisation does not necessarily scale well to large data sets, but it can provide a good baseline to furnish *any* machine learning classifier—including a neural network—with simple topology-based feature vectors. The use of this technique appears to be restricted at present; we hope that our article will help increase its adoption.

As a somewhat more complicated, but also more expressive, representation, Bubenik (2015) introduced topological descriptors called *persistence landscapes* that map persistence

diagrams into a (Banach or Hilbert) function space in an invertible manner that satisfies stability properties with respect to the bottleneck distance of PDs. The *persistence landscape* $\lambda: \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{R}$ of a PD $\mathcal{D} = \{(b_i, d_i)\}_{i \in I}$ can be defined in the following way. For $b < d$, we consider the auxiliary function $f_{(b,d)}(t) := \max\{0, \min\{t - b, d - t\}\}$ and define the persistence landscape as

$$\lambda(k, t) := \text{kmax}\{f_{(b_i, d_i)}(t)\}_{i \in I},$$

where kmax denotes the k -th largest element of the set. In addition to injectivity and stability, persistence landscapes do not require any choice of auxiliary parameters in their construction (see **Figure 6** for a depiction of the persistence landscape computation process). They also afford various summary statistics, such as a norm calculation as well the calculation of both a kernel and a distance measure, making them a versatile representation of topological features. While, persistence landscapes have seen applications in time series analysis (Stolz et al., 2017), their most successful integration into machine learning algorithms is provided in the form of a new *layer*: persistence landscapes form the basis of a robust (with respect to noise) topological layer for deep neural networks, which is differentiable with respect to its inputs, the so-called PLLay (persistence landscape based topological layer) established in Kim et al. (2020). This layer exhibits good performance in image classification tasks as well as orbit classification, where it is shown to provide new state-of-the-art performance. We note that persistence landscapes are often considered in a vectorised form, which is obtained through binning their domain. While this is possible and useful for certain applications, we want to stress that the persistence landscape, as a lossless representation, should ideally be treated as such. The calculation of persistence landscapes imposes additional computational complexity, but the empirical performance reported by Kim et al. (2020) suggests that the landscapes are well-suited as a feature descriptor.

The *persistence images* (PIs), introduced by Adams et al. (2017), constitute an elegant hierarchical vectorisation step, representing a PD as a vector through the following steps. First the PD \mathcal{D} is transformed from “birth–death”-coordinates into “birth–persistence”-coordinates via the transformation

$$T: \mathbb{R}^2 \longrightarrow \mathbb{R}^2: (x, y) \longmapsto (x, y - x).$$

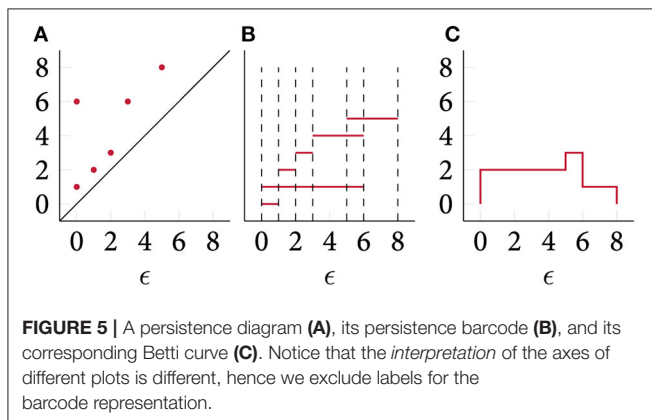


FIGURE 5 | A persistence diagram (A), its persistence barcode (B), and its corresponding Betti curve (C). Notice that the *interpretation* of the axes of different plots is different, hence we exclude labels for the barcode representation.

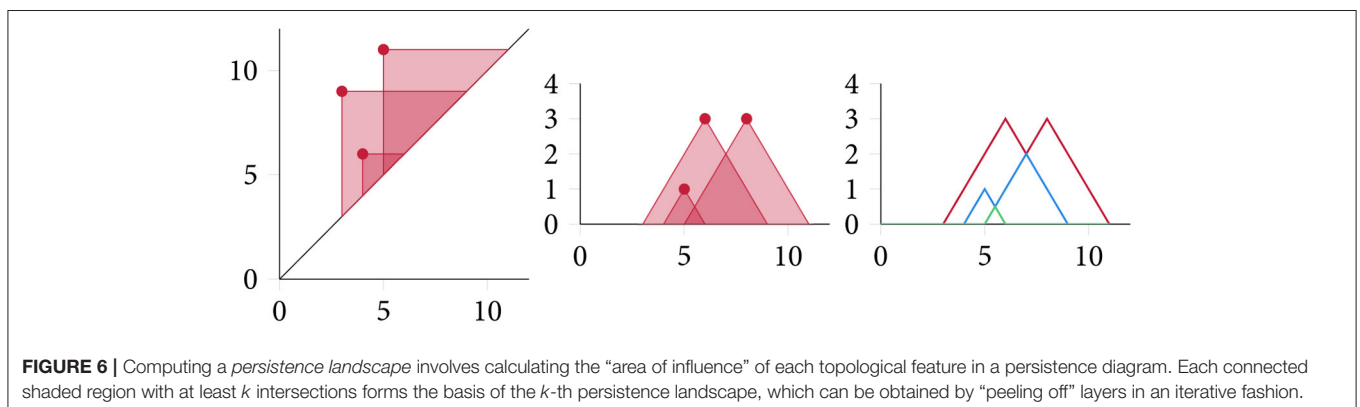


FIGURE 6 | Computing a *persistence landscape* involves calculating the “area of influence” of each topological feature in a persistence diagram. Each connected shaded region with at least k intersections forms the basis of the k -th persistence landscape, which can be obtained by “peeling off” layers in an iterative fashion.

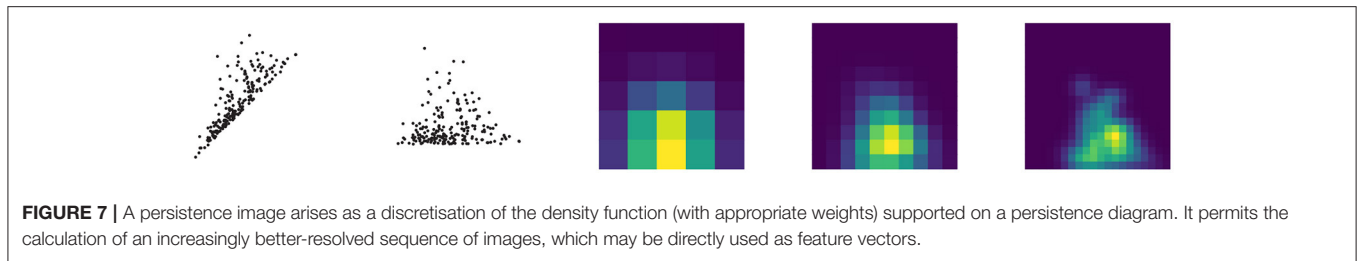


FIGURE 7 | A persistence image arises as a discretisation of the density function (with appropriate weights) supported on a persistence diagram. It permits the calculation of an increasingly better-resolved sequence of images, which may be directly used as feature vectors.

Next, for each $u \in \mathbb{R}^2$ a differentiable probability density ϕ_u on \mathbb{R}^2 is chosen (the standard choice being a normalised symmetric Gaussian with $\mathbb{E}[\phi_u] = u$), as well as a weighting function $f: \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}^2$ satisfying $f|_{\{0\} \times \mathbb{R}} \equiv 0$. Additionally one chooses a discretisation of a relevant subdomain of \mathbb{R}^2 by a standard grid. Each region R of this grid then corresponds to a pixel in the persistence image with value given by

$$\int_R \sum_{u \in T(\mathcal{D})} f(u) \phi_u(z) dz.$$

In the process of generating persistence images, there are three non-canonical choices to be made. First, the choice of the weighting function, which is often chosen to emphasise features in the PD with large persistence value, next the distributions ϕ_u , and lastly the resolution of the discretisation grid. Adams et al. (2017) prove that PIs are stable with respect to the 1-Wasserstein distance between persistence diagrams. **Figure 7** illustrates their calculation. Persistence images are highly flexible and are often employed to make a classifier “topology-aware” to some extent (Zhao and Wang, 2019; Carrière and Blumberg, 2020; Rieck et al., 2020b). A paper by Zhao and Wang (2019), for instance, showcases their utility for graph classification. Interestingly, this paper constitutes also one of the few interventional approaches that employ extrinsic topological features; specifically, the authors use pre-defined filtrations to obtain graph-based persistence diagrams, and learn task-based weights for individual “pixels” (or “cells”) in the diagram. This approach is seen to surpass several graph classification algorithms on standard benchmark data sets—a remarkable feat, considering that the method does not employ any label information. The main drawbacks of persistence images are their quadratic storage and computation complexity, as well as the choice of appropriate parameters. While recent work found them to be remarkably stable in practice with respect to the Gaussian kernel parameters (Rieck et al., 2020b), there are no guidelines for picking such hyperparameters, necessitating a (cross-validated) grid search, for instance.

3.2.2. Kernel-Based Representations

As an alternative to the previously-discussed representations, we now want to briefly focus on persistence diagrams again. The space of persistence diagrams can be endowed with metrics, such as the bottleneck distance. However, there is no natural Hilbert space structure on it, and such metrics tend to be computationally prohibitive or require the use of complex

approximation algorithms (Kerber et al., 2017). Kernel methods provide a way of implicitly introducing such a Hilbert space structure to which persistence diagrams can be mapped via the feature map of the kernel. This then allows for a downstream use in machine learning models. To be more specific, given a set X , a function $k: X \times X \rightarrow \mathbb{R}$ is called a (positive definite) *kernel* if there exists a Hilbert space \mathcal{H}_k together with a *feature map* $\phi: X \rightarrow \mathcal{H}_k$ such that $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}_k}$ for all $x_1, x_2 \in X$. Thus, by defining a kernel on the set of persistence diagrams, one obtains a vector representation via the feature map. However, in order for such a kernel to be useful in practice, it should additionally preserve the metric stability properties of persistence diagrams. Some pertinent examples of the kernel method are the following. Reininghaus et al. (2015) define a kernel on the set of persistence diagrams that is stable with respect to the 1-Wasserstein distance (Villani, 2009). The kernel is based on the idea of heat diffusion on a persistence diagram and offers a feature map that can be discretised (in fact, there are interesting similarities to persistence images). It was subsequently shown to satisfy *universality* (Kwitt et al., 2015), a desirable property for a kernel to have because it implies suitability for hypothesis testing. The *sliced Wasserstein kernel*, which is metric-preserving, was introduced by Carrière et al. (2017). It is based on the idea of the sliced Wasserstein distance (Kolouri et al., 2016), which ensures positive definiteness of the kernel through low-dimensional projections. Kusano et al. (2018) propose *persistence weighted Gaussian kernels* that incorporate a weighting and satisfy stability results with respect to the bottleneck distance and the 1-Wasserstein distance. The expressive power of kernels is in contrast to their computational complexity. Naïve implementations scale quadratically in the number of points, thus impeding the use of kernels for persistence diagrams with a large number of points. Some mitigation strategies exist (Greengard and Strain, 1991; Rahimi and Recht, 2008), but have not been adopted by implementations so far (moreover, their use is not always applicable, necessitating additional research). Nevertheless, such kernels are attractive because they are *not* limited with respect to the input data. Most of the papers exhibit good performance for shape classification or segmentation tasks, as well as in orbit classification.

While most of the aforementioned kernels are used to directly compare persistence diagrams, there are also examples of kernels *based on* topological information. An interesting example is provided by Rieck et al. (2019a), who introduce the Persistent Weisfeiler–Lehman (P-WL) kernel for graphs. It computes topological features during a Weisfeiler–Lehman (WL) procedure. The WL procedure refers to an iterative scheme in

which vertex label information is aggregated over the neighbours of each vertex, resulting in a label multiset. A perfect hashing scheme is now applied to every multiset and the graph is relabelled with the ensuing hashes. This process can be repeated until a pre-defined limit has been reached or until the labels do not change any more. While originally intended as a test for graph isomorphism, it turns out that there are non-isomorphic graphs that cannot be distinguished by the WL procedure. However, it turns out to be an exceptionally useful way of assessing the dissimilarity between two graphs in polynomial time, leading to the WL kernel framework (Shervashidze and Borgwardt, 2009; Shervashidze et al., 2011), which enjoys great popularity for graph learning tasks (Borgwardt et al., 2020; Kriege et al., 2020). The P-WL extension of WL is characterised by its capability to extract topological information of the graph with respect to the current node labelling for each WL iteration. This kernel is particularly notable since it constitutes the first (to our knowledge) method that imbues data-based labels into the calculation of persistent homology.

3.2.3. Integrating Topological Descriptors Into Neural Networks

One of the seminal methods that built a bridge between modern machine learning techniques and TDA is a work by Hofer et al. (2017). Using a differentiable projection function for persistence diagrams (with learnable parameters), the authors demonstrate that persistence diagrams of a data set can be easily integrated into *any* deep learning architecture. While the primary focus of the paper lies on developing such a projection function, the authors demonstrate the general feasibility of topological descriptors in both shape and graph classification tasks. A follow-up publication (Hofer C. D. et al., 2019) discusses more theoretical requirements for learning representations of topological descriptors.

This approach, as well as the development of the “DeepSets” architecture (Zaheer et al., 2017), which makes deep learning methods capable of learning *sets*, i.e., unordered sequences of varying cardinalities, spurred the development of *layers* that can be easily integrated into a deep learning workflow. An excellent example of such a layer is Carrière et al. (2020), which employs extended persistence (Cohen-Steiner et al., 2009) and heat kernel signatures to *learn* a vectorisation of persistence diagrams suited to the learning task at hand. PersLay is a neural network layer, defined by

$$\text{PersLay}(\mathcal{D}) := \text{op}(\{w(p) \cdot \phi(p)\}_{p \in \mathcal{D}}),$$

where \mathcal{D} is a persistence diagram, op is an permutation invariant mapping, $w: \mathbb{R}^2 \rightarrow \mathbb{R}$ is a weight function and $\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^d$ is a vector representation function. Its generic definition allows PersLay to subsume and recover many existing representations by appropriate choices of op and ϕ (Carrière et al., 2020).

3.3. Intrinsic Topological Features in Machine Learning

This section reviews methods that either incorporate topological information directly into the design of a machine learning model

itself, or leverage topology to study aspects of such a model. We refer to such features as *intrinsic topological features*. The primary examples are regularisation techniques as well as techniques for analysing neural network architectures.

3.3.1. Regularisation Techniques

As a recent example, Moor et al. (2020) propose a topological autoencoder, which aims to preserve topological features of the input data in low-dimensional representations. This is achieved via a regularisation term that incentivises the persistence diagrams of both the latent and input space to be topologically similar. This method acts on the level of mini-batches, treating each of them as a point cloud. Persistence diagrams are obtained from the Vietoris–Rips complex of each space. By tracking the simplices that are relevant for the creation and destruction of topological features, and by consistently mapping simplices to a given edge in the Vietoris–Rips complex, each filtration can be interpreted as a selection of distances from the full distance matrix of the point cloud. The proposed regularisation term then compares the “selected” distances in the data space with the corresponding distances in the latent space (and vice versa). Finally, this regularisation is differentiable under the assumption that the persistence diagram is discrete (i.e., for each of its points, there is an infinitesimal neighbourhood containing no other points). The scheme can thus be directly integrated into the end-to-end training of an autoencoder, making it aware of the topology in the data space. This work can also be considered as an extension of previous work by Hofer C. et al. (2019), who introduced a differentiable loss term for one-class learning that controls the topology of the latent space; in effect, their loss term enforces a preferred “scale” for topological features in the latent space. It does not have to harmonise topological features *across* different spaces. It turns out that an autoencoder trained with this loss term on unlabelled data can be used on other data sets for one-class learning. This hints at the fact that enforcing a certain topological structure can be beneficial for learning tasks; we will later see that such empirical observations can also be furnished with a theoretical underpinning.

An approach by Chen et al. (2019) takes a different perspective. The authors develop a measure of the *topological complexity* (in terms of connected components) of the classification boundary of a given classifier. Said topological information is then used for regularisation in order to force the topological complexity of the decision boundary to be simpler, containing fewer features of low persistence. Thus, topological information serves as a penalty during classification such that training the classifier itself can be improved. In contrast to the aforementioned approach, differentiability is obtained through a “surrogate” piecewise linear approximation of the classifier. The method is seen to yield competitive results and the authors observe that the method performs well even in the presence of label noise. Analysing the decision boundary of a classifier also turns out to be advantageous for *model selection*, as we will later see in section 3.3.2.

Hofer et al. (2020a) analyse more fundamental principles of regularisation by means of topological features. Specifically, they study regularisation in a regime of small sample sizes

with over-parametrised neural networks. By developing a new topological constraint for per-class probability measures, mass concentration effects in the vicinity of the learned representations of training instances are observed, leading to overall improvements of generalisation performance. The authors observe that controlling topological properties of learned representations presents numerous avenues for future research. These theoretical findings validate the empirical improvements observed in previous works of this domain.

As a more involved example of methods that make use of intrinsic features, Zhao et al. (2020) include topological features of graph neighbourhoods into a standard graph neural network (GNN) architecture. Their method combines a shortest-path filtration with persistence images, which are subsequently compressed to a single scalar value using a multilayer perceptron. The resulting scalar is then used to re-weight the message passing scheme used in training the GNN, thus obtaining topologically-based representations of graph neighbourhoods. In contrast to the previously-described loss terms, this method is not end-to-end differentiable, though, because the conversion from persistence diagrams to persistence images involves non-continuous parameters, i.e., the image dimensions. Zhao et al. (2020) primarily propose this method for node classification tasks, but we hypothesise that other graph tasks would profit from the integration of topological features.

Last, to provide a somewhat complementary perspective to preceding work, a paper by Hofer et al. (2020b) discusses how to employ graph neural networks (GNNs) to *learn* an appropriate filtration in an end-to-end fashion. The authors demonstrate that a GNN can be used to successfully initialise a scalar-valued filtration function, which can then subsequently be trained under mild assumptions (specifically, injectivity at the vertices of the graph needs to hold). The learned filtration turns out to surpass fixed filtrations combined with a persistent homology baseline, thus demonstrating the benefits of making topological representations differentiable—and thus *trainable*.

3.3.2. Model Analysis

Shifting our view from regularisation techniques, topological analysis has been applied to evaluate generative adversarial networks (GANs). A GAN (Goodfellow et al., 2014) is comprised of two sub-networks, a generator and a discriminator. Given a data distribution P_{data} , the generator's objective is to learn a distribution P_{model} with the same statistics, whereas the discriminator learns to distinguish generated samples from actual data samples. The topological evaluation of GANs is motivated by the manifold hypothesis (Fefferman et al., 2013), which poses that a data distribution P_{data} is sampled from an underlying manifold $\mathcal{M}_{\text{data}}$. The idea is to assess the topological similarity of $\mathcal{M}_{\text{data}}$ and the underlying manifold $\mathcal{M}_{\text{model}}$ of the model generated distribution P_{model} . Based on the persistent homology of witness complexes, Khrulkov and Oseledets (2018) introduce the *Geometry Score*, which is a similarity measure of the topologies of $\mathcal{M}_{\text{data}}$ and $\mathcal{M}_{\text{model}}$ and can be used to evaluate generative models. Later work by Zhou et al. (2021) generalises this approach and additionally extends it to the disentanglement evaluation of generative models in unsupervised settings.

In a different direction, the topological analysis of the intrinsic structure of a classifier, such as a neural network, makes it possible to improve a variety of tasks. This includes the analysis of training behaviour as well as model selection—or *architecture selection* in the case of neural networks.

While the literature dedicated to the better understanding of deep neural networks has typically focused on its functional properties, Rieck et al. (2019b) took a different perspective to focus on the graph structure of a neural network. Specifically, they treat a (feed-forward) neural network as a stack of bipartite graphs. From this view, they propose “neural persistence,” a complexity measure which summarizes topological features that arise when calculating a filtration of the neural network graph where the filtration weights are given by the network parameters. They showed that neural persistence can distinguish between well-trained and badly-trained (i.e., diverged) networks. This measure is oblivious to the functional behaviour of the underlying network, but only focuses on its (weighted) *structure*. Nevertheless, Rieck et al. (2019b) showed that it can be used for guiding early stopping solely based on topological properties of the neural network, potentially saving validation data used for the early stopping decision.

Ramamurthy et al. (2019) employ labelled variants of simplicial complexes, such as a labelled Vietoris–Rips complex, to analyse the decision boundary (i.e., classification boundary) of a given classifier. The authors are able to provide theoretical guarantees that the correct homology of a decision boundary can be recovered from samples, thus paving the way for an efficient approximation scheme that incorporates local scale estimates of the data set. Such a construction is required because the density of available samples is not guaranteed to be uniform, leading to simplicial complexes with spurious simplices in high-density regions, while running the risk of “undersampling” low-density regions. Next to “matching” models based on the *Decision Boundary Topological Complexity* (DBTC) score, Ramamurthy et al. (2019) also enable matching data sets to pre-trained models. The underlying assumption is that a model that closely mimics the topological complexity of a data set is presumably a better candidate for this particular data set.

Gabrielsson and Carlsson (2019) utilise topological data analysis to analyse topological information encoded in the weights of convolutional neural networks (CNNs). They show that the weights of convolutional layers encode simple global structures which dynamically change during training of the network and correlate with the network's ability to generalise to unseen data. Moreover, they find that topological information on the trained weights of a network can lead to improvements in training efficiency and reflect the generality of the data set on which the training was performed.

4. OUTLOOK AND CHALLENGES

This survey provided a glimpse of the nascent field of *topological machine learning*. We categorised existing work depending on its intention (interventional vs. observational) and according to what type of topological features are being

calculated (extrinsic vs. intrinsic), finding that most extrinsic approaches are observational, i.e., they do not inform the choice of model afterwards, while most intrinsic approaches are interventional, i.e., they result in changes to the choice of model or its architecture.

Numerous avenues for future research exist. Of the utmost importance is the improvement of the “software ecosystem.” Software libraries such as GUDHI (Maria et al., 2014) and *giotto-tda* (Tauzin et al., 2021) are vital ingredients for increasing the adoption of TDA methods, but we envision that there is a specific niche for libraries that integrate *directly* with machine learning frameworks such as *pytorch*. This will make it easier to disseminate knowledge and inspire more research. A challenge that the community yet has to overcome involves the overall scalability of methods, though. While certain improvements on the level of filtrations are being made (Sheehy, 2013; Cavanna et al., 2015), those improvements have yet to be integrated into existing algorithms. A more fundamental question is to what extent TDA has to rely on “isotropic” complexes such as the Vietoris–Rips complex, and whether scale-dependent complexes that incorporate sparsity can be developed.

On the side of applications, we note that several papers already target problems such as graph classification, but they are primarily based on fixed filtrations (with the notable exception of Hofer et al. (2020b), who learn a filtration end-to-end). We envision that future work could target more involved scenarios, such as the creation of “hybrid” GNNs, and the use of end-to-end differentiable features for other graph tasks, such as node classification, link prediction, or community detection.

REFERENCES

- Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., et al. (2017). Persistence images: a stable vector representation of persistent homology. *J. Mach. Learn. Res.* 18, 1–35.
- Anirudh, R., Venkataraman, V., Ramamurthy, K. N., and Turaga, P. (2016). “A Riemannian framework for statistical analysis of topological persistence diagrams,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1023–1031. doi: 10.1109/CVPRW.2016.132
- Atienza, N., Gonzalez-Diaz, R., and Rucco, M. (2019). Persistent entropy for separating topological features from noise in vietoris-rips complexes. *J. Intell. Inform. Syst.* 52, 637–655. doi: 10.1007/s10844-017-0473-4
- Blumberg, A. J., Gal, I., Mandell, M. A., and Pancia, M. (2014). Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. *Found. Comput. Math.* 14, 745–789. doi: 10.1007/s10208-014-9201-4
- Borgwardt, K., Ghisu, E., Llinares-Lopez, F., O’Bray, L., and Rieck, B. (2020). Graph kernels: state-of-the-art and future challenges. *Found. Trends Mach. Learn.* 13, 531–712. doi: 10.1561/22000000076
- Bredon, G. E. (1993). *Topology and Geometry, Volume 139 of Graduate Texts in Mathematics*. New York, NY: Springer-Verlag. doi: 10.1007/978-1-4757-6848-0
- Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.* 16, 77–102.
- Carrière, M., and Blumberg, A. (2020). “Multiparameter persistence image for topological machine learning,” in *Advances in Neural Information Processing Systems*, Vol. 33, eds H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc.), 22432–22444.
- Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., and Umeda, Y. (2020). “PersLay: a neural network layer for persistence diagrams and new graph topological signatures,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, eds S. Chiappa and R. Calandra (PMLR), 2786–2796.
- Carrière, M., Cuturi, M., and Oudot, S. (2017). “Sliced Wasserstein kernel for persistence diagrams,” in *Proceedings of the 34th International Conference on Machine Learning* eds D. Precup and Y. W. The (Sydney, NSW: International Convention Centre; PMLR), 664–673.
- Carrière, M., Oudot, S., and Ovsjanikov, M. (2015). Stable topological signatures for points on 3D shapes. *Comput. Graph. Forum* 34, 1–12. doi: 10.1111/cgf.12692
- Cavanna, N. J., Jahanseir, M., and Sheehy, D. R. (2015). “A geometric perspective on sparse filtrations,” in *Proceedings of the Canadian Conference on Computational Geometry* (Kingston, ON).
- Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., and Wasserman, L. (2014). “Stochastic convergence of persistence landscapes and silhouettes,” in *Proceedings of the Thirtieth Annual Symposium on Computational Geometry, SOCG’14* (New York, NY: Association for Computing Machinery), 474–483. doi: 10.1145/2582112.2582128
- Chen, C., and Edelsbrunner, H. (2011). “Diffusion runs low on persistence fast,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Red Hook, NY: Curran Associates, Inc.), 423–430. doi: 10.1109/ICCV.2011.6126271
- Chen, C., Ni, X., Bai, Q., and Wang, Y. (2019). “A topological regularizer for classifiers via persistent homology,” in *Proceedings of Machine Learning Research*, eds K. Chaudhuri and M. Sugiyama (PMLR), 2573–2582.
- Chevyrev, I., Nanda, V., and Oberhauser, H. (2018). Persistence paths and signature features in topological data analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 192–202. doi: 10.1109/TPAMI.2018.2885516
- Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. (2007). Stability of persistence diagrams. *Discrete Comput. Geom.* 37, 103–120. doi: 10.1007/s00454-006-1276-5

As another upcoming topic, we think that the analysis of time-varying data sets using topology-based methods is long overdue. With initial work by Cohen-Steiner et al. (2006) on time-varying topological descriptors providing a theoretical foundation, there are nevertheless few topology-based approaches that address time series classification or time series analysis. Several—theoretical and practical—aspects for such an endeavour are addressed by Perea et al. (2015), who develop a persistence-based method for quantifying periodicity in time series. The method is based on the fundamental embedding theorem by Takens (1981) and is combined with a sliding window approach. Future work could build on such approaches, or find other ways to characterise time-series, for instance based on complex networks (Lacasa et al., 2008). This could pave the road toward novel applications of TDA such as anomaly detection.

AUTHOR CONTRIBUTIONS

FH, MM, and BR performed the literature search and revised the draft. FH and BR drafted the original manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was partially funded and supported by the Swiss National Science Foundation [Spark grant 190466, FH and BR]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

- Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. (2009). Extending persistence using Poincaré and Lefschetz duality. *Found. Comput. Math.* 9, 79–103. doi: 10.1007/s10208-008-9027-z
- Cohen-Steiner, D., Edelsbrunner, H., Harer, J., and Mileyko, Y. (2010). Lipschitz functions have L_p -stable persistence. *Found. Comput. Math.* 10, 127–139. doi: 10.1007/s10208-010-9060-6
- Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. (2006). “Vines and vineyards by updating persistence in linear time,” in *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry, SCG '06* (New York, NY: Association for Computing Machinery), 119–126. doi: 10.1145/1137856.1137877
- Di Fabio, B., and Ferri, M. (2015). “Comparing persistence diagrams through complex vectors,” in *Image Analysis and Processing – ICIAP 2015*, eds V. Murino and E. Puppo (Cham: Springer International Publishing), 294–305. doi: 10.1007/978-3-319-23231-7_27
- Edelsbrunner, H., and Harer, J. (2010). *Computational Topology: An Introduction*. Providence, RI: American Mathematical Society. doi: 10.1090/mbk/069
- Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). “Topological persistence and simplification,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science* (Redondo Beach, CA), 454–463. doi: 10.1109/SFCS.2000.892133
- Fefferman, C., Mitter, S., and Narayanan, H. (2013). Testing the manifold hypothesis. *J. Am. Math. Soc.* 29, 983–1049. doi: 10.1090/jams/852
- Gabrielsson, R. B., and Carlsson, G. (2019). “Exposition and interpretation of the topology of neural networks,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1069–1076. doi: 10.1109/ICMLA.2019.00180
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Vol. 27, eds Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc.).
- Greengard, L., and Strain, J. (1991). The fast Gauss transform. *SIAM J. Sci. Stat. Comput.* 12, 79–94. doi: 10.1137/0912004
- Hatcher, A. (2000). *Algebraic Topology*. Cambridge: Cambridge University Press.
- Hofer, C., Graf, F., Niethammer, M., and Kwitt, R. (2020a). “Topologically densified distributions,” in *Proceedings of the 37th International Conference on Machine Learning*, eds H. Daumé III and A. Singh (PMLR), 4304–4313.
- Hofer, C., Graf, F., Rieck, B., Niethammer, M., and Kwitt, R. (2020b). “Graph filtration learning,” in *Proceedings of the 37th International Conference on Machine Learning*, eds H. Daumé III and A. Singh (PMLR), 4314–4323.
- Hofer, C., Kwitt, R., Niethammer, M., and Dixit, M. (2019). “Connectivity-optimized representation learning via persistent homology,” in *Proceedings of the 36th International Conference on Machine Learning*, eds K. Chaudhuri and R. Salakhutdinov (PMLR), 2751–2760.
- Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. (2017). “Deep learning with topological signatures,” in *Advances in Neural Information Processing Systems*, Vol. 30, eds I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc.).
- Hofer, C. D., Kwitt, R., and Niethammer, M. (2019). Learning representations of persistence barcodes. *J. Mach. Learn. Res.* 20, 1–45.
- Kerber, M., Morozov, D., and Nigmatov, A. (2017). Geometry helps to compare persistence diagrams. *ACM J. Exp. Algorith.* 22. doi: 10.1145/3064175
- Khrulkov, V., and Oseledets, I. (2018). “Geometry score: a method for comparing generative adversarial networks,” in *Proceedings of the 35th International Conference on Machine Learning*, eds J. Dy and A. Krause (Stockholm: PMLR), 2621–2629.
- Kim, K., Kim, J., Zaheer, M., Kim, J., Chazal, F., and Wasserman, L. (2020). “PLay: efficient topological layer based on persistent landscapes,” in *Advances in Neural Information Processing Systems*, Vol. 33, eds H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc.), 15965–15977.
- Kolouri, S., Zou, Y., and Rohde, G. K. (2016). “Sliced Wasserstein kernels for probability distributions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5258–5267. doi: 10.1109/CVPR.2016.568
- Kriege, N. M., Johansson, F. D., and Morris, C. (2020). A survey on graph kernels. *Appl. Netw. Sci.* 5:6. doi: 10.1007/s41109-019-0195-3
- Kusano, G., Fukumizu, K., and Hiraoka, Y. (2018). Kernel method for persistence diagrams via kernel embedding and weight factor. *J. Mach. Learn. Res.* 18, 1–41.
- Kwitt, R., Huber, S., Niethammer, M., Lin, W., and Bauer, U. (2015). “Statistical topological data analysis—a kernel perspective,” in *Advances in Neural Information Processing Systems*, Vol. 28, eds C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc.).
- Lacasa, L., Luque, B., Ballesteros, F., Luque, J., and Nuno, J. C. (2008). From time series to complex networks: the visibility graph. *Proc. Natl. Acad. Sci. U.S.A.* 105, 4972–4975. doi: 10.1073/pnas.0709247105
- Maria, C., Boissonnat, J.-D., Glisse, M., and Yvinec, M. (2014). “The GUDHI library: simplicial complexes and persistent homology,” in *Mathematical Software-ICMS 2014*, eds H. Hong and C. Yap (Berlin; Heidelberg: Springer), 167–174. doi: 10.1007/978-3-662-44199-2_28
- Moor, M., Horn, M., Rieck, B., and Borgwardt, K. (2020). “Topological autoencoders,” in *Proceedings of the 37th International Conference on Machine Learning*, eds H. Daumé III and A. Singh (PMLR), 7045–7054.
- Perea, J., Deckard, A., Haase, S., and Harer, J. (2015). SW1PerS: sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC Bioinformatics* 16:257. doi: 10.1186/s12859-015-0645-6
- Rahimi, A., and Recht, B. (2008). “Random features for large-scale kernel machines,” in *Advances in Neural Information Processing Systems*, Vol. 20, eds J. Platt, D. Koller, Y. Singer, and S. Roweis (Curran Associates, Inc.).
- Ramamurthy, K. N., Varshney, K., and Mody, K. (2019). “Topological data analysis of decision boundaries with application to model selection,” in *Proceedings of the 36th International Conference on Machine Learning*, eds K. Chaudhuri and R. Salakhutdinov (PMLR), 5351–5360.
- Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. (2015). “A stable multi-scale kernel for topological machine learning,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4741–4748. doi: 10.1109/CVPR.2015.7299106
- Rieck, B., Bock, C., and Borgwardt, K. (2019a). “A persistent Weisfeiler-Lehman procedure for graph classification,” in *International Conference on Machine Learning*, eds K. Chaudhuri and R. Salakhutdinov (PMLR), 5448–5458.
- Rieck, B., Sadlo, F., and Leitte, H. (2020a). “Topological machine learning with persistence indicator functions,” in *Topological Methods in Data Analysis and Visualization V*, eds H. Carr, I. Fujishiro, F. Sadlo, and S. Takahashi (Cham: Springer), 87–101. doi: 10.1007/978-3-030-43036-8_6
- Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., et al. (2019b). “Neural persistence: a complexity measure for deep neural networks using algebraic topology,” in *International Conference on Learning Representations*.
- Rieck, B., Yates, T., Bock, C., Borgwardt, K., Wolf, G., Turk-Browne, N., et al. (2020b). “Uncovering the topology of time-varying fMRI data using cubical persistence,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33, eds H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc.), 6900–6912.
- Sheehy, D. R. (2013). Linear-size approximations to the Vietoris-Rips filtration. *Discrete Comput. Geom.* 49, 778–796. doi: 10.1007/s00454-013-9513-1
- Shervashidze, N., and Borgwardt, K. (2009). “Fast subtree kernels on graphs,” in *Advances in Neural Information Processing Systems*, Vol. 22, eds Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Inc.), 1660–1668.
- Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-Lehman graph kernels. *J. Mach. Learn. Res.* 12, 2539–2561.
- Skraba, P., and Turner, K. (2020). Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*.
- Stolz, B. J., Harrington, H. A., and Porter, M. A. (2017). Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos* 27:047410. doi: 10.1063/1.4978997
- Takens, F. (1981). “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence, Warwick 1980 (Coventry, 1979/1980)*, eds D. Rand and L. S. Young (Berlin; New York, NY: Springer), 366–381. doi: 10.1007/BFb0091924
- Tauzin, G., Lupo, U., Tunstall, L., Perez, J. B., Caorsi, M., Medina-Mardones, A. M., et al. (2021). giotto-tda: a topological data analysis toolkit for machine learning and data exploration. *J. Mach. Learn. Res.* 22, 1–6.
- Umeda, Y. (2017). Time series classification via topological data analysis. *Trans. Jpn. Soc. Artif. Intell.* 32, 1–12. doi: 10.1527/tjsai.D-G72

- Villani, C. (2009). *Optimal Transport, Volume 338 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Berlin: Springer-Verlag. doi: 10.1007/978-3-540-71050-9
- Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). “Deep sets,” in *Advances in Neural Information Processing Systems*, Vol. 30, eds I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc.).
- Zhao, Q., and Wang, Y. (2019). “Learning metrics for persistence-based summaries and applications for graph classification,” in *Advances in Neural Information Processing Systems*, Vol. 32, eds H. Wallach, H. Larochelle, A. Beygelzimer, Florence d’Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc.).
- Zhao, Q., Ye, Z., Chen, C., and Wang, Y. (2020). “Persistence enhanced graph neural network,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, eds S. Chiappa and R. Calandra (PMLR), 2896–2906.
- Zhou, S., Zelikman, E., Lu, F., Ng, A. Y., Carlsson, G. E., and Ermon, S. (2021). “Evaluating the disentanglement of deep generative models through manifold topology,” in *International Conference on Learning Representations*.
- Zomorodian, A., and Carlsson, G. (2005). Computing persistent homology. *Discrete Comput. Geom.* 33, 249–274. doi: 10.1007/s00454-004-1146-y
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Hensel, Moor and Rieck. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.