Artificial Intelligence Review (2024) 57:77
https://doi.org/10.1007/s10462-024-10710-9

Topological deep learning: a review of an emerging paradigm

Ali Zia[1,2] · Abdelwahed Khamis[2] · James Nichols[1] · Usman Bashir Tayab[3] · Zeeshan Hayder[2] · Vivien Rolland[2] · Eric Stone[1] · Lars Petersson[2]

Accepted: 18 January 2024 / Published online: 29 February 2024

Abstract
Topological deep learning (TDL) is an emerging area that combines the principles of *Topological data analysis* (TDA) with deep learning techniques. TDA provides insight into data shape; it obtains global descriptions of multi-dimensional data whilst exhibiting robustness to deformation and noise. Such properties are desirable in deep learning pipelines, but they are typically obtained using non-TDA strategies. This is partly caused by the difficulty of combining TDA constructs (e.g. barcode and persistence diagrams) with current deep learning algorithms. Fortunately, we are now witnessing a growth of deep learning applications embracing topologically-guided components. In this survey, we review the nascent field of topological deep learning by first revisiting the core concepts of TDA. We then explore how the use of TDA techniques has evolved over time to support deep learning frameworks, and how they can be integrated into different aspects of deep learning. Furthermore, we touch on TDA usage for analyzing existing deep models; *deep topological analytics*. Finally, we discuss the challenges and future prospects of topological deep learning.

Keywords Persistent homology · Deep learning · Topological data analysis · TDA representations

1 Introduction

In this article, we explore the growing interface between deep learning and topology. We examine deep learning methods that make use of topological information to understand the shape of data, as well as the use of deep learning in calculating topological signatures. We broadly refer to this intersection of fields as *topological deep learning*. The advancements in topological deep learning have been enabled by the development of *topological data analysis* (TDA) over the last two decades.

TDA is a relatively recent amalgam of theory and algorithms that aims to obtain a geometric and topological understanding of data from real-world applications. The approach to data employed in TDA fundamentally differs from that in statistical learning. Rather than finding summary statistics, estimators, fitting approximate distributions, clustering,

Extended author information available on the last page of the article or training neural nets, TDA instead seeks to understand the properties of the geometric object, often a *manifold*, on which the data resides. This reflects the common intuition that data tends to lie on, or close to, a lower-dimensional manifold that is embedded in high-dimensional feature space. In this article, we sometimes refer to this as the *data manifold*.

The main goal of TDA is to infer information about the global structure of the data manifold, such as its connectivity and the presence of multi-dimensional holes. In the pure mathematical setting, this information is characterized by the *persistence homology* and the related concept of *Betti numbers*, that counts the number of n-dimensional holes in a manifold. With a finite set of data points, the Betti numbers are unavailable, but TDA employs various substitutes such as *persistence diagrams* and the *barcode*. An important property of the topological information obtained is its invariance to continuous deformation and scaling. This property also lends itself to robustness against perturbation and noise. Another benefit is the versatility of the TDA methods, owed mostly to the abstract origins of algebraic topology. The methods are applicable to a wide variety of data types and objects. This includes point cloud data in Euclidean spaces, categorical data, or the analysis of images and functions. TDA is backed by explainable theory but lacks the learning ability and other practical aspects of deep neural networks. Conversely, neural networks suffer from the need for large training datasets and billions of tunable parameters. Due to these aspects, integration of TDA with deep neural networks poses a number of challenges.

Despite much recent activity in co-opting topological approaches in deep learning, what the leading approach should be remains unclear, mostly because of computational and theoretical concerns. The TDA methods discussed in this paper form but a small part of the ever-expanding interface between topological data analysis and machine learning. However, it is important to state that this survey does not provide exhaustive background on TDA background and literature. For that, we refer our reader to the following excellent studies: (Pun et al. 2022; Edelsbrunner and Harer 2008; Ali et al. 2023; Carlsson 2009; Carlsson and Zomorodian 2009). Whilst a number of recent studies have focused on TDL targeting specific families of architectures (e.g. Message Passing Neural Networks in Papillon et al. (2023b)), our work provides broad coverage of TDA integration into various DL pipelines and architectures. We did our best to choose work that has a historical and linear connection with deep learning approaches to improve understandability.

This survey provides the broader machine learning community with a convenient starting point to explore how TDA has been integrated with deep learning. Such interaction brings novel perspectives, benefits, and challenges. We shed light on the benefits of this interaction demonstrated by the growing adoption of TDA in various deep learning applications. To the best of our knowledge, this is the first work that comprehensively covers topological deep learning and organizes the research works in this field in a unified taxonomy (Sect. 3).

We start in Sect. 2 by introducing the key theoretical concepts of TDA and their representations for learning. In Sect. 3, we explain how topological approaches can fit into different deep learning constructs, such as learnable features, feature transformations, and loss functions. In Sect. 3.4 we shed light on a promising use of TDA to understand and dissect trained deep models, called *deep topological analytics*.

We continue in Sect. 4.2 with a discussion of the known challenges of TDA and its adaptation to deep neural networks. We further discuss future directions and adjacent applications of topological deep learning, and we present some current libraries. Finally, we make some concluding remarks in Sect. 5.

Notations We write $X \in \mathbb{R}^{n\times d}$ to denote the data set, where n is the number of samples and d the number of features or dimensions. We write $\mathcal{M}$ to denote the underlying data

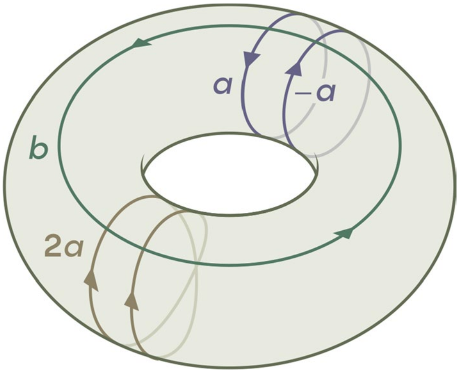

Fig. 1 Illustration to explain how infinitely many different loops on a donut surface are actually homeomorphisms of just two loops (a, b) that are themselves not homeomorphic to each other

manifold, which, for the purposes of this survey, is a locally Euclidean space embedded in $\mathbb{R}^d$. We write BD and PD as abbreviations of *barcode diagram* and *persistence diagram*.

2 Overview of TDA

An object's *topology* is broadly defined as the characteristics that remain invariant under continuous deformation, as if the object was made of soft rubber. How many connected components the object possesses, the holes or voids it contains, or how the object loops back on itself are a few examples of topological properties. In a sense, topological information can be considered qualitative. For example, if we demonstrate that data points lie on two totally disconnected sub-manifolds, then we know that the data comes from two very distinct sources, or that the underlying system has two distinct states.

A central concept is that of *homology*, which is a powerful tool to characterize the topological features of a space. Homology is an abstract concept; its general definition is outside the scope of this paper (Carlsson 2009). In essence, however, the k-th homology (where $0 \le k \le d$) is a group (in the mathematical sense) that characterizes the set of k-dimensional loops in a topological space. The relationships between the various k-dimensional loops then characterize the holes or voids in the manifold.

When we say there is a 0-dimensional hole, it means that the space has disjoint connected components or isolated points. In other words, there are no paths connecting certain points in the manifold. The 0-th homology group identifies and counts these connected components, treating them as separate entities.

A 1-dimensional hole can be traced around with a 1-dimensional loop (like a loop of string). Consider, for example, a typical donut shape that has a single 1-dimensional hole, as illustrated in Fig. 1. One can draw a loop in two ways on its surface: a loop that follows the central hole or one that traverses through the hole. An infinite number of nontrivial loops can be generated that may wind, double back, or wrap around multiple times before returning to their origin, but they will all be equivalent to one of these two loops.

For example, let's refer to any loop that travels through the central hole and around the tube as a. Due to the fact that such a loop can go around the tube once, twice, or an arbitrary number of times, or in the opposite direction, we can represent these loops as a, $2a$, $-a$, etc. In Fig. 1, there is another loop present that is not multiple of a. It goes around the central hole along the long circumference of the tube, and we will refer to it as b. Any loop in a donut figure can be deformed (without breaking) to follow either of the loops *a* or *b* an integer number of times. The fact that there are exactly two one-dimensional loops from which all others can be constructed indicates that the number of one-dimensional holes of the donut is one. Hence, the 1-homology counts the number of these holes. A 2-dimensional hole is a void, for example, the void within a hollow sphere, torus, or Swiss cheese in 3 dimensions.

The k-dimensional holes are counted specifically by the *Betti numbers*. The k-th Betti number is defined as the group rank of the k-th homology. In group theory, *rank* refers to the concept of independence, it is closely related to the concept of rank from linear algebra, and it represents dimensionality (Hatcher 2002). In general, the Betti numbers can be quite difficult to compute, but fortunately, there are some settings where the calculations are straightforward.

2.1 Simplicial complexes and persistent homology

The k-th homology is much more convenient to work with when we restrict ourselves to *simplicial complexes*, which are structures built upon discrete sets. This is the natural domain for data-driven and machine-learning applications.

A *simplex* can be considered a generalization of a triangle or tetrahedron. It is the simplest polytope of any given dimension. A simplex in zero dimensions is a point, in one dimension is a line segment, in two dimensions is a triangle, in three is a tetrahedron, and so on. We use *k-simplex* to refer to a simplex of dimension k. Note that any simplex is composed of *faces*, which are themselves simplices of lower dimension. A *simplicial complex* K is a collection of simplices with two properties: each face of a simplex in K must also be in K, and the intersection of any two simplices of K is either empty or a face of both of them (Munkres 1993).

Consider each point in our data set X to be a vertex (a 0-simplex). We can define a set of 1-simplices as connections between pairs of vertices, 2-simplices between collections of three vertices, and so on. Thus, we build a simplicial complex K that gives some sense of "connectivity" between data points. It can be thought of as a *hyper-graph* on X. Note that K is not necessarily unique on X.

Homological information is much easier obtained for a simplicial complex, and in particular, the k-th Betti number can be obtained through tractable linear algebra (Robins 1999). The Betti numbers in this setting are closely related to *Euler characteristic*, which gives the relationship between the numbers of vertices, edges, and faces in a polyhedron.

The goal now is to construct simplicial complexes on X that reflect the underlying topology of $\mathcal{M}$. This is done by varying scale, typically a radius $r > 0$. The Čech complex and the Vietoris-Rips complex are two typical constructions (Chazal and Michel 2021). A Čech complex $C_r(X)$ includes a k-simplex on $(k + 1)$ vertices of X if the collection of balls of radius r centered on each vertex has a non-empty intersection. The Vietoris-Rips (or simply Rips) complex $V_r(X)$ includes a k-simplex on any set of $(k + 1)$ vertices that all have a pairwise distance less than r of each other (Zomorodian 2010). These two constructions of simplicial complexes can yield very different results on the same data set with the same r.

Persistent homology is obtained through a *filtration* F. Typically, an initial simplicial complex captures the fundamental structure of the space. It serves as the analysis' starting point. The complex is then subjected to a series of additions of simplices, gradually introducing higher-dimensional characteristics and capturing global details of the space. These additions to F are governed by a filtration parameter that determines the analysis'

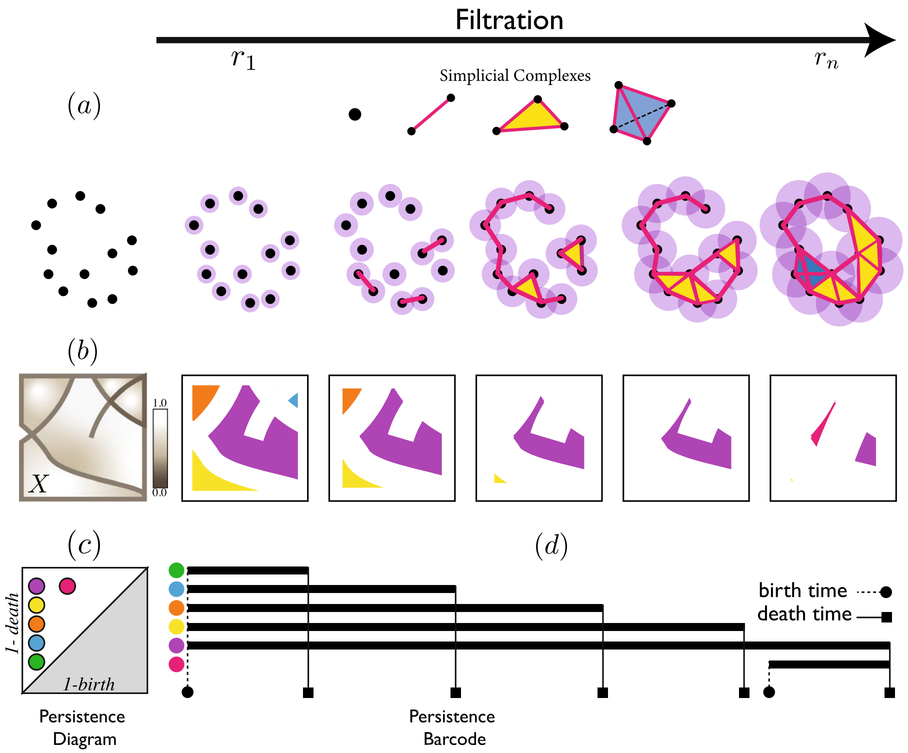

Fig. 2 Example Persistent Homology for **a point cloud and b grey scale image** objects. Intuitively, the filtration captures the multiscale structural properties of the object by recording the persistence state of the topological features (e.g. holes) as the filtration threshold changes. Considering a monotonically increasing threshold $r_1 \leq r_2 \leq \cdots \leq r_n$. **b** the sequence of **sublevel sets** for different values after applying the filteration $f^r = \{x \in X | f(x) \geq r\}$ on the image X. The corresponding **c, left persistence diagram** and **c, right barcode** are convenient summaries of topological features' lifetime. Some holes (**c** and **d**, purple) persist much longer than others, while some (**c** and **d**, pink) are born later. (Colour figure online)

scale or resolution. As the value of the filtration parameter increases, simplices with higher assigned values are added to the complex, resulting in the emergence of new topological structures or the maintenance of existing ones.

In other words, F is a growing sequence of sub-complexes: $K_1 \subseteq K_2 \subseteq \ldots \subseteq K_n = K$. Two commonly used examples of filtration are the sets of simplicial complexes, $C_r(X)$ or $V_r(X)$, that are obtained with increasing radius r of the balls around the data points. As we vary r, these constructs will naturally reflect different aspects of the topology of $\mathcal{M}$. There is monotone inclusion of these simplicial complexes with increasing r, i.e. for two radii $r \leq r'$ we have that $C_r(X) \subseteq C_{r'}(X)$ and $V_r(X) \subseteq V_{r'}(X)$.

Throughout the filtration, the evolving complexes form a nested sequence that reflects the evolution of the topological characteristics of the space across scales. The key idea is to track the appearance and disappearance of topological features over the filtration. We may see new loops created, separate components connected, or holes filled in as we increase r. We record the lifetime of these features with respect to r, that is, the *appearance* (at b_i for birth) and *disappearance* (at d_i for death) of a particular topological feature. Figure 2 shows an example of filtration and the corresponding lifetime of topological features.

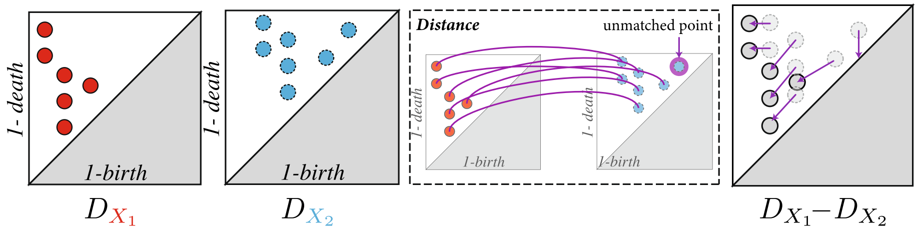

Fig. 3 Visual depiction the distance computation between two persistence diagrams D_{X_1} and D_{X_2}. Note that **comparing PDs** ($D_{X_1} - D_{X_2}$) shows an example of PDs points assignment. Here, the points in D_{X_1} are matched with the nearest adjacent point in D_{X_2} for differentiation, and the unmatched point is ignored (or assigned to diagonal)

2.2 Representations of persistent homology

The set of birth and death coordinates obtained from the filtration forms the backbone of persistent homology. The two most popular representations of this information are *barcode diagrams* and *persistence diagrams* (Carlsson 2009). The multiset of intervals (b_i, d_i) form the barcode diagram (BD), the name coming from the visual representation of the set of intervals as stacked line segments. In the persistence diagram (PD), the lifetime of each feature is represented by a point in $\mathbb{R}^2$ with coordinates (b_i, d_i). A filtration may have several copies of the same birth and death interval, which is represented in the PD by giving the point (b_i, d_i) an integer-valued multiplicity. It is important to note that the BD and PD contain equivalent information, and one can define a bijection between the two. From here onwards we use the term PD to refer to either construct unless BD is explicitly referred to.

The PD of a data set contains a wealth of topological information. Features that have a long persistence interval $(d_i - b_i)$ are considered to be likely to reflect the true topological features of the underlying manifold $\mathcal{M}$. These features are represented in the PD by points far away from the diagonal. A short persistence interval describes a feature that is possibly generated from noise or is otherwise insignificant. Features with short persistence will be represented by points close to the diagonal line in the PD. Hence, points in the PD that are further away from the diagonal are considered more informative than those that are closer to it.

Comparing the PDs of two objects is a way to assess their topological similarity. However, this is a challenging task due to the multiset information contained in the PDs. Figure 3 shows the basic underlying issue of differentiating PDs. In the next section, we discuss various methods to represent them in manners suitable for traditional machine learning and computation, with Sect. 3.3 exploring this issue from the perspective of deep learning loss.

2.3 Homological feature vectorizations

Most machine learning methods assume that the input data resides in $\mathbb{R}^d$ or, more generally, some Hilbert space $\mathcal{H}$. Hence, they cannot be directly applied to datasets comprised of PDs, and the multiset information contained in the PD needs to be represented in some vector format. This process is called *vectorization*, which requires the definition of a continuous map $f : \text{PD} \rightarrow \mathcal{H}$. There is a plethora of different published methods to achieve this, each having subtle consequences (Ali et al. 2023). It is important to note that these vectorization methods can be thought of as handcrafted feature engineering rather than feature learning. In this section, we discuss various strategies that have evolved over time.

A simple approach for representing PDs is using their *statistical properties* such as the sum, mean, variance, maximum, minimum, etc. (Adcock et al. 2016). The total Betti number of a certain filtration can also be used as a summary representation (Cang et al. 2015). These approaches yield a univariate output and lose information; however, they can still be useful.

Another approach is to vectorize BDs using *histogram-like methods* (Cang and Wei 2017; Cang et al. 2018). The basic concept is to discretize the BD along the filtration axis, creating equal-sized bins in which we count the number of persistent intervals. Alternatively, *tropical coordinates* defined on the space of BDs are a useful and stable algebraic representation (Kališnik 2018).

Yet a different approach is to construct various forms of *persistence functions* from PDs. These functions are readily vectorized themselves, however, it is also convenient to work with them directly for many tasks (Bubenik and Dłotko 2017; Adams et al. 2017). Example of these persistent functions includes persistence landscape (Bubeni 2020; Bubenik and Dłotko 2017), persistence Betti number (Edelsbrunner et al. 2002), persistence Betti function (Xia et al. 2017), persistence surfaces and persistence images (Adams et al. 2017), etc.

A useful feature representation technique called *persistence codebooks* (Zieliński et al. 2020) uses bag-of-words quantization techniques to group data points into a fixed-sized vector. Chevyrev et al. (2020) proposed persistence paths, which is a feature map for barcodes.

Representation can vary from simple to complex structures. To get better structural representations there is scope to investigate new methods of vectorization which can benefit topological learning models. Note, however, that when a large feature vector is used to represent PDs, the curse of dimensionality comes into play. In this case, variable selection, regularization approaches, or dropout methods should be considered (Pun 2021; Chiu et al. 2017; Cai and Liu 2011; Srivastava et al. 2014).

In addition, it is important to consider the comparison of different PDs. To this end, various metrics have been proposed, such as bottleneck distance (Mileyko et al. 2011), as well as adaptations of the Gromov-Hausdorff and Wasserstein metric (Bubenik et al. 2018) amongst others. A central consideration is the *stability* of vectorizations and metrics, which we discuss in Sect. 3.3.

Whilst vectorization methods can be used in the input space, combining PD information with machine learning models can also be achieved with kernel-based models (Kwitt et al. 2015; Reininghau et al. 2015). Since metrics can be modified into kernels, various approaches have been proposed to induce kernel function from PD information (Bubenik et al. 2018; Mileyko et al. 2011) and into traditional machine learning approaches like PCA and SVM. Topological-based kernel methods have been used successfully in various ways (Zhu et al. 2016; Kwitt et al. 2015). However, techniques based on kernel methods suffer from scalability issues (Kusano et al. 2016), as training typically scales poorly with the sample number (e.g., roughly cubic in the case of kernel-SVMs). For this reason, we do not discuss topological kernel methods any further in this paper.

Many of the aforementioned methods have advantageous stability properties with respect to standard metrics in TDA, like the Wasserstein or Bottleneck distances. However, they all have the same drawback: the mapping of topological representation that is compatible with existing learning techniques is predefined. Therefore, it is fixed and agnostic to any specific learning task, which makes it suboptimal. The phenomenal success of deep neural networks (e.g. He et al. (2016), Krizhevsky et al. (2012)) has shown that learning representations (i.e. feature learning) is a preferable approach.

3 Topological deep learning (TDL)

Topological representations that incorporate structural information hold great promise for topological deep learning models (Hofer et al. 2017). Combining these cues with deep learning approaches has inherent benefits in various applications. On the flip side, deep learning approaches can be useful in overcoming some common hurdles faced by TDA approaches in estimating robust topological features. The incorporation of topological concepts into deep learning has only recently been investigated and the following benefits have been observed:

- **Global features** from input data can be efficiently and robustly extracted that would otherwise be inaccessible via traditional feature maps.
- TDA is **versatile and adaptable**, meaning that we are not limited to specific problems and types of data (such as images, sensor measurements, time series, graphs, etc.).
- TDA is **noise-resistant** across a number of problems, which include the classification of 3D surface meshes (Som et al. 2018; Reininghau et al. 2015; Li et al. 2014), the recognition of 2D object shapes (Turner et al. 2014), the manifold of natural image patches (Carlsson et al. 2007), the analysis of activity patterns in the visual cortex (Singh et al. 2008), and clustering (Chazal et al. 2013).
- TDA can be applied to **arbitrary data structures** without any prepossessing provided the right filtrations are used.
- A new trend is emerging that allows **efficient backpropagation** through persistent homology components. This has been a long-standing challenge in TDA (further discussed in Sect. 3.3), but topological layers are now becoming compatible with deep learning and end-to-end training schemes.

We reiterate that though the benefits of using TDA (more specifically, persistent homology) and deep learning together have demonstrated success, there are still some theoretical and computational challenges in the application of TDA to data. We discuss these issues at length in Sect. 4.2.

In the rest of this section, we investigate TDA for deep learning from lenses of different magnifications and perspectives, as shown in Fig. 4. In particular, we explore the use of persistent homology in various different ways. The discussion in Sects. 3.1–3.3 is focused on the *on-training integration* of TDA. That is, building topological neural architectures. However, a holistic view should also consider TDA's contribution to *post-training* (deep topological analytics). These analytics use TDA to study the 'shape' of a trained model. Thus, we review works that studied deep model complexity and interpretability using TDA in Sect. 3.4.

3.1 Learning topological features embedding

In this section, we extend the discussion of fixed vectorization methods (Sect. 2.3) by introducing deep learnable vectorization (i.e. embedding). A key advantage here is the possibility of leveraging the deep model to simultaneously learn the vectorization of data and the

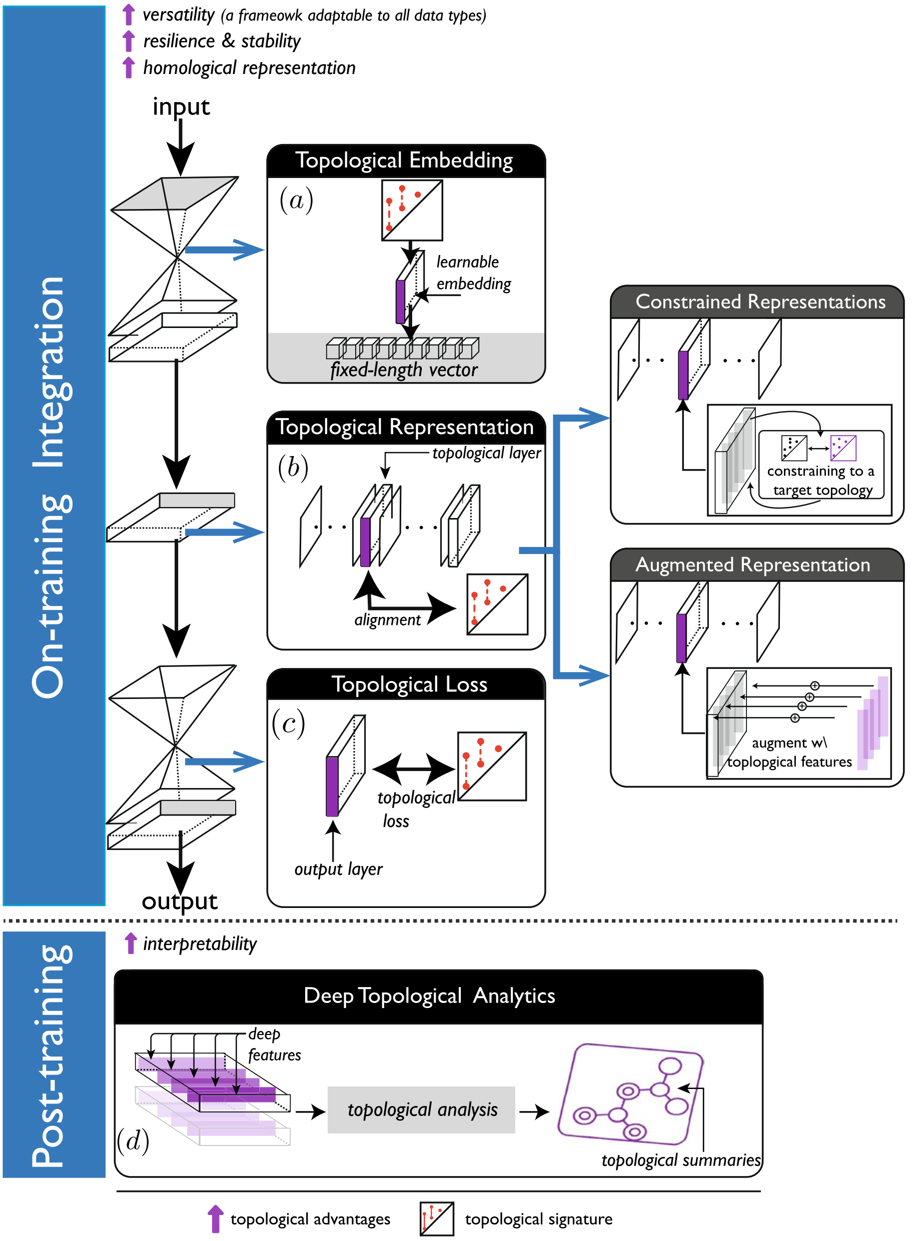

Fig. 4 **Topological Deep Learning** introduces TDA methods to deep models leading to topological neural architectures that can potentially address deep learning limitations. This is done by plugging topological components for **a** learning features **Embedding** (Sect. 3.1), **b** enhancing the learned **Representations** (Sect. 3.2), and/or **c** regularizing the model using a topological **Loss** (Sect. 3.3). Beyond that, **d** TDA can be used *post-training* to reveal insights of trained models (interpretability) (Sect. 3.4)

representation of the target task. For example, we may parameterize the vectorization of persistence diagrams PD to embedding vector $V \in \mathbb{R}^d$ by neural layers f_w where w denotes the trainable parameters. Guided by the task loss, we can efficiently learn mapping $f_w : \mathrm{PD}_{\mathrm{x}} \rightarrow V_x$ and automatically answer the question of "which family of vectorizations should best work for the given task".

Handling PDs by neural networks is the focus of many deep topological embedding studies. Generally, PDs deep vectorization layers should be continuous and permutation invariant with respect to the input. The latter requirement is motivated by the *set* nature of the persistence diagram. Hofer et al. (2017, 2019) introduced the first learnable deep vectorization of PDs. It adopts a permutation invariant transformation by evaluating the PD's points against Gaussian(s) whose mean and variance are learned during the training. Since permutation invariance was explored in other deep learning problems (e.g. Deep Set (Zaheer et al. 2017) for points cloud), some vectorization techniques for PD were borrowed from them. For example, PersLay (Carrière et al. 2020) builds on DeepSets for embedding extended PDs encoding graphs and uses it for graph classification. Recently, transformers were used for PDs embedding. Persformer (Reinauer et al. 2021) architecture showed superiority in synthetic and graph tasks while having some interpretability features. Note that transformers without positional encoding can be made as expressive as Deep Sets. Thus, the permutation invariance requirement can be maintained.

Zhou et al. (2022) proposed TopologyNet, a novel approach, to directly fit the output of topological representations derived from input point cloud data. This innovative method substantially reduces computation time for generating topological representations, in contrast to traditional pipelines, while maintaining a minimal approximating error in practical scenarios. The resultant output of TopologyNet holds potential for various downstream tasks that require efficient topological representations. Experimental evaluations involved incorporating TopologyNet as a topological branch within an autoencoder framework. The results demonstrated that the inclusion of the topological branch led to superior topology quality in the generated point clouds compared to an autoencoder lacking such a branch. Furthermore, the latent vectors generated by a topological autoencoder were employed to train a latent generative adversarial network (GAN), enabling the generation of new point clouds from Gaussian noise. Evaluation indices indicated that the inclusion of the topological autoencoder within the generative adversarial network resulted in improved quality of the newly generated point clouds, surpassing the performance of a GAN lacking the topological autoencoder.

Beyond PDs, deep embedding was explored for other topological signatures. For example, PLLay (Kim et al. 2020) provides a layer for embedding persistence landscapes. PLLay claim to robustness to extreme topological distortion is backed by a tight stability bound that's independent of the input complexity.

Topological embedding transforms the topological input with a complex structure into a vector representation compatible with deep models. As discussed in this section, the process uses a custom topological input layer for embedding. In the next section, we explore topological components that enhance deep learning representation and usually have the flexibility to be plugged anywhere in the network.

Algorithm 1 Deep learnable topological embedding

```
procedure DeepTopologicalEmbedding(PD)
    $V \leftarrow$ Initialize embedding vector in $\mathbb{R}^d$
    $f_w \leftarrow$ Initialize neural layers with parameters $w$
    for each $x$ in PD do
        $V_x \leftarrow f_w(\mathrm{PD}_x)$                ▷ Map PD to embedding vector
    end for
    return $V$                                          ▷ Return the embedding vector
end procedure
```

Algorithm 1 represents the process of embedding persistence diagrams (PDs) into a vector space using deep neural network layers. The procedure DeepTopologicalEmbedding takes a persistence diagram as input, initializes an embedding vector and neural layers, and then maps each point in the PD to the embedding vector. The process is guided by a loss function to determine the best vectorization for the given task.

3.2 Integration of topological representations

Representation learning is the process of learning features from data that can be used to improve the accuracy of the model. Deep learning excels in this regard thanks to its powerful feature learning, but having a good representation goes further than achieving good performance on a target task (Bengio et al. 2013). For example, TDA's stability can make deep representation resilient to input perturbation (de Surrel et al. 2022). Below, we review two categories of deep topological representations.

Constrained representations One approach is to train deep neural networks to learn representations that preserve the persistent homology of the input data. Again, TDA's versatility ensures the feasibility of this as the topological signature can be computed for both the input and the internal representation. For example, Topological Autoencoders (Moor et al. 2020) perform the alignment through a loss, minimizing the divergence between input and latent representation topologies (both captured by PDs).

Augmented representations Another approach for topological representation is augmenting the deep features with topological signatures. Persistence Enhanced Graph Network (PEGN) (Zhao et al. 2020) developed graph spatial convolution that builds on persistence homology. Normally, convolution filters can adapt to local graph structures through the use of node degree information. In contrast, PEGN weights the message passing between nodes through neighborhood information captured by persistence images. Moreover, Graph Filtration Learning (GFL) (Hofer et al. 2020) adapts the readout operation (a graph pooling-like operation) in Graph Neural Network (GNN) to be topologically aware. BDs are computed for the graph nodes feature and vectorized. Interestingly, the filtration function is learned end-to-end. Topological Graph Layer (TOGL) (Horn et al. 2022) extends GFL's idea and learns multiple filtrations of a graph (rather than one) in an end-to-end manner.

Unlike the embedding layers (e.g. PersLay Carrière et al. (2020)) that expect a pre-specified input type (e.g. PDs), the topological representation layers discussed in this section enjoy more flexibility regarding the input and placement in the network. This comes with the attached cost of requiring careful design choices and guarantees on the layer characteristics (e.g. consistency of gradients in Hofer et al. (2020)).

Algorithm 2 Topological Representation Integration in Deep Neural Networks

```
procedure TopologicalRepresentationIntegration(Input Data)
    Topological Features ← ComputeTopologicalFeatures(Input Data)
    Deep Model ← InitializeDeepModel()
    if Constrained Representation then
        Apply Topological Autoencoder
        Minimize Divergence between Input and Latent Topologies
    else if Augmented Representation then
        Augment Deep Features with Topological Signatures
        Apply Enhanced Representation like PEGN/GFL/TOGL
    end if
    Output ← DeepModel(Topological Features)
    return Output
end procedure
```

The process of integrating topological representations into deep learning models is outlined in Algorithm 2. The exact method used (e.g. Topological Autoencoders, PEGN, GFL, TOGL) depends on the specific approach chosen.

3.3 Topological loss

The most common approach for leveraging topology in deep learning is incorporating a topological penalty in the loss. The popularity of the approach stems from the fact that loss-based integration is straightforward and does not require changing the architecture or adding additional layers. The only caveat is that the loss should be differentiable and easy to compute. As iterated previously, the capability of topological features to capture the complex structure of the data means that deep learning can learn robust representations guided by topological loss. Thus, the representations are likely invariant with respect to typical transformations present in real-world datasets, such as noise and outliers. An example of this is a common persistence loss (Hu et al. 2019), which minimizes the difference between a predicted persistence diagram PD_X and the true diagram PD_Y:

$$\mathcal{L}_{\text{topological}} = d(\mathrm{PD}_X, \mathrm{PD}_Y) \tag{1}$$

This has been used either as a standalone loss or as a regularizer (i.e. augmenting another loss) (Hu et al. 2019) in applications such as semantic segmentation (Hu et al. 2019), or generative modeling (Wang et al. 2020).

As discussed in 3.1, PDs do not lend themselves to vector representations in Euclidean space. Moreover, the PD is not differentiable (a key requirement for using backpropagation). One strategy to resolve this is to leverage a divergence or metric that can handle PDs. The p-Wasserstein[1] distance and the bottleneck distance are popular choices:

$$d_{p,q}(\mathrm{PD}_X, \mathrm{PD}_Y) = \left[\inf_{\pi \in \Pi(\mathrm{PD}_X, \mathrm{PD}_Y)} \sum_{t \in \mathrm{PD}_X} \|t - \pi(t)\|_q^p \right]^{\frac{1}{p}} \tag{2}$$

$$d_\infty(\mathrm{PD}_X, \mathrm{PD}_Y) = \inf_{\pi \in \Pi(\mathrm{PD}_X, \mathrm{PD}_Y)} \sup_{t \in \mathrm{PD}_X} \|t - \pi(t)\|_\infty \tag{3}$$

where t is a point corresponding to a $(b_i, d_i) \in \mathbb{R}^2$ that is in PD_X, and where $\Pi(\mathrm{PD}_X, \mathrm{PD}_Y)$ denotes the set of bijection between PD_X and PD_Y, and $\|.\|_q$ is the ℓ_q Euclidean norm. It can be seen that the bottleneck distance is the largest distance between any pair of corresponding points across all bijections that preserve the partial ordering of the points (i.e. we cannot match a point with a birth time greater than another point's death time). This ensures that the topological features to be matched are comparable.

The initial popularity of the bottleneck distance is perhaps fueled by a *stability theorem* (Cohen-Steiner et al. 2005) for PDs of continuous functions. According to this theorem, the bottleneck distance is controlled by L_∞ distance, that is

$$d_\infty(\mathrm{PD}_{f_1}, \mathrm{PD}_{f_2}) \leq C\|f_1 - f_2\|_\infty \tag{4}$$

form some constant C. In effect, this means that the diagrams are stable with respect to small perturbations of the underlying data. A similar stability result exists for the p-Wasserstein distance. These are the foundation of the stability guarantees by recent deep learning works such as the stability of Heat Kernel Signature in graphs (Carrière et al. 2020) and stability of mini-batch-based diagram distances in Topological Autoencoders (Moor et al. 2020).

Among the limitations of (2) and (3) is the high computational budget needed by these distances when the number of points is large. As the distance requires point-wise matching, the computational complexity is $\mathcal{O}(n^3)$ for n points (Anirudh et al. 2016). Also, in many applications (Wang et al. 2020; Chen et al. 2019), we aim to learn a model f_w that aligns a predicted diagram PD_P with a target (i.e. ground truth) diagram PD_T by gradually moving PD_P points towards PD_T. This is typically achieved by pushing w in the negative direction of $\nabla_w \mathcal{L}_{\text{topological}}$ and, obviously, assumes that the loss is differentiable with respect to the diagram. While the Wasserstein distance satisfies this requirement in general, it can have some instability issues (Solomon et al. 2021). Below, we select a few representative papers using topological losses in various applications and show how they handle these issues.

In generative modeling, TopoGAN (Wang et al. 2020) uses a slightly modified 1-Wassertsein distance to align the diagrams of generated and real images in medical image applications. The loss ignores the death time and focuses only on the birth time of the diagram features. Framed in this way, the loss becomes similar to the Sliced Wasserstein (Peyré et al. 2019), which can be computed efficiently and is still differentiable. A similar loss was used by Hu et al. (2019) for segmentation to encourage the deep model to

[1] The "Wasserstein" distance in TDA literature is slightly different from the common Wasserstein (i.e. Kantrovich optimal mass transport (Peyré et al. 2019)) metric. The first seeks a deterministic bijection that best aligns the diagrams (hard assignment) and the mass can be freely added to or removed from the diagonal. The latter is based on probabilistic coupling (soft assignment). This also has implications for the kind of algorithms that can be used to estimate the distance.

produce output with a topology that was close to the ground truth. The cross-entropy loss is augmented with the 2-Wasserstein loss between persistence diagrams. To alleviate the computational burden, the method performs the calculation on a single small image patch (part of the image) at a time. In (Clough et al. 2022), the authors rely on Betti numbers for semi-supervised image segmentation. A notable advantage here is the output of a network trained on a small set of labeled images can still capture the actual Betti numbers correctly. This gives us the opportunity to initially train the model on a small labeled dataset guided by the Betti numbers loss $\mathcal{L}_\beta$. The model is then fine-tuned using a large unlabeled dataset and guided by a loss (that incorporates $\mathcal{L}_\beta$). Since the estimation of Betti numbers is robust for unlabeled data, $\mathcal{L}_\beta$ will regularize the second stage of training (fine-tuning). In classification, (Chen et al. 2019) uses a topological regularizer. To speed up the computation, it focuses on the zero homological dimension, where the persistence computations are particularly faster.

Algorithm 3 Topological Loss for Deep Learning

```
procedure TopologicalLoss($\mathrm{PD}_X, \mathrm{PD}_Y, p, q$)
    $\Pi \leftarrow$ Set of all bijections between $\mathrm{PD}_X$ and $\mathrm{PD}_Y$
    Calculate $d_{p,q}$, the $p$-norm distance:
        $d_{p,q} \leftarrow \left[\inf_{\pi\in\Pi} \sum_{t\in\mathrm{PD}_X} \|t - \pi(t)\|_q^p\right]^{\frac{1}{p}}$
    OR calculate $d_\infty$, the infinity norm distance:
        $d_\infty \leftarrow \inf_{\pi\in\Pi} \sup_{t\in\mathrm{PD}_X} \|t - \pi(t)\|_\infty$
    $\mathcal{L}_{\text{topological}} \leftarrow d_{p,q}$ or $d_\infty$        ▷ Choose based on application
    return $\mathcal{L}_{\text{topological}}$        ▷ Return the topological loss
end procedure
```

Algorithm 3 outlines the computation of topological loss using either the p-Wasserstein distance or the bottleneck distance. The procedure TopologicalLoss takes two persistence diagrams PD_X and PD_Y, and the parameters p and q, then computes the p-Wasserstein or bottleneck distance as the topological loss. This loss can be used in deep learning models to minimize the difference between predicted and true topological features.

3.4 Deep topological analytics

The complementary value of TDA goes beyond *on-training integration* and constructing topological neural architectures. In fact, leveraging TDA methods *post-training* can be even more insightful and powerful. Currently, researchers use TDA to address deep learning transparency (Liu et al. 2020), studying model complexity (Rieck et al. 2019; Carlsson and Gabrielsson 2020) and even tracking down answers for seemingly mysterious aspects of deep learning, e.g. why deep networks outperform shallow ones (Naitzat et al. 2020). These efforts are centered around analyzing deep models using TDA approaches. Hence, we call it *deep topological analytics*. We explore two aspects of it below.

Quantifying structural complexity Watanabe and Yamana (2021) treats the neural networks as a weighted graph $G(V, E)$ where V and E denote the network neurons and the relevance scores (computed from weights); respectively. By computing persistence features (e.g. Betti numbers) across filtration, we can gain insight into the network complexity.

For example, the increase in the Betti number (the occurrence of a cycle between a set of neurons) can reflect the complexity of knowledge in deep neural networks. In Rieck et al. (2019), the authors follow the same line and further develop training optimization strategies (e.g. early stopping) informed by homological features.

Visual exploration of models Another use of TDA here is to provide a post-hoc explanation and/or visual exploration of the internal functioning of deep models. For example, topological information provides insight into the overall structure of high-dimensional functions. The authors in Liu et al. (2020) use this to offer a scalable visual exploration tool for data-driven black box models. This is an important research problem, where doing so in an intuitive way is a challenge. They also use *topological splines* to visualize the high-dimensional error landscape of the models. Similarly, TopoAct (Rathore et al. 2021) offers insightful information on neural network learned representations and provides a visual exploration tool to study topological summaries of activation vectors. Works such as Polianskii (2018) shed light on how neural networks maintain the topological properties of the data when they are projected into low-dimensional space.

DNN focused topology optimization The concept of “Inverting Representation of Image” and “Physically Informed Neural Network” served as inspiration for the creation of the topology optimization via neural reparameterization framework (TONR) (Zhang et al. 2021), which aims to address a variety of topology optimization issues. In this approach, the density field is optimized through the updating of DNN parameters and carefully choosing the initial parameters. This leads to quicker training and suggests a good measure for topology optimization.

4 Discussion

TDA is a steadily developing and promising area, with successes in a wide variety of applications. However, there are open questions in applying TDA with deep neural networks. In this section, we discuss various successes and applications of deep TDA, we highlight several open challenges for future research on deep TDA in both practical and theoretical aspects, and paint a speculative picture by outlining what persistent homology holds for the future. We also note some open-source implementations available for researchers to get started.

4.1 Successes and applications

Deep TDA has demonstrated potential in a variety of challenging settings. The invariance of PH information to continuous deformation means TDA applies well to settings where objects should have consistent shapes but may be transformed in some way. TDA also performs well to bridge the gap between structural information and prior knowledge. If we have prior knowledge of the topology of a class of objects, then PDs are an effective tool for the classification and comparison of data against this class, even in the presence of noise or limited data. This robustness is well adapted to deep learning.

A potential area of application for topological data analysis (TDA) combined with deep learning lies in **multi-class segmentation tasks**. In such tasks, it becomes feasible to delineate the topology of individual classes as well as the boundaries between each class. This extension can be viewed as an implementation of persistent homology (PH) to address the issue examined in a study by Clough et al. (2022) and Haft-Javaherian et al. (2020), where prior information was utilized to define the adjacencies amongst different brain regions.

TDA can produce good results in **small datasets** (Byrne et al. 2021; BenTaieb and Hamarne 2016), and is especially useful for medical imaging applications where cost and privacy concerns often limit data acquisition. Byrne et al. (2021), BenTaieb and Hamarne (2016) have investigated the limitations of conventional deep learning training procedures when applied to small datasets. It reveals that these procedures heavily rely on pixel-wise loss functions, which restrict the optimization process in terms of extended or global features. They used persistent homology and constructed topological loss functions to evaluate image segments against a known prior, resulting in a richer description of segmentation topology with better accuracy.

As persistence homology describes the global structure, developing **topological loss functions** could suppress small false positives or false negatives related to the topology of an object. For example, in the segmentation task, techniques such as morphological operations or CRF-based techniques are used to remove local errors; they do not have the concept of global topology. The benefit of PH-based loss is that the correct global topology can be propagated with local label smoothness. TDA has been used in settings with limited or noisy data, such as power forecasting (Senekane et al. 2021), segmenting aerial photography (Mosinska et al. 2018) and astronomy (Murugan and Robertson 2019).

In some applications, **topological information** may be more significant (e.g. finding anomalies or changes in topology) than statistical (e.g. pixel-wise) information. For example, in Vukicevic et al. (2017), Byrne et al. (2016) detecting holes between heart chambers was more important than inferring the thickness of septal walls. For these types of applications, a loss function combining topological and statistical information can be adjusted in favor of topology, when training a network.

Given its ability to preserve the global structure, TDA emerges as a promising approach for capturing intricate structural details and can be effectively **integrated into generative models** to produce new data that aligns with the topology of the training set. In a recent study conducted by Zhou et al. (2022), a topological network was trained and incorporated as a branch within a generative adversarial network (GAN) framework. This integration aimed to enhance the performance of generating new point clouds. By leveraging the strengths of TDA and GANs, the researchers demonstrated significant improvements in the generation process, yielding more accurate and topologically consistent synthetic data.

Performance and comparative analysis of TDL typically focuses on evaluating the effectiveness against traditional machine learning and deep learning models. Common metrics used in these studies include accuracy, precision, recall, and computational efficiency (Hofer et al. 2017; Moor et al. 2020; Huynh et al. 2021; Clough et al. 2022; Haft-Javaherian et al. 2020). Deep TDA often demonstrates superior performance in handling complex data structures and noisy datasets, showing resilience in maintaining accuracy under such conditions (Clough et al. 2022). Moreover, Deep TDA models are frequently found to be more interpretable, a key advantage in critical applications where understanding model decisions is crucial (Singh et al. 2023; Fan et al. 2023).

The integration of Topological Data Analysis (TDA) with deep learning methodologies has recently exhibited remarkable potential and practical application across various disciplines. Following are some of the pivotal fields that highlight the significance of this synergistic approach:

Biomedical imaging In biomedical imaging, this combination has been used for more accurate analysis of complex medical images. Researchers utilize these techniques for enhanced feature extraction and classification in areas such as tumor detection and organ segmentation (Hajij et al. 2021; Singh et al. 2023; Fan et al. 2023; Glatt and Liu 2023).

Genomics In genomics, it aids in the analysis of high-dimensional genetic data (Amézquita et al. 2023). It's particularly useful for understanding genetic diseases by identifying patterns and connections in genomic data that traditional methods might miss (Shapanis et al. 2023; Narayana et al. 2023; Amézquita et al. 2023; Yu et al. 2023; Wamil et al. 2023; Chulián et al. 2023; Morilla et al. 2022).

Protein engineering Topological Deep Learning is revolutionizing the way scientists approach the vast mutational space of proteins. It is particularly transformative when combined with existing protein structure prediction tools like AlphaFold2, enabling more precise and powerful structure-based strategies in protein engineering. Topological Deep Learning in this field is not just enhancing the speed and accuracy of protein design and analysis, but also opening new pathways for advancements in drug discovery, antibody development, and beyond (Qiu and Wei 2023a; Chen et al. 2022; Qiu and Wei 2023b).

Smart manufacturing TDA with deep learning enables enhanced detection of patterns and anomalies in manufacturing processes. This integration not only improves predictive maintenance but also optimizes production efficiency and quality control, paving the way for more intelligent and responsive manufacturing systems (Ko and Koo 2023; Sarpietro et al. 2022; Uray et al. 2023).

Finance and economics The financial sector employs these techniques for market analysis and risk assessment (Goel et al. 2020). By analyzing complex market data, this integrated approach helps in predicting stock market trends and in algorithmic trading (Chang and Lin 2023; Hafez et al. 2022).

Cybersecurity In cybersecurity, combining TDA with deep learning enhances the detection of anomalies and threats in network data, aiding in the identification and prevention of cyber-attacks (Zhen et al. 2022; Guo et al. 2022).

Topological analysis, as a general methodology, serves as a means of formalizing qualitative aspects inherent in reality. Integrating topological analysis with deep learning techniques proves to be highly advantageous for a wide range of tasks and applications, as highlighted previously. Additionally, the representation of data using TDA provides enhanced interpretability to human observers compared to the utilization of conventional black-box deep neural networks. This attribute allows for a deeper understanding of the underlying patterns and structures present in the data, thus enabling more meaningful insights to be derived from the analysis.

4.2 Challenges

Despite the success of TDA and its use in deep learning, we describe a few notable challenges that, if properly addressed, could benefit the field greatly.

Computational cost Many aspects of calculating persistent homology are computationally intractable. The construction of the Čech complex for a given r is known to be an NP-hard task. Computing Betti numbers for a given simplicial complex is also infeasible for very large-scale complexes. The costs of calculating TDA information add to already computationally expensive deep learning routines.

Lack of universal framework for vectorization: There is no universally accepted framework for incorporating topological information into deep learning, with earlier representations created in an ad-hoc manner or learned independently (Hofer et al. 2017; Moor et al. 2020). This is both a theoretical and a computational matter, with the lack of strong theory encoding persistence diagrams as vectors as an example of the issues encountered. There have been a variety of ad-hoc solutions of varying merit, recently catalogued in (Ali et al. 2023). Alternatively, vectorization methods have been chosen as part of learning strategies (Hofer et al. 2017; Moor et al. 2020).

Statistical guarantees Through this article, we have not discussed the statistical aspects of persistence due to finite sampling. For example, there is no guarantee that the PD derived from X reflects the true homology of $\mathcal{M}$. The framework for understanding the statistical robustness of persistence information is evolving. Some simple strategies for verification, such as sub-sampling and cross-validation, have been used in the literature (Chazal and Michel 2021). There is scope to further understand issues such as the minimum number of data points required to guarantee robust PDs. Furthermore, persistence is not well understood from a probabilistic point of view (e.g. the distribution of persistence from a distribution of shapes).

High-dimensional learning challenge There is no underlying theoretical framework for what topological features to expect with high-dimensional data. While abstract topological spaces can be enormously complex in high dimensions, we do not know whether to expect data to behave similarly. Moreover, high dimensional homological features are unattainable due to computational cost, and in any case, the sensitivity of PDs to sampling or noise is not well understood in high dimensions. This makes learning the underlying topology of the data for use in deep neural networks challenging.

The need for a good backpropagation strategy The differentiability of PDs or other homological quantities is not guaranteed or necessarily well understood. This makes backpropagation in deep neural networks that incorporate topological signatures extremely challenging or only feasible under special conditions (Moor et al. 2020).

Capturing multi-variate persistence In some cases, multiple concurrent filtrations are needed to fully capture the topology of the data manifold, especially for data in higher dimensions. This leads to *multi-variate persistence*, where the birth and death of topological features occur in multiple dimensions. This notion of persistence does not have a complete discrete invariant, unlike the one-dimensional BD that we have discussed so far. For the practical use of multi-variate persistence in deep learning, we would need new theoretical frameworks and better computational methods.

4.3 Future directions

It would be interesting to explore sophisticated deep learning architectures that learn mappings between high dimensional data and their corresponding PDs or other topological representations, furthering the work of de Surrel et al. (2022).

As deep learning models continue to grow in complexity and dataset to grow in size, scalability and efficiency become even more crucial. Future directions in TDA for deep learning involve the development of scalable algorithms and efficient computational frameworks capable of handling large-scale datasets. This would enable the application of topological data analysis to diverse domains and real-world problems.

Interpreting deep learning models' decisions remains a challenging endeavor. TDA offers a unique perspective by providing interpretable representations of complex data. Future directions in this area will focus on developing methodologies to extract meaningful topological features and interpret their significance in the context of deep learning tasks. This will facilitate a better understanding of the decision-making process for deep neural networks and increase their trustworthiness.

Regularization plays a crucial role in preventing overfitting and improving the generalization ability of deep learning models. Future research will explore how TDA-based regularization techniques can be integrated into deep learning frameworks. This could involve incorporating topological penalties or constraints to encourage models to capture meaningful topological features, leading to improved model generalization and robustness.

Many real-world applications involve multimodal data, such as images, text, and sensor data. Combining TDA with deep learning techniques provides a promising avenue for analyzing and integrating information from multiple modalities. Future directions include the development of TDA methods that can handle multimodal data and exploit the interactions between different modalities to uncover complex relationships and structures.

Transfer learning has proven to be an effective strategy for leveraging knowledge gained from one task to improve performance on a related task. Integrating TDA into transfer learning frameworks can enable the transfer of topological knowledge between domains or datasets. This could facilitate the adaptation of deep learning models to new domains by preserving the underlying topological structure and transferring relevant information.

Moreover, deep learning may yet yield new kinds of topological representation other than PDs, with robustness to different data deformations. PH could have further applications in multi-class open-set problems (where data may have unknown classes). If the topology among classes is relatively consistent, then the object labels of unknown classes could be better predicted.

4.4 Implementations

There are a number of open-source implementations of TDA available to practitioners. Here, we present three libraries that have interfaces with deep learning architectures.

GUDHI[2] is an open-source library that implements relevant geometric data structures and TDA algorithms, and it can be integrated into the TensorFlow framework. PersLay (Carrière et al. 2020) and RipsLayer are implementations using GUDHI that learn persistence representations from complexes and PDs. They can handle automatic differentiation and are readily integrated in deep learning architectures.

Giotto-deep[3] is an open-source extension of the Giotto-TDA library. It aims to provide seamless integration between TDA and deep learning on top of PyTorch. To use topology for both pre-processing data (using a variety of available methods) and using it within neural networks, the developers aim to provide several off-the-shelf architectures. One such example is that of Persformer (Reinauer et al. 2021).

TopoModelX[4] is a recent Python package that extends Graph Neural Networks (GNNs) for application in topological domains, demonstrating a substantial development in the field of topological deep learning. The implementation of topological neural networks in TopoModelX started as the ICML 2023 Topological Deep Learning Challenge (Papillon et al. 2023a), hosted by the second annual Topology and Geometry (TAG) in Machine Learning Workshop at ICML. Participants contributed by implementing existing topological neural network methods from the literature and applying them to train on a benchmark dataset. TopoModelX offers a robust framework and essential functionalities, enabling researchers to either implement new GNN-based TDL algorithms or apply existing methodologies from scholarly literature to their specific problems.

[2] https://gudhi.inria.fr/.

[3] https://github.com/giotto-ai/giotto-deep.

[4] https://github.com/pyt-team/TopoModelX.

5 Conclusion

The recent growth in TDA and the established efficacy of deep learning have meant that the integration of these techniques has been inevitable. There is no universal paradigm for combining TDA and deep learning. This article surveyed numerous ways in which these frameworks have benefited each other. We began with an overview of the key TDA concepts. Following this, we reviewed TDA in deep learning from a variety of perspectives. We described numerous challenges and opportunities that remain in this field, as well as some observed successes.

Author contributions All authors whose names appear on the submission made substantial contributions to the conception or design of the work; or the acquisition, analysis, or interpretation of data, drafted the work or revised it critically for important intellectual content.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of interest The authors declare no competing interests.

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):218–252

Adcock A, Carlsson E, Carlsson G (2016) The ring of algebraic functions on persistence bar codes. Homol Homotopy Appl 18(1):381–402

Ali D, Asaad A, Jimenez MJ, Nanda V, Paluzo-Hidalgo E, Soriano-Trigueros M (2023) A survey of vectorization methods in topological data analysis. IEEE Trans Pattern Anal Mach Intell 45(12):14069–14080

Amézquita EJ, Nasrin F, Storey KM, Yoshizawa M (2023) Genomics data analysis via spectral shape and topology. PLoS ONE 18(4):e0284820

Anirudh R, Venkataraman V, Natesan Ramamurthy K, Turaga P (2016) A riemannian framework for statistical analysis of topological persistence diagrams. In: Conference on computer vision and pattern recognition workshops. pp 68–76

Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828

BenTaieb A, Hamarneh G (2016) Topology aware fully convolutional networks for histology gland segmentation. In: Medical image computing and computer-assisted intervention. Springer, New York, pp 460–468

Bubenik P (2020) The persistence landscape and some of its properties. In: Topological data analysis. Springer, New York, pp 97–117

Bubenik P, Dłotko P (2017) A persistence landscapes toolbox for topological statistics. J Symb Comput 78:91–114

Bubenik P, de Silva V, Scott J (2018) Interleaving and Gromov-Hausdorff distance. ArXiv preprint. arXiv:1707.06288

Byrne N, Forte MV, Tandon A, Valverde I, Hussain T (2016) A systematic review of image segmentation methodology, used in the additive manufacture of patient-specific 3D printed models of the cardiovascular system. JRSM Cardiovasc Dis 5:204800401664546

Byrne N, Clough JR, Montana G, King AP (2021) A persistent homology-based topological loss function for multi-class CNN segmentation of cardiac MRI. In: Statistical atlases and computational models of the heart. Springer, New York, pp 3–13

Cai T, Liu W (2011) A direct estimation approach to sparse linear discriminant analysis. J Am Stat Assoc 106(496):1566–1577

Cang Z, Wei GW (2017) TopologyNet: topology based deep convolutional and multi-task neural networks for biomolecular property predictions. PLOS Comput Biol 13(7):e1005690

Cang Z, Mu L, Wu K, Opron K, Xia K, Wei GW (2015) A topological approach for protein classification. Comput Math Biophys 3(1)

Cang Z, Mu L, Wei GW (2018) Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. PLoS Comput Biol 14(1):e1005929

Carlsson G (2009) Topology and data. Bull Am Math Soc 46(2):255–308

Carlsson G, Gabrielsson RB (2020) Topological approaches to deep learning. In: Topological data analysis. Springer, New York, pp 119–146

Carlsson G, Zomorodian A (2009) The theory of multidimensional persistence. Discret Comput Geom 42(1):71–93

Carlsson G, Ishkhanov T, de Silva V, Zomorodian A (2007) On the local behavior of spaces of natural images. Int J Comput Vis 76(1):1–12

Carrière M, Chazal F, Ike Y, Lacombe T, Royer M, Umeda Y (2020) Perslay: a neural network layer for persistence diagrams and new graph topological signatures. In: International conference on artificial intelligence and statistics. PMLR, pp 2786–2796

Chang C, Lin H (2023) A topological based feature extraction method for the stock market. Data Sci Financ Econ 3(3):208–229

Chazal F, Michel B (2021) An introduction to topological data analysis: fundamental and practical aspects for data scientists. Front Artif Intell 4:108

Chazal F, Guibas LJ, Oudot SY, Skraba P (2013) Persistence-based clustering in Riemannian manifolds. J ACM 60(6):1–38

Chen C, Ni X, Bai Q, Wang Y (2019) A topological regularizer for classifiers via persistent homology. In: International conference on artificial intelligence and statistics. PMLR, pp 2573–2582

Chen J, Qiu Y, Wang R, Wei GW (2022) Persistent Laplacian projected Omicron BA.4 and BA.5 to become new dominating variants. Comput Biol Med 151:106262

Chevyrev I, Nanda V, Oberhauser H (2020) Persistence paths and signature features in topological data analysis. IEEE Trans Pattern Anal Mach Intell 42(1):192–202

Chiu MC, Pun CS, Wong HY (2017) Big data challenges of high-dimensional continuous-time mean-variance portfolio selection and a remedy. Risk Anal 37(8):1532–1549

Chulián S, Stolz BJ, Martínez-Rubio Á, Blázquez Goñi C, Rodríguez Gutiérrez JF, Caballero Velázquez T et al (2023) The shape of cancer relapse: Topological data analysis predicts recurrence in paediatric acute lymphoblastic leukaemia. PLoS Comput Biol 19(8):e1011329

Clough JR, Byrne N, Oksuz I, Zimmer VA, Schnabel JA, King AP (2022) A topological loss function for deep-learning based image segmentation using persistent homology. IEEE Trans Pattern Anal Mach Intell 44(12):8766–8778

Cohen-Steiner D, Edelsbrunner H, Harer J (2005) Stability of persistence diagrams. In: Symposium on computational geometry, pp 263–271

de Surrel T, Hensel F, Carrière M, Lacombe T, Ike Y, Kurihara H et al (2022) RipsNet: a general architecture for fast and robust estimation of the persistent homology of point clouds. In: Topological, algebraic and geometric learning workshops. PMLR. pp 96–106

Edelsbrunner H, Harer J (2008) Persistent homology—a survey. In: Surveys on discrete and computational geometry. vol 453. Amer Mathematical Society, p 257

Edelsbrunner, Letscher, Zomorodian (2002) Topological persistence and simplification. Discret Comput Geom 28(4):511–533

Fan Q, Sun C, Hu B, Wang Q (2023) Recent advances of lanthanide nanomaterials in Tumor NIR fluorescence detection and treatment. Mater Today Bio 100646

Glatt R, Liu S (2023) Topological data analysis guided segment anything model prompt optimization for zero-shot segmentation in biological imaging. ArXiv preprint. arXiv:2306.17400

Goel A, Pasricha P, Mehra A (2020) Topological data analysis in investment decisions. Expert Syst Appl 147:113222

Guo W, Qiu H, Liu Z, Zhu J, Wang Q (2022) GLD-Net: deep learning to detect DDoS attack via topological and traffic feature fusion. Comput Intell Neurosci

Hafez SM, Nainay ME, Abougabal M, Kosba A (2022) Ethereum price prediction using topological data analysis. In: Global conference on artificial intelligence and Internet of Things, pp 146–153

Haft-Javaherian M, Villiger M, Schaffer CB, Nishimura N, Golland P, Bouma BE (2020) A topological encoding convolutional neural network for segmentation of 3D multiphoton images of brain vasculature using persistent homology. In: Conference on computer vision and pattern recognition workshops, pp 4262–4271

Hajij M, Zamzmi G, Batayneh F (2021) TDA-Net: fusion of persistent homology and deep learning features for COVID-19 detection from chest X-ray images. In: International conference of the IEEE engineering in medicine & biology society. IEEE, pp 4115–4119

Hatcher A (2002) Algebraic topology. Cambridge University Press, Cambridge

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Conference on computer vision and pattern recognition. IEEE, pp 770–778

Hofer C, Kwitt R, Niethammer M, Uhl A (2017) Deep learning with topological signatures. Adv Neural Inf Process Syst 30

Hofer CD, Kwitt R, Niethammer M (2019) Learning representations of persistence barcodes. J Mach Learn Res 20(126):1–45

Hofer C, Graf F, Rieck B, Niethammer M, Kwitt R (2020) Graph filtration learning. In: III HD, Singh A (eds) International conference on machine learning. vol. 119 of proceedings of machine learning research. PMLR, pp 4314–4323

Horn M, Brouwer ED, Moor M, Moreau Y, Rieck B, Borgwardt K (2022) Topological graph neural networks. In: International conference on learning representations. p x

Hu X, Li F, Samaras D, Chen C (2019) Topology-preserving deep image segmentation. Adv Neural Inf Process Syst 32

Huynh V, Phung DQ, Zhao H (2021) Optimal transport for deep generative models: state of the art and research challenges. In: International joint conference on artificial intelligence, pp 4450–4457

Kališnik S (2018) Tropical coordinates on the space of persistence Barcodes. Found Comput Math 19(1):101–129

Kim K, Kim J, Zaheer M, Kim J, Chazal F, Wasserman L (2020) Pllay: efficient topological layer based on persistent landscapes. Adv Neural Inf Process Syst 33:15965–15977

Ko S, Koo D (2023) A novel approach for wafer defect pattern classification based on topological data analysis. Expert Syst Appl 120765

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25

Kusano G, Hiraoka Y, Fukumizu K (2016) Persistence weighted Gaussian kernel for topological data analysis. In: International conference on machine learning. PMLR. pp 2004–2013

Kwitt R, Huber S, Niethammer M, Lin W, Bauer U (2015) Statistical topological data analysis-a kernel perspective. Adv Neural Inf Process Syst 28

Li C, Ovsjanikov M, Chazal F (2014) Persistence-based structural recognition. In: Conference on computer vision and pattern recognition, pp 2003–2010

Liu S, Gaffney J, Peterson L, Robinson PB, Bhatia H, Pascucci V et al (2020) Scalable topological data analysis and visualization for evaluating data-driven models in scientific applications. IEEE Trans Visual Comput Graphics 26(1):291–300

Mileyko Y, Mukherjee S, Harer J (2011) Probability measures on the space of persistence diagrams. Inverse Prob 27(12):124007

Moor M, Horn M, Rieck B, Borgwardt K (2020) Topological autoencoders. In: International conference on machine learning. PMLR, pp 7045–7054

Morilla I, Chan P, Caffin F, Svilar L, Selbonne S, Ladaigue S et al (2022) Deep models of integrated multiscale molecular data decipher the endothelial cell response to ionizing radiation. Icience 25(1):103685

Mosinska A, Marquez-Neila P, Kozinski M, Fua P (2018) Beyond the pixel-wise loss for topology-aware delineation. In: Conference on computer vision and pattern recognition. IEEE, pp 3136–3145

Munkres J (1993) 1. In: Homology groups of a simplicial complex. CRC Press, New York, pp 1–78

Murugan J, Robertson D (2019) An introduction to topological data analysis for physicists: from LGM to FRBs. ArXiv preprint. arXiv:1904.11044

Naitzat G, Zhitnikov A, Lim LH (2020) Topology of deep neural networks. J Mach Learn Res 21(1):7503–7542

Narayana J, Mac Aogáin M, Ivan F, Jaggi T, Keir H, Dicker A et al (2023) Topological data analysis reveals antimicrobial resistotypes associated to the microbiome in bronchiectasis: an international multi-centre study. In: Microbiome research. American Thoracic Society, pp A2652–A2652

Papillon M, Hajij M, Myers A, Frantzen F, Zamzmi G, Jenne H et al (2023a) Topological deep learning challenge: design and results. In: Workshop on topology, algebra, and geometry in machine learning. vol 221 of Proceedings of machine learning research. PMLR, p 3–8

Papillon M, Sanborn S, Hajij M, Miolane N (2023b) Architectures of topological deep learning: a survey on topological neural networks. ArXiv preprint. arXiv:2304.10031

Peyré G, Cuturi M et al (2019) Computational optimal transport: with applications to data science. Found Trends Mach Learn 11(5–6):355–607

Polianskii V (2018) An investigation of neural network structure with topological data analysis [Master's Thesis]. KTH, School of Electrical Engineering and Computer Science (EECS)

Pun CS (2021) A sparse learning approach to relative-volatility-managed portfolio selection. SIAM J Financ Math 12(1):410–445

Pun CS, Lee SX, Xia K (2022) Persistent-homology-based machine learning: a survey and a comparative study. Artif Intell Rev 55(7):5169–5213

Qiu Y, Wei GW (2023a) Persistent spectral theory-guided protein engineering. Nat Comput Sci 3(2):149–163

Qiu Y, Wei GW (2023b) Artificial intelligence-aided protein engineering: from topological data analysis to deep protein language models. Briefings Bioinform 24(5):bbad289

Rathore A, Chalapathi N, Palande S, Wang B (2021) TopoAct: visually exploring the shape of activations in deep learning. Comput Graphics Forum 40(1):382–397

Reinauer R, Caorsi M, Berkouk N (2021) Persformer: a transformer architecture for topological machine learning. ArXiv preprint. arXiv:2112.15210

Reininghaus J, Huber S, Bauer U, Kwitt R (2015) A stable multi-scale kernel for topological machine learning. In: Conference on computer vision and pattern recognition. IEEE. pp 4741–4748

Rieck B, Togninalli M, Bock C, Moor M, Horn M, Gumbsch T et al (2019) Neural persistence: a complexity measure for deep neural networks using algebraic topology. In: International conference on learning representations, p x

Robins V (1999) Towards computing homology from approximations. Topol Proc 24:503–532

Sarpietro RE, Pino C, Coffa S, Messina A, Palazzo S, Battiato S et al (2022) Explainable deep learning system for advanced silicon and silicon carbide electrical wafer defect map assessment. IEEE Access 10:99102–99128

Senekane M, Matjelo NJ, Taele BM (2021) Improving short-term output power forecasting using topological data analysis and machine learning. In: International conference on electrical, computer and energy technologies. IEEE, pp 1–6

Shapanis A, Jones MG, Schofield J, Skipp P (2023) Topological data analysis identifies molecular phenotypes of idiopathic pulmonary fibrosis. Thorax 78(7):682–689

Singh G, Memoli F, Ishkhanov T, Sapiro G, Carlsson G, Ringach DL (2008) Topological analysis of population activity in visual cortex. J Vis 8(8):11

Singh Y, Farrelly CM, Hathaway QA, Leiner T, Jagtap J, Carlsson GE et al (2023) Topological data analysis in medical imaging: current state of the art. Insights Imaging 14(1):1–10

Solomon Y, Wagner A, Bendich P (2021) A fast and robust method for global topological functional optimization. In: International conference on artificial intelligence and statistics. PMLR, pp 109–117

Som A, Thopalli K, Natesan Ramamurthy K, Venkataraman V, Shukla A, Turaga P (2018) Perturbation robust representations of topological persistence diagrams. In: European conference on computer vision (ECCV), pp 617–635

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(56):1929–1958

Turner K, Mukherjee S, Boyer DM (2014) Persistent homology transform for modeling shapes and surfaces. Inf Inference 3(4):310–344

Uray M, Giunti B, Kerber M, Huber S (2023) Topological data analysis in smart manufacturing processes—a survey on the state of the art. ArXiv preprint. arXiv:2310.09319

Vukicevic M, Mosadegh B, Min JK, Little SH (2017) Cardiac 3D printing and its future directions. Cardiovasc Imaging 10(2):171–184

Wamil M, Hassaine A, Rao S, Li Y, Mamouei M, Canoy D et al (2023) Stratification of diabetes in the context of comorbidities, using representation learning and topological data analysis. Sci Rep 13(1):11478

Wang F, Liu H, Samaras D, Chen C (2020) Topogan: a topology-aware generative adversarial network. In: European conference on computer vision. Springer, NewYork, pp 118–136

Watanabe S, Yamana H (2021) Topological measurement of deep neural networks using persistent homology. Ann Math Artif Intell 90(1):75–92

Xia K, Li Z, Mu L (2017) Multiscale persistent functions for biomolecular structure characterization. Bull Math Biol 80(1):1–31

Yu Z, Su Y, Lu Y, Yang Y, Wang F, Zhang S et al (2023) Topological identification and interpretation for single-cell gene regulation elucidation across multiple platforms using scMGCA. Nat Commun 14(1):400

Zaheer M, Kottur S, Ravanbakhsh S, Poczos B, Salakhutdinov RR, Smola AJ (2017) Deep sets. In: Advances in neural information processing systems. vol 30, pp 3391–3401

Zhang Z, Li Y, Zhou W, Chen X, Yao W, Zhao Y (2021) TONR: an exploration for a novel way combining neural network with topology optimization. Comput Methods Appl Mech Eng 386:114083

Zhao Q, Ye Z, Chen C, Wang Y (2020) Persistence enhanced graph neural network. In: International conference on artificial intelligence and statistics. vol 108 of proceedings of machine learning research. PMLR, pp 2896–2906

Zhen Z, Chen Y, Segovia-Dominguez I, Gel YR (2022) Tlife-GDN: detecting and forecasting spatio-temporal anomalies via persistent homology and geometric deep learning. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, New York, pp 511–525

Zhou C, Dong Z, Lin H (2022) Learning persistent homology of 3D point clouds. Comput Graph 102:269–279

Zhu X, Vartanian A, Bansal M, Nguyen D, Brandl L (2016) Stochastic multiresolution persistent homology kernel. In: International joint conferences on artificial intelligence. pp 2449–2457

Zieliński B, Lipiński M, Juda M, Zeppelzauer M, Dłotko P (2020) Persistence codebooks for topological data analysis. Artif Intell Rev 54(3):1969–2009

Zomorodian A (2010) Fast construction of the Vietoris-Rips complex. Comput Graph 34(3):263–271

Authors and Affiliations

Ali Zia[1,2] · Abdelwahed Khamis[2] · James Nichols[1] · Usman Bashir Tayab[3] · Zeeshan Hayder[2] · Vivien Rolland[2] · Eric Stone[1] · Lars Petersson[2]

✉ Ali Zia
ali.zia@anu.edu.au

Abdelwahed Khamis
abdelwahed.khamis@data61.csiro.au

James Nichols
James.Nichols@anu.edu.au

Usman Bashir Tayab
usman.bashir.tayab@rmit.edu.au

Zeeshan Hayder
zeeshan.hayder@data61.csiro.au

Vivien Rolland
Vivien.Rolland@csiro.au

Eric Stone
Eric.Stone@anu.edu.au

Lars Petersson
Lars.Petersson@data61.csiro.au

[1] Biological Data Science Institute, Australian National University, 46 Sullivans Creek Rd, Acton, ACT 2601, Australia

[2] CSIRO, Building 801, Cnr of Dickson Way and, N Science Rd, Acton, ACT 2601, Australia

[3] School of Engineering, RMIT, Bundoora Campus, 264 Plenty Rd, Mill Park, VIC 3082, Australia