Journal of Machine Learning Research 20 (2019) 1-45 Submitted 6/18; Revised 3/19; Published 7/19

# Learning Representations of Persistence Barcodes

**Christoph D. Hofer** CHR.DAV.HOFER@GMAIL.COM
*Department of Computer Science*
*University of Salzburg*
*Salzburg, Austria*

**Roland Kwitt** ROLAND.KWITT@GMAIL.COM
*Department of Computer Science*
*University of Salzburg*
*Salzburg, Austria*

**Marc Niethammer** MN@CS.UNC.EDU
*Department of Computer Science*
*University of North Carolina*
*Chapel Hill, NC, USA*

**Editor:** Michael Mahoney

## Abstract

We consider the problem of supervised learning with summary representations of topological features in data. In particular, we focus on persistent homology, the prevalent tool used in topological data analysis. As the summary representations, referred to as barcodes or persistence diagrams, come in the unusual format of multi sets, equipped with computationally expensive metrics, they can not readily be processed with conventional learning techniques. While different approaches to address this problem have been proposed, either in the context of kernel-based learning, or via carefully designed vectorization techniques, it remains an open problem how to leverage advances in representation learning via deep neural networks. Appropriately handling topological summaries as input to neural networks would address the disadvantage of previous strategies which handle this type of data in a *task-agnostic* manner. In particular, we propose an approach that is designed to learn a *task-specific* representation of barcodes. In other words, we aim to learn a representation that adapts to the learning problem while, at the same time, preserving theoretical properties (such as stability). This is done by projecting barcodes into a finite dimensional vector space using a collection of parametrized functionals, so called structure elements, for which we provide a generic construction scheme. A theoretical analysis of this approach reveals sufficient conditions to preserve stability, and also shows that different choices of structure elements lead to great differences with respect to their suitability for numerical optimization. When implemented as a neural network input layer, our approach demonstrates compelling performance on various types of problems, including graph classification and eigenvalue prediction, the classification of 2D/3D object shapes and recognizing activities from EEG signals.

**Keywords:** Topological data analysis, persistent homology, topological summary, supervised learning, deep learning

## 1. Introduction

Over the past decade, concepts from the field of algebraic topology have evolved into computationally efficient algorithms to analyze data. Methods from this field are now succinctly summarized under the term *topological data analysis (TDA)* (Carlsson, 2009) and have found a broad range of applications, ranging from studying activity patterns of the visual cortex (Singh et al., 2008), breast cancer (Nicolau et al., 2011), the manifold of natural image patches (Carlsson et al., 2008), to analyzing brain artery trees (Bendich et al., 2016), 3D surfaces (Reininghaus et al., 2015; Li et al., 2014; Hofer et al., 2017b), clustering (Chazal et al., 2013b) and the recognition of 2D object shapes (Turner et al., 2014b). Arguably, the most prevalent method used in practice is *persistent homology* (see Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2004) which offers a concise summary representation of topological features in data. In short, persistent homology tracks topological changes as we analyze data at multiple "scales". As the scale changes, topological features (i.e., connected components, holes, etc.) appear and disappear. Persistent homology associates a lifespan to these features, resulting in a multi set of $(birth, death)$ tuples, typically visualized as a *barcode* or a *persistence diagram*. These *topological signatures* can offer complementary information that is not easily extractable by other methods. This opens up novel pathways to address learning problems based on topological information.

Despite the advantages of TDA in capturing topological invariants of data, the field is still largely disconnected from recent developments in machine learning. With respect to persistent homology, this can be attributed to the unusual data structure underlying topological signatures, that is, multi sets, and the associated (computationally intensive) metrics in that space. In fact, Mileyko et al. (2011) and Turner et al. (2014a) have investigated the theoretical properties of this metric space and shown that it is not easily amenable to statistical computations (for example, there is no unique Fréchet mean), or machine learning for that matter. Nevertheless, several works (see Section 3) have recently shown advances towards bridging the gap between machine learning and TDA; either in the context of kernel-based learning, or via suitable vectorization techniques for persistence diagrams / barcodes, based on algebraic ideas. However, both strategies typically come at the cost of computational complexity. In case of vectorization techniques, computational bottlenecks typically arise when computing the vectorization itself. In case of kernels, it is well-known that kernel-based learning (such as SVMs) scales poorly with sample size and sometimes even the kernel computation itself can be computationally challenging. Importantly, both strategies rely on an a-priori *fixed* representation of topological signatures.

With respect to the latter issue, the success of deep neural networks in vision (see Krizhevsky et al., 2012; He et al., 2016; Huang et al., 2017), or natural language processing (see Graves, 2013; Sutskever et al., 2014), has shown that it is preferable to learn *task-specific representations*, instead of hand-crafting them. While neural networks exhibit remarkable performance on many types of problems, input data with strong geometric structure, such as graphs or manifold-valued objects, pose considerable algorithmic and theoretical challenges. Topological summaries fall exactly into this category because of their nature as multi sets and the associated metrics. This has, so far, largely prevented principled approaches to use barcodes as input(s) to neural networks.

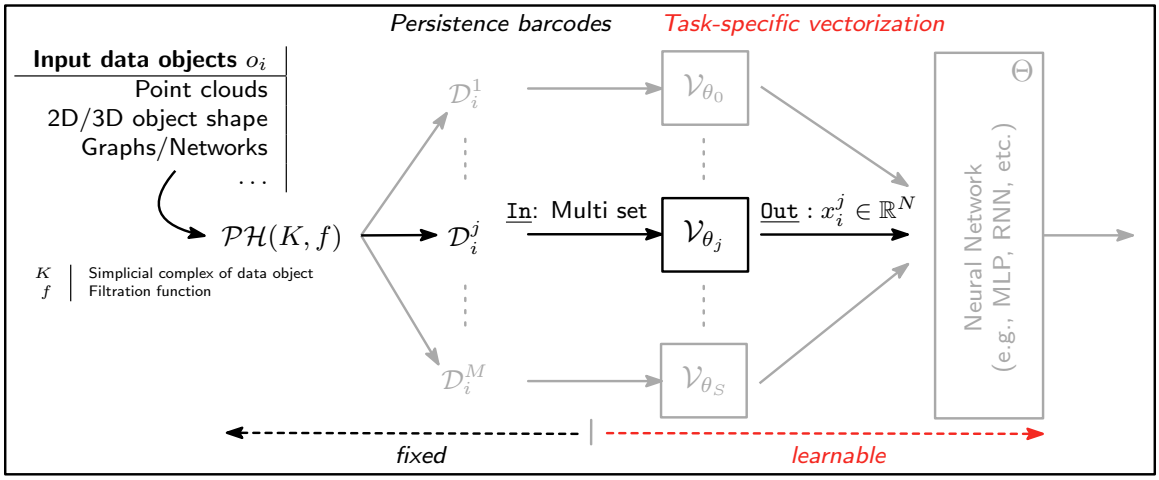



Figure 1: Illustration of *learnable, task-specific vectorizations of barcodes*. Inputs are barcodes $\mathcal{D}_i^j$, obtained by (1) constructing a simplicial complex $K$ from data object $o_i$ and (2) computing persistent homology ($\mathcal{PH}$) of a filtration of $K$. The superscript $j$ in $\mathcal{D}_i^j$ denotes that we can have multiple barcodes per $o_i$, e.g., considering homology groups of different dimension, or from multiple filtrations. Each barcode is a *multi set*, fed to one (or more) parametrized (by $\theta_j$) input layers that implement a mapping $\mathcal{V}_{\theta_j}$ to $\mathbb{R}^N$. These vectorizations are then fed to a neural network implementing, e.g., a discriminant classifier.

**Contributions**. This work extends Hofer et al. (2017a), where we introduced a neural network layer that can handle barcodes in a principled manner. The core idea is to project points in a barcode by a collection of parametrized functionals, so called structure elements. The parametrization is learned during training and allows to obtain a task-specific vectorization of barcodes (see Figure 1). In this work, we conduct an in-depth theoretical analysis of this approach. In particular, we prove that such a construction leads to an induced mapping of barcodes that is continuous with respect to the $p$-Wasserstein distance, although it *cannot* be stable with respect to $p$-Wasserstein for $p > 1$. This complements a similar recent result of Reininghaus et al. (2015) in the context of kernels. Further, we show that the original strategy of Hofer et al. (2017a) to enforce the required conditions on structure elements has limitations and present an alternative allowing a broader range of functional families to be considered. This then enables us to introduce new structure elements with desirable properties for learning. Finally, we present experiments on a variety of problems, including graph classification, eigenvalue prediction of (normalized) graph Laplacian matrices, 2D/3D shape recognition and the classification of EEG signals.

**Organization**. Section 2 introduces the required background material on persistent homology. Section 3 reviews related work on machine learning with topological signatures and highlights connections to our approach. Section 4 presents our main analysis of the construction scheme and Section 5 discusses its practical and theoretical aspects. Finally, Section 6 presents experimental results and Section 7 concludes the paper with a discussion of the most relevant points.

## 2. Background

For brevity, we only provide a brief overview of the mathematical concepts relevant to this work and refer the reader to Hatcher (2002) or Edelsbrunner and Harer (2010) for details.

**Homology.** The key idea of homology theory is to study the properties of some object $X$ by means of (commutative) algebra. In particular, we assign to $X$ a sequence of modules $C_n$ which are connected by homomorphisms $\partial_n$, that is,

$$\cdots \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \mathbf{0}$$

with

$$\partial_n : C_n \to C_{n-1} \quad \text{such that} \quad \operatorname{im} \partial_n \subseteq \ker \partial_{n-1} \ .$$

A structure of this form is called a *chain complex* and by studying its homology groups

$$H_n = \ker \partial_n / \operatorname{im} \partial_{n+1}$$

we can derive properties of $X$.

A prominent example of a homology theory is *simplicial homology*. Throughout this work, it is the used homology theory and hence we will now concretize the already presented ideas. Let $K$ be a simplicial complex and $K_n$ its $n$-skeleton. Then, we set $C_n(K)$ as the vector space generated (freely) by $K_n$ over $\mathbb{Z}/2\mathbb{Z}$[1]. The connecting homomorphisms

$$\partial_n : C_n(K) \to C_{n-1}(K)$$

are called boundary operators. For a $n$-simplex $\sigma = [x_0, \ldots, x_n] \in K_n$, we define them as

$$\partial_n(\sigma) = \sum_{i=0}^{n} [x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$$

and linearly extend this to $C_n(K)$, that is, $\partial_n(\sum \sigma_i) = \sum \partial_n(\sigma_i)$.

**Persistent homology.** Let $K$ be a simplicial complex. A sequence of simplicial complexes, $(K^i)_{i=0}^m$, such that

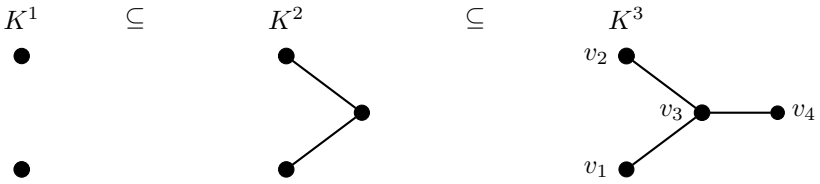
$$\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K$$

is called a *filtration* of $K$. If we use the extra information provided by the filtration of $K$, we obtain the following sequence of chain complexes

$$\begin{array}{ccccccccc}
\cdots & \xrightarrow{\partial_3} & C_2^1 & \xrightarrow{\partial_2} & C_1^1 & \xrightarrow{\partial_1} & C_0^1 & \xrightarrow{\partial_0} & \mathbf{0} \\
 & & \downarrow \iota & & \downarrow \iota & & \downarrow \iota & & \\
\cdots & \xrightarrow{\partial_3} & C_2^2 & \xrightarrow{\partial_2} & C_1^2 & \xrightarrow{\partial_1} & C_0^2 & \xrightarrow{\partial_0} & \mathbf{0} \\
 & & \vdots \iota & & \vdots \iota & & \vdots \iota & & \\
\cdots & \xrightarrow{\partial_3} & C_2^m & \xrightarrow{\partial_2} & C_1^m & \xrightarrow{\partial_1} & C_0^m & \xrightarrow{\partial_0} & \mathbf{0}
\end{array}$$

1. Simplicial homology is not specific to $\mathbb{Z}/2\mathbb{Z}$, but it's a typical choice, since it allows us to interpret $n$-chains as sets of $n$-simplices.

where $C_n^i = C_n(K_n^i)$, $\iota$ denotes the inclusion and $\mathbf{0}$ is the trivial group. To illustrate this more conveniently, we provide a simple *example*:




The chain groups in this example are as follows:

| For $K^1$ | For $K^2$ | For $K^3$ |
|---|---|---|
| $C_0^1 = [[v_1], [v_2]]_{\mathbb{Z}_2}$ | $C_0^2 = [[v_1], [v_2], [v_3]]_{\mathbb{Z}_2}$ | $C_0^3 = [[v_1], [v_2], [v_3], [v_4]]_{\mathbb{Z}_2}$ |
| $C_1^1 = \mathbf{0}$ | $C_1^2 = [[v_1, v_3], [v_2, v_3]]_{\mathbb{Z}_2}$ | $C_1^3 = [[v_1, v_3], [v_2, v_3], [v_3, v_4]]_{\mathbb{Z}_2}$ |
| $C_2^1 = \mathbf{0}$ | $C_2^2 = \mathbf{0}$ | $C_2^3 = \mathbf{0}$ |

This then leads to the concept of *persistent homology groups*, defined by

$$H_n^{i,j} = \ker \partial_n^i / (\operatorname{im} \partial_{n+1}^j \cap \ker \partial_n^i) \quad \text{for} \quad i \leq j \ .$$

The ranks, $\beta_n^{i,j} = \operatorname{rank} H_n^{i,j}$, of these homology groups (that is, the *$n$-th persistent Betti numbers*), capture the number of homological features of dimensionality $n$ (for example, connected components for $n = 0$, holes for $n = 1$, etc.) that persist from $i$ to (at least) $j$. In fact, according to (Edelsbrunner and Harer, 2010, Fundamental Lemma of Persistent Homology), the quantities

$$\mu_n^{i,j} = (\beta_n^{i,j-1} - \beta_n^{i,j}) - (\beta_n^{i-1,j-1} - \beta_n^{i-1,j}) \quad \text{for} \quad i < j \tag{1}$$

encode all the information about the persistent Betti numbers of dimension $n$. Before we continue to introduce barcodes, we take a short detour and discuss the concept of multi sets, a data structure that appears naturally when encoding information captured by persistent homology.

**Multi sets**. We focus on multi sets over a fixed domain. Informally, multi sets extend the concept of ordinary sets by allowing elements to occur more than once. For example, in the multi set $M = \{1, 1, 2\}$, the element 1 is contained 2 times. For multi sets over a fixed domain, $D$, this concept can be formalized by following the idea that *ordinary* sets over a fixed domain can be represented by their indicator function. Analogously, we can represent a multi set by its multiplicity function. Hence, a multi set $M$ can be interpreted as a function

$$M : D \to \mathbb{N}_0^\infty = \mathbb{N} \cup \{0, \infty\} \ ,$$

where $M(x) = n$ denotes that $x$ is contained in $M$ $n$-times. However, we will write

$$\operatorname{mult}_M(x) \text{ instead of } M(x)$$

to address the multiplicity of $x$ in $M$, since it is not common to write sets as functions. Based on this representation, we introduce the multi set operations needed in the subsequent parts of this work.

**Definition 1** *Let $M, N$ be multi sets and $A$ an ordinary set over the domain $D$. We adhere to the following conventions:*

*(M1)* $\operatorname{supp}(M) = \operatorname{supp}(\operatorname{mult}_M) = \{x \in D : \operatorname{mult}_M(x) > 0\}$ *Support*

*(M2)* $x \in M \Leftrightarrow x \in \operatorname{supp}(M)$ *Set membership*

*(M3)* $|M| = \sum_{x \in \operatorname{supp}(M)} \operatorname{mult}_M(x)$ *Cardinality*

*(M4)* $(M \uplus N)(x) = \operatorname{mult}_{M \uplus N}(x) = \operatorname{mult}_M(x) + \operatorname{mult}_N(x)$ *Union*

*(M5)* $(M \cap A)(x) = \operatorname{mult}_{M \cap A}(x) = \operatorname{mult}_M(x) \cdot \mathbb{1}_A(x)$ *Intersection with set*

*(M6)* $(M \setminus A)(x) = \operatorname{mult}_{M \setminus A}(x) = \operatorname{mult}_M(x) \cdot \mathbb{1}_{D \setminus A}(x)$ *Difference to set*

*(M7) Let $f : D \to \mathbb{R}^n$; then,* $\sum_{x \in M} f(x) = \sum_{x \in \operatorname{supp}(M)} \operatorname{mult}_M(x) \cdot f(x)$ *Function evaluation*

If necessary, we can always interpret an ordinary set as a multi set, since the characteristic function can be interpreted as a multiplicity function with values in $\{0, 1\}$.

**Topological signatures**. A typical way to obtain a filtration of a simplicial complex $K$ is to consider sub level sets of a real valued function, $f : K \to \mathbb{R}$ , defined on $K$ such that for $\rho \in K$ and the corresponding boundary operator, $\partial$, it holds that

$$f(\sigma) \leq f(\rho) \text{ for } \sigma \in \partial(\rho) \ .$$

Let

$$a_1 < \cdots < a_m \text{ with } m = |f(K)|$$

be the sorted sequence of values of $f(K)$. Then, we obtain $(K^i)_{i=0}^m$ by setting

$$K^0 = \emptyset \text{ and } K^i = f^{-1}\big((-\infty, a_i]\big) \text{ for } 1 \leq i \leq m \ .$$

If we construct a multi set such that, for $i < j$, the point $(a_i, a_j)$ is inserted with multiplicity $\mu_n^{i,j}$, see Eq. (1), we effectively encode the persistent homology of dimension $n$ with respect to the sub level set filtration induced by $f$.

**Definition 2 (Barcode)** *Let $\Omega = \{(b, d) \in \mathbb{R}^2 : d > b\}$ the upper-diagonal part of the real plane. A* barcode*, $\mathcal{D}$, is a multi set, over the domain $\Omega$. We denote by $\mathbb{D}$ the set of all barcodes with* finite *cardinality.*

**Remark 3** *While, in theory, barcodes are not necessarily finite, in any practical setup where data is analyzed, we deal with barcodes of finite cardinality. Thus, in all subsequent parts of this work, we assume $\mathcal{D} \in \mathbb{D}$.*

For a given complex $K$ of dimension $n_{\max}$ and a function $f$ (of the discussed form), we can interpret persistent homology ($\mathcal{PH}$) as a mapping

$$(K, f) \xmapsto{\mathcal{PH}} (\mathcal{D}^0, \ldots, \mathcal{D}^{n_{\max}-1}) \ ,$$

where $\mathcal{D}^i$ is the barcode of dimension $i$ and $n_{\max}$ is the dimension of $K$.

Equipping $\mathbb{D}$ with a metric structure, allows studying the sensitivity of $\mathcal{PH}$ to perturbations in $K$ or $f$. Before we can give the definitions of the metrics usually used for this purpose, we have to introduce the concept of *relative bijective matchings* between two multi sets.

**Definition 4 ($\Delta$-relative bijective matchings between finite multi sets)** *Let $M, N$ be finite multi sets over the domain $D$. Further, let $\Delta \subset D$ such that*

$$\operatorname{supp}(M) \cap \Delta = \emptyset \text{ and } \operatorname{supp}(N) \cap \Delta = \emptyset \ .$$

*A $\Delta$-relative* bijective matching *between $M$, $N$ is a multi set, $\varphi$, over $D \times D$ such that*

*(C1)* $\operatorname{supp}(\varphi) \cap (\Delta \times \Delta) = \emptyset$ ,

*(C2)* $\forall x \in \operatorname{supp}(M) : \sum_{(x,y)\in\operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) = \operatorname{mult}_M(x)$, *and*

*(C3)* $\forall y \in \operatorname{supp}(N) : \sum_{(x,y)\in\operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) = \operatorname{mult}_N(y)$ .

Intuitively, $\operatorname{mult}_\varphi\big((x,y)\big) = n$ means that $x$ is matched to $y$ $n$-times. Condition (C1) ensures that there are no matchings from $\Delta$ to $\Delta$, while (C2) and (C3) ensure that $x$, $y$ are used in a matching exactly as often as they are contained in $M$, $N$.

**Definition 5 (Bottleneck, Wasserstein distance)** *Let $(\mathbb{R}^2, \delta)$ be a metric space and let*

$$\Delta = \{(x,y) \in \mathbb{R}^2 : x = y\}$$

*be the diagonal of the real plane. For two barcodes $\mathcal{D}, \mathcal{E} \in \mathbb{D}$, the Bottleneck ($\mathrm{w}_\infty^\delta$) and Wasserstein ($\mathrm{w}_p^\delta$) distances (with respect to the metric $\delta$) are defined by*

$$\mathrm{w}_\infty^\delta(\mathcal{D}, \mathcal{E}) = \inf_\varphi \sup_{(x,y)\in\operatorname{supp}(\varphi)} \delta(x,y)$$

*and*

$$\mathrm{w}_p^\delta(\mathcal{D}, \mathcal{E}) = \inf_\varphi \left( \sum_{(x,y)\in\operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) \cdot \delta(x,y)^p \right)^{\frac{1}{p}} \text{ with } p \in [1,\infty) \ ,$$

*where the infimum is taken over all $\Delta$-relative bijective matchings $\varphi$ between $\mathcal{D}$ and $\mathcal{E}$.*

**Remark 6** *In the case where $\delta$ is the metric induced by the $q$-norm, $\|\cdot\|_q$, we write $\mathrm{w}_\infty$ for $\mathrm{w}_\infty^{\|\cdot\|_\infty}$ and $\mathrm{w}_p^q$ for $\mathrm{w}_p^{\|\cdot\|_q}$.*

We also remark that introducing the Wasserstein distances as in Definition 5, using $\Delta$-relative bijective matchings, deviates from the original formulation in Cohen-Steiner et al. (2007). However, this variant simplifies any of our subsequent analyses of stability/continuity properties involving those metrics. We refer the reader to (Cohen-Steiner et al., 2007, 2010; Chazal et al., 2009) for a selection of existing stability results in a broader context and further details.

**Remark 7** *By setting $\mu_n^{i,\infty} = \beta_n^{i,m} - \beta_n^{i-1,m}$, we extend Eq. (1) to features which never disappear, also referred to as "essential". This change can be lifted to $\mathbb{D}$ by setting $\Omega =$*

$\{(b,d) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\}) : d > b\}$. *However, if the filtration of $K$ is defined by the sub level sets of a function $f$, a more pragmatic way of handling essential features is to map their death time to the maximum of the function $f$. In many cases, for example height filtrations, this has a more natural interpretation (see Section 6.3).*

## 3. Related work

In order to deal with the practical inconveniences associated to a (direct) handling of barcodes in machine learning problems, several strategies have been proposed over the last couple of years. On a high level, these approaches can be categorized into (1) *kernel-based* techniques and (2) approaches that aim for a *vectorization*. Next, we summarize these techniques, draw connections between them, and elaborate on how they relate to our work. We additionally discuss related research on learning with (multi) set data structures and highlight important differences to our approach.

**Kernel-based techniques.** In short, the idea of kernel-based learning techniques (Schölkopf and Smola, 2001) is to rely on a positive-definite *kernel* function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that realizes an inner product $\langle \phi(x), \phi(y) \rangle_{\mathcal{G}} = k(x,y)$, for $x, y \in \mathcal{X}$, in a Hilbert space $\mathcal{G}$ for some (possibly unknown) feature mapping $\phi : \mathcal{X} \to \mathcal{G}$. A kernel can either be constructed by (1) defining a function $k$ that captures some notion of similarity between two input objects and showing its positive-definiteness, or (2) by explicitly constructing $\phi$ and using the inner product in $\mathcal{G}$ as a kernel (that is positive-definite by construction). Both strategies have found application in the context of kernel-based learning with topological signatures.

As a representative of the latter strategy, Reininghaus et al. (2015) introduced the *persistence scale-space (PSS) kernel*. The construction is based on first representing a diagram as a sum of Dirac deltas and then using this representation as the initial condition of a heat-diffusion process with a Dirichlet boundary condition on the diagonal $\Delta$. The solution of this partial differential equation (at time $t$) resides in $L^2(\Omega)$ and serves as an explicit feature mapping. The inner-product in $L^2(\Omega)$ is then used as a kernel function. A different, yet conceptually similar approach is taken by Bubenik (2015) to construct *persistence landscapes*, that is, functions of the form $\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R} \cup \{+\infty, -\infty\}$. Upon the definition of a suitable norm $\|\cdot\|_p$ on $\mathbb{N} \times \mathbb{R}$, landscapes map barcodes into a (separable) Banach space $L^p(\mathbb{N} \times \mathbb{R})$. Notably, for $p = 2$, we get the Hilbert space $L^2(\mathbb{N} \times \mathbb{R})$ which then facilitates to use the inner-product again to define a kernel. In Kwitt et al. (2015), it is further shown that under some mild restrictions on the barcodes (that is, birth-death boundedness and an upper-bound on the multiplicities of points), an exponentiated version of the persistence scale-space kernel is universal[2] (Steinwart and Christmann, 2008, Def. 4.52). A similar argumentation would give a universal kernel constructed from persistence landscapes, for $p = 2$. When working in this setting, statistical computations (see Gretton et al., 2012) become feasible, although, for landscapes many theoretical properties for statistics have already been developed (Chazal et al., 2013a, 2014; Fasy et al., 2014).

2. that is, the RKHS associated to the kernel is dense in the space of continuous functions $\mathbb{D} \to \mathbb{R}$.

A different kernel-based technique is introduced by Kusano et al. (2016), where the authors leverage the theory of reproducing kernel Hilbert space (RKHS) embeddings of probability measures (see Berlinet and Thomas-Agnan, 2004). Different to Reininghaus et al. (2015), a diagram is represented as a weighted measure, that is, a weighted sum of Dirac deltas, centered at each point of the diagram. The weighting function accounts for the different persistence of each point. Notably, the authors show stability of the kernel-induced distance between barcodes with respect to the Hausdorff distance (for two compact metric spaces embedded in the same metric space). Essentially, this is achieved via the aforementioned weighting function. Contrary to that, Reininghaus et al. (2015) achieve stability by enforcing the Dirichlet boundary condition on the diagonal. In the latter case, this only leads to stability of the kernel-induced distance with respect to $\mathrm{w}_1^q$ and it is shown that no kernel for which $k(\mathcal{F} \uplus \mathcal{G}, \mathcal{D}) = k(\mathcal{F}, \mathcal{D}) + k(\mathcal{G}, \mathcal{D})$ holds (with $\mathcal{F}, \mathcal{G}, \mathcal{D} \in \mathbb{D}$), is stable with respect to $\mathrm{w}_p^q$ for $p > 1$.

The recently proposed kernel approach of Carriére et al. (2017) follows the strategy of directly defining $k$, instead of explicitly constructing a feature map $\phi$. While there is strong indication (via counterexamples) that the negative of the $\mathrm{w}_p^q$ distance cannot be used in a construction of the form $k(\mathcal{F}, \mathcal{G}) = \exp(-\mathrm{w}_p^q(\mathcal{F}, \mathcal{G}))$, as $\mathrm{w}_p^q$ is not negative semi-definite, the authors circumvent this problem by introducing the *sliced Wasserstein distance* which can be shown to be negative semi-definite. The distance induced by the resulting kernel, termed the *sliced Wasserstein kernel*, is strongly equivalent to $\mathrm{w}_1^\infty$.

**Vectorization techniques.** A conceptually different line of developments is to "coordinatize" the barcode space which allows vectorization. In (Adcock et al., 2016), for example, the authors define a ring of algebraic functions on the space of barcodes. A similar approach with appealing properties is based on tropical algebraic geometry (Kališnik Verovšek, 2018). In particular, the coordinatizations in the latter work are stable with respect to $\mathrm{w}_p^q$ and $\mathrm{w}_\infty$ (which is not the case for the approach by Adcock et al. (2016)) and facilitates the development of sufficient statistics for barcodes (Monod et al., 2017). While this presents a promising approach, the challenge is to construct suitable polynomials in a computationally efficient manner.

Another instance of a *vectorization* technique is presented in Adams et al. (2017), where barcodes are mapped to so-called *persistence surfaces*. This is done by computing a weighted sum of normalized (isotropic) Gaussians, evaluated at each point in the diagram. Upon discretization of this *persistence surface*, one obtains the *persistence image* that can then be vectorized and fed to, for example, a linear SVM. Notably, a similar approach, motivated from a statistical point of view, was introduced by Chen et al. (2015) based on earlier ideas presented in Edelsbrunner et al. (2012). Similar to Kusano et al. (2016), stability of persistence images (with respect to $\mathrm{w}_1^\infty$) is achieved via a continuous piecewise-differentiable weighting function that evaluates to zero on the diagonal $\Delta$. By taking a measure-theoretic point of view, Chazal and Divol (2018) have recently shown that, in a wide range of situations, the persistence surface is a kernel density estimator for the expected barcode.

A different vectorization scheme is presented by Bendich et al. (2016), where the authors obtain a fixed-dimensional representation of barcodes by extracting the $N$ longest bars. Based on this representation, it is then possible to apply standard statistical techniques, for

example, PCA, or a discriminant classifier. However, it is unclear how the hyper-parameter $N$ should be chosen appropriately, a choice that will most likely depend on the application.

In summary, most vectorization and kernel-based techniques retain certain stability properties of persistence diagrams / barcodes with respect to the common metrics in the field of persistent homology. Yet, they also share one common *drawback*: the mapping of the topological signatures to a representation that is compatible with existing learning techniques is *pre-defined*. While this is, a-priori, not a disadvantage for statistical computations, it can be undesirable for learning, as the representation is *agnostic* to the learning task. In fact, the success of deep neural networks (see Krizhevsky et al., 2012; He et al., 2016) has shown that *learning* representations is a preferable approach. It is worth pointing out that algebraic constructions (Adcock et al., 2016; Kališnik Verovšek, 2018; Monod et al., 2017) might be amenable to learning task-specific representations, but it is unclear if this is computationally feasible. Furthermore, techniques based on kernels, such as (Reininghaus et al., 2015; Kwitt et al., 2015; Kusano et al., 2016; Carriére et al., 2017), additionally suffer scalability issues as training kernel SVMs scales poorly with the number of samples (Chapelle, 2007) and even evaluating the kernel function itself can be computationally challenging. In the spirit of end-to-end training, we therefore aim for a computationally efficient approach that allows to learn a *task-specific* representation.

**Learning with sets.** As mentioned earlier, learning with barcodes requires to appropriately handle *multi sets* and to respect the topology induced by the metrics. In the context of deep learning, using *multi sets* as input has gained limited attention so far, except for approaches where the multi set is the prediction target (Welleck et al., 2017). On the other hand, learning with *sets* as input to neural networks has spawned considerable research interest recently. In Qi et al. (2017a,b), for instance, the authors primarily focus on handling point clouds in a way that is permutation invariant with respect to the points. Zaheer et al. (2017) present a characterization theorem for valid set functions that allow the design of appropriate neural networks layers. In particular, it is shown that if set elements are from a *countable* domain, a necessary condition for a function to be a valid set function is to have a decomposition of the form $\rho(\sum_{x\in S}\phi(x))$, for a suitable choice of $\rho$ and $\phi$. In the *uncountable* case, this was only shown to hold for sets of fixed size, but it is conjectured that the previous characterization holds even for sets of arbitrary size. In this work, we are exactly in the latter setting, as we consider multi sets over an uncountable subset of $\mathbb{R}^2$ as input. Moreover, the fact that we are dealing with multi sets *and* a topology induced by the metrics in Definition 5 is crucial. It is, for example, straightforward to show (see Section 4.2) that blind application of the approach in Zaheer et al. (2017) does neither lead to a stable representation, nor is it clear how to appropriately handle points with higher multiplicities. Our construction scheme is specifically designed to avoid these problems.

## 4. Learning vectorizations of barcodes

This section is divided into two major parts. In the first part, we develop the intuition of the proposed (learnable) vectorization technique. Our intention is to give the reader a deeper

insight into the core ideas that motivate the construction scheme which we then introduce and analyze in the second part.

**Motivation**. As mentioned in Section 2, *barcodes* are multi sets residing over the upper diagonal part of the real plane, $\Omega$. From both a computational and an implementation perspective, the handling of (multi) sets poses considerable challenges, simply because there is no notion of order.

While many previous works have resorted to a mapping of barcodes to function spaces (with linear structure) in order to facilitate learning techniques, our strategy is to construct a learnable vectorization $\mathcal{V} : \mathbb{D} \to \mathbb{R}^N$ which operates on (*finite*) barcodes and respects the topology induced by the metrics in Definition 5.

### 4.1. Vectorization of multi sets

We start with a rather basic strategy to handle multi sets and then refine this idea through an analysis of the desired properties in the context of barcodes. First, consider a multi set $M$ over some fixed domain $D$. Then, a natural finite vectorization is to fix a finite subset of $D$, say $\{\mu_1, \ldots, \mu_N\}$, and set

$$\mathcal{V}(M) = \big(\mathrm{mult}_M(\mu_1), \ldots, \mathrm{mult}_M(\mu_N)\big) \ .$$

While possible, such a strategy is undesirable for multiple reasons. *First*, it is a rather strict representation, as for a multi set $M$ where $M \cap \{\mu_1, \ldots, \mu_N\} = \emptyset$, we would obviously get $\mathcal{V}(M) = (0, \ldots, 0)$. *Second*, and possibly more fundamental, such a mapping would be inherently discontinuous, as it can not capture continuous changes of the points in $M$ (if $D$ is equipped with a topology). This is easy to see by considering the example of $D = \mathbb{R}$. In that case, we have

$$\mathcal{V}(\{\mu_1, \mu_1 + \varepsilon\}) = (1, 0, \ldots, 0) \text{ for } \varepsilon > 0$$

if $(\mu_1 + \varepsilon) \notin \{\mu_1, \ldots, \mu_N\}$, however,

$$\mathcal{V}(\{\mu_1, \mu_1\}) = (2, 0, \ldots, 0) \ .$$

At first sight, given the obvious drawbacks of $\mathcal{V}$, continuing along this direction does not seem promising. However, the basic problem essentially resides in the (rigorously local) way of how elements of the multi set are represented by the multiplicity function. It is therefore interesting, to study a relaxed (that is, less local) version of this idea. In particular, consider a functional such that, for some metric $\delta$ residing on $D$,

$$s_{\mu_i} : D \to [0, 1] \ , \ s_{\mu_i}(\mu_i) = 1 \text{ and } \lim_{\delta(\mu_i, x) \to \infty} s_{\mu_i}(x) = 0 \tag{2}$$

for $1 \leq i \leq N$. Under this functional, we get $\mathrm{mult}_M(\mu_i) = \sum_{x \in M} s_{\mu_i}(x)$ as a boundary case. By further requiring that $s_{\mu_i}$ is continuous with respect to $\delta$ and by controlling the convergence speed of

$$s_{\mu_i}(x) \to 0 \quad \text{as} \quad \delta(\mu_i, x) \to \infty \tag{3}$$

we can reformulate a *relaxed*, but *continuous*[3] version of $\mathcal{V}$ as

$$\mathcal{V}(M) = \left( \sum_{x \in M} s_{\mu_1}(x), \ldots, \sum_{x \in M} s_{\mu_N}(x) \right) \quad . \tag{4}$$

Revisiting our initial example, we now obtain a more reasonable mapping of $\{\mu_1, \mu_1\}$ and $\{\mu_1, \mu_1 + \varepsilon\}$ as

$$\mathcal{V}(\{\mu_1, \mu_1\}) = (2, \varepsilon_2, \ldots, \varepsilon_N) \text{ for } \varepsilon_i > 0$$

and

$$\mathcal{V}(\{\mu_1, \mu_1 + \varepsilon\}) = (2 - \varepsilon_1', \varepsilon_2', \ldots, \varepsilon_N') \text{ for } \varepsilon_i' > 0 \ ,$$

respectively[4]. In other words, the mapping changes continuously with respect to the points in $M$. However, this comes at the cost of precision with respect to the multiplicity function.

### 4.2. From multi sets to barcodes

So far, we have introduced a first strategy for vectorizing multi sets which could already be applied to barcodes. However, $\mathbb{D}$ is not simply a collection of multi sets, but a *metric space* (with respect to, for example, $\mathrm{w}_\infty^\delta$ or $\mathrm{w}_p^\delta$) and hence has a topological structure. By keeping in mind that typical stability results in persistent homology are formulated with respect to those metrics, it seems imperative that the mapping $\mathcal{V}$ should be stable to at least a selection of those metrics. To highlight this, consider the following sequence of barcodes

$$\mathcal{D}_j = \{x_j\} \text{ with } x_j \in \Omega \text{ and } \lim_{j \to \infty} x_j \in \Delta \ .$$

For every choice of the proposed barcode distances, the empty barcode, $\emptyset$, is the limit of this sequence. A vectorization, $\mathcal{V}$, of the proposed form and continuous with respect to the metrical structure of the barcodes, $\mathbb{D}$, should yield

$$\mathcal{V}(\mathcal{D}_j) \to (0, \ldots, 0) = \mathcal{V}(\emptyset) \text{ for } j \to \infty \ .$$

Hence, in addition to *continuity* of $s$ on $\Omega \cup \Delta$, we should demand that

$$s(x) = 0 \text{ for } x \in \Delta \ .$$

**Remark 8** *As mentioned in Section 3, a similar (multi) set vectorization method is introduced in the "deep sets" approach of Zaheer et al. (2017) by setting*[5]

$$\mathcal{V}(M) = \sum_{x \in M} \phi(x) \text{ for } M \subset \mathbb{R}^m \ ,$$

*where $\phi : \mathbb{R}^m \to \mathbb{R}^n$ is a mapping implemented by a neural network (possibly through multiple layers). We highlight that this approach does not intrinsically respect the topological structure of $\mathbb{D}$. The reason is that $\phi$ is not constrained (or constructed) to vanish on the diagonal $\Delta$. Hence, we have no guarantee that the vectorization is continuous with respect to the introduced metrics.*

3. Continuity, at this point, is meant as continuity in the points of $M$.
4. For simplicity, we assume that the limit process in Eq. (3) is monotone.
5. In detail, we would have $\rho(\sum_{x \in M} \phi(x))$, but $\rho$ does not affect our argument.

### 4.3. Learnable vectorizations

Until now, we have developed a first idea how a *fixed* vectorization $\mathcal{V}$ could be constructed. However, most of the progress in supervised learning over the last years can be attributed to the fact that state-of-the-art approaches (such as deep neural networks) do not rely on fixed representations. Instead, they operate on a *family of representations* and aim to find a task-specific one for the problem at hand. This is achieved by back-propagating the error under a suitable loss function and adjusting the representation parameters to minimize the loss. The sought-for vectorization $\mathcal{V}$ (as introduced in the previous section) can be interpreted as such a representation. Hence, it seems beneficial to define a family of mappings, $\mathcal{V}_\theta$, and let the learner determine a suitable parametrization $\theta$. When learning via gradient descent, this demands (sub-)differentiability in the parameters $\theta$. If $\theta = (\theta_i)$ for a defined set of (real) parameters, we have to put an additional constraint on a practically useful vectorization, that is, the existence of the partial (sub-)derivative with respect to $\theta_i$,

$$\frac{\partial}{\partial \theta_i}\big(s_\theta(\mathcal{D})\big) \text{ for } \mathcal{D} \in \mathbb{D} \ .$$

### 4.4. Construction & Theoretical analysis

We propose a construction, based on parametrized functionals on $\Omega \cup \Delta$, for a vectorization that possesses the properties outlined in Sections 4.1 to 4.3. Our two main results, in Theorems 12 and 13, establish sufficient conditions such that the induced vectorization is (1) Lipschitz continuous with respect to $\mathrm{w}_1^\delta$ (that is, *stable* in the sense of Cohen-Steiner et al. (2010)) and (2) *continuous* with respect to $\mathrm{w}_p^\delta$. Further, we show that stability with respect to $\mathrm{w}_p^\delta$ for $p > 1$ is not possible. We begin by defining the notion of a *structure element*.

**Definition 9 (Structure element)** *Let $s$ be a family of continuous functionals, parametrized over some parameter space $\Theta \subset \mathbb{R}^d$, that is,*

$$\Theta \ni \theta \overset{s}{\mapsto} s_\theta$$

*such that*

$$s_\theta : \Omega \cup \Delta \to \mathbb{R} \text{ with } s_\theta(x) = 0 \text{ for } x \in \Delta \ .$$

*We call $s_\theta$ a structure element over the parameter space $\Theta$.*

Now, we introduce a mapping of barcodes with respect to $s_\theta$.

**Definition 10 (Induced mapping of barcodes)** *Let $s$ be a family of functionals as in Definition 9. Then, for each $s_\theta$, we define by*

$$s_\theta : \mathbb{D} \to \mathbb{R} \qquad \mathcal{D} \mapsto \sum_{x \in \mathcal{D}} s_\theta(x)$$

*the $s_\theta$-induced mapping of barcodes.*

The question is, if the induced mapping from Definition 10 allows to guarantee stability with respect to the Wasserstein distances, or a subset of them. To answer this, we provide a technical result (in Lemma 11) that allows us to shorten the proofs of Theorems 12 and 13.

**Lemma 11** *Let $\mathcal{D}, \mathcal{E} \in \mathbb{D}$ and let $\varphi$ be a $\Delta$-relative bijective matching between $\mathcal{D}$ and $\mathcal{E}$. Further, let $f : \Omega \cup \Delta \to \mathbb{R}$ with $f(x) = 0$ for $x \in \Delta$. Then,*

$$\sum_{x \in \mathcal{D}} f(x) - \sum_{y \in \mathcal{E}} f(y) = \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x,y)\big)\big(f(x) - f(y)\big) \ .$$

**Proof** see Appendix B.1

**Theorem 12 ($\mathrm{w}_1^\delta$ stability of the induced mapping)** *Let $s$ be a family of functionals as in Definition 9 and $s_\theta$ such that*

$$s_\theta \text{ is Lipschitz continuous w.r.t. } \delta \text{ and constant } K_{s_\theta} \ .$$

*Then, for two barcodes $\mathcal{D}, \mathcal{E} \in \mathbb{D}$, it holds that*

$$|s_\theta(\mathcal{D}) - s_\theta(\mathcal{E})| \leq K_{s_\theta} \cdot \mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E}) \ . \tag{5}$$

**Proof** Let $\varphi$ be a $\Delta$-relative bijective matching between $\mathcal{D}$ and $\mathcal{E}$, realizing $\mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E})$. As $\mathcal{D}, \mathcal{E}$ are assumed to be of finite cardinality it holds that $|\operatorname{supp}(\varphi)| < \infty$. Thus,

$$\begin{aligned}
|s_\theta(\mathcal{D}) - s_\theta(\mathcal{E})| &= \left| \sum_{x \in \mathcal{D}} s_\theta(x) - \sum_{y \in \mathcal{E}} s_\theta(y) \right| \\
&= \left| \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x,y)\big) \cdot \big(s_\theta(x) - s_\theta(y)\big) \right| && \text{(by Lemma 11)} \\
&\leq \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x,y)\big) \cdot \big|s_\theta(x) - s_\theta(y)\big| && \text{(by the triangle inequality)} \\
&\leq K_{s_\theta} \cdot \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x,y)\big) \cdot \delta(x,y) && \text{(by Lipschitz continuity of } s_\theta) \\
&= K_{s_\theta} \cdot \mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E}) \ .
\end{aligned}$$

■

Next, we show that the mapping induced by a structure element $s_\theta$ can not achieve stability for $\mathrm{w}_p^\delta$ with $p > 1$ (including $p = \infty$, that is, Bottleneck stability). Our argumentation is similar to Reininghaus et al. (2015, Theorem 3), where the authors characterize kernels on barcodes that cannot be stable with respect to $\mathrm{w}_p^q$ for $p > 1$. To this end, let $s_\theta$ be non-trivial, that is, there is some $x \in \Omega$ such that $s_\theta(x) > 0$ and define a sequence of barcodes based on $\mathcal{D} = \{x\}$ by

$$\mathcal{D}_j = \biguplus_{i=1}^{j} \mathcal{D} \text{ with } j \geq 1 \ .$$

Now consider

$$|s_\theta(\mathcal{D}_j) - s_\theta(\emptyset)| = \left| \sum_{x \in \mathcal{D}_j} s_\theta(x) - 0 \right| = j \cdot \left| \sum_{x \in \mathcal{D}} s_\theta(x) \right| = j \cdot |s_\theta(x)| \tag{6}$$

and observe that

$$\mathrm{w}_p^\delta(\mathcal{D}_j, \emptyset) = \begin{cases} \sqrt[p]{j} \cdot \mathrm{w}_p^\delta(\mathcal{D}_1, \emptyset), & p < \infty, \\ 1 \cdot \mathrm{w}_\infty(\mathcal{D}_1, \emptyset), & p = \infty \ . \end{cases} \tag{7}$$

As we can see from Eq. (6), the order of growth is linear. For Eq. (7), this only holds in case $p = 1$. Hence, for $p > 1$, we cannot find a constant such that Eq. (6) is bounded from above by Eq. (7). In particular,

$$|s_\theta(\mathcal{D}_j) - s_\theta(\emptyset)| > K \cdot \mathrm{w}_p^\delta(\mathcal{D}_j, \emptyset)$$

for $K \in \mathbb{R}$ and $j$ sufficiently large. While this is obviously a *negative* result, it does not necessarily mean that this adversely effects us in a learning setting . Nevertheless, although we cannot guarantee stability for $p > 1$, we can find sufficient conditions for a weaker property, namely *continuity* with respect to $\mathrm{w}_p^\delta$.

**Theorem 13 ($\mathrm{w}_p^\delta$ continuity of the induced barcode mapping)** *Let $s$ be a family of functionals as in Definition 9 such that $s_\theta$ satisfies the growth condition*

$$|s_\theta(x)| \leq \kappa_{s_\theta} \cdot \delta(x, y)^p \text{ for } x \in \Omega, y \in \Delta \text{ and } \kappa_{s_\theta} > 0 \ . \tag{8}$$

*Then, the induced barcode mapping is continuous with respect to $\mathrm{w}_p^\delta$.*

**Proof** Let $\mathcal{D} \in \mathbb{D}$ be arbitrary but fixed. Consider a sequence of barcodes, $(\mathcal{D}_j) \in \mathbb{D}$, converging to $\mathcal{D}$, that is,

$$\lim_{j \to \infty} \mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j) = 0 \ .$$

We have to show that $s_\theta(\mathcal{D}_j) \to s_\theta(\mathcal{D})$ for $j \to \infty$. Let $\varphi_j$ be a $\Delta$-relative bijective matching that realizes $\mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j)$. We get

$$\lim_{j \to \infty} \mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j)^p = \lim_{j \to \infty} \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \delta(x_j, y_j)^p = 0 \ . \tag{9}$$

Thus, for $(x_j, y_j) \in \mathrm{supp}(\varphi_j)$, $\delta(x_j, y_j)$ converges to 0 for $j \to \infty$. In the following, we show that for $\varepsilon > 0$ it holds that $|s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| \leq \varepsilon$. In particular, consider

$$\begin{aligned} |s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| &= \left| \sum_{x \in \mathcal{D}} s_\theta(x) - \sum_{y \in \mathcal{D}_j} s_\theta(y) \right| \\ &= \left| \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big(s_\theta(x_j) - s_\theta(y_j)\big) \right| \quad \text{(by Lemma 11)} \\ &\leq \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big| \quad \text{(triangle inequality)} \\ &= C_j \ . \end{aligned}$$

To proceed, we split the matching $\varphi_j$ into two disjoint parts, that is,

$$A_j = \mathrm{supp}(\varphi_j) \cap \big(\Omega \times (\Omega \cup \Delta)\big) \text{ and } B_j = \mathrm{supp}(\varphi_j) \cap \big(\Delta \times (\Omega \cup \Delta)\big) \ .$$

$A_j$ contains the matchings from $\mathcal{D}$ to either points in $\mathcal{D}_j$ or $\Delta$, while $B_j$ contains the remaining matchings from $\Delta$ to $\mathcal{D}_j$. This allows us to represent $C_j$ by $A_j$ and $B_j$, that is,

$$C_j = \underbrace{\sum_{(x_j,y_j)\in A_j} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big|}_{D_j} + \underbrace{\sum_{(x_j,y_j)\in B_j} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big|}_{E_j} .$$

First, we see that

$$\begin{aligned} D_j &\leq \sum_{(x_j,y_j)\in A_j} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \max_{(x_j,y_j)\in A_j} \big|s_\theta(x_j) - s_\theta(y_j)\big| \\ &= |\mathcal{D}| \cdot \max_{(x_j,y_j)\in A_j} \big|s_\theta(x_j) - s_\theta(y_j)\big| && \text{(by (C2) of Definition 4)} \\ &\leq \varepsilon/2 \end{aligned}$$

for $j$ sufficiently large, as $\delta(x_j, y_j) \to 0$ and $s_\theta$ is continuous on $\Omega$ (Definition 9). Second, to bound $E_j$, we observe the following: it holds that

$$\begin{aligned} E_j &= \sum_{(x_j,y_j)\in B_j} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \big|0 - s_\theta(y_j)\big| && (\text{as } x_j \in \Delta) \\ &\leq \sum_{(x_j,y_j)\in B_j} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \kappa_{s_\theta} \cdot \delta(x_j,y_j)^p && \text{(by Eq. (8))} \\ &\leq \kappa_{s_\theta} \cdot \sum_{(x_j,y_j)\in \operatorname{supp}(\varphi_j)} \operatorname{mult}_{\varphi_j}\big((x_j,y_j)\big) \cdot \delta(x_j,y_j)^p && (\text{as } B_j \subset \operatorname{supp}(\varphi_j)) \\ &= \kappa_{s_\theta} \cdot \mathrm{w}_p^\delta(\mathcal{D},\mathcal{D}_j)^p && \text{(by Definition 5)} \\ &\leq \varepsilon/2 && \text{(by Eq. (9))} \end{aligned}$$

for $j$ sufficiently large. Overall, we get

$$|s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| \leq D_j + E_j \leq \varepsilon$$

for $j$ sufficiently large, which concludes the proof.

■

Overall, our theoretical analysis of the proposed construction scheme shows that the induced mapping of barcodes has favorable properties, as long as the discussed conditions on the structure element are satisfied. Based on these developments, it is now straightforward to introduce a neural network module that can handle barcodes as input and extend this scheme to multiple families of functionals.

**Definition 14 (Neural network input layer)** *Let $s$ be a family of functionals over the parameter space $\Theta$ and $\theta = (\theta_1, \dots, \theta_N)$ a vector of parameter realizations in $\Theta^N$. Then,*

$$\begin{aligned} \mathcal{V}_s : \Theta^N &\to \{f : \mathbb{D} \to \mathbb{R}^N\} \\ \theta &\mapsto \mathcal{V}_{s,\theta} \end{aligned}$$

*with*

$$\begin{aligned} \mathcal{V}_{s,\theta} : \mathbb{D} &\to \mathbb{R}^N \\ \mathcal{D} &\mapsto \big(s_{\theta_1}(\mathcal{D}), \dots, s_{\theta_N}(\mathcal{D})\big) \end{aligned}$$

*gives a mapping that is implementable as a neural network (input) layer.*

Notably, Lipschitz continuity of the input layer $\mathcal{V}_{s,\theta}$ with respect to $\mathrm{w}_1^\delta$ is a simple consequence of Theorem 12.

In the following part of the paper, we take a closer look at the practical aspects of the proposed construction scheme.

## 5. From theory to practice

In addition to the conditions on the structure elements, we now additionally require that the elements are (sub-)differentiable in their parameter space to facilitate numerical optimization in a gradient descent manner (see Section 4.3). In the context of neural networks, this allows to learn a task-specific representation of barcodes via error backpropagation. Taking this into consideration and setting $\delta = \|\cdot\|_q$, we arrive at the following conditions for a *practical* structure element $s_\theta$:

(P1) $s_\theta(x) = 0$ for $x \in \Delta$

(P2) $s_\theta$ is Lipschitz continuous with respect to $\|\cdot\|_q$

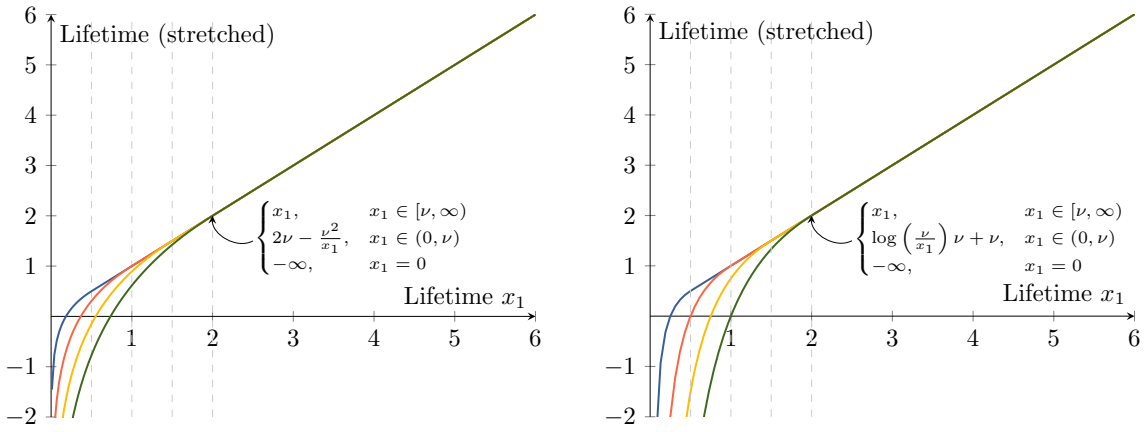
(P3) For $\mathcal{D} \in \mathbb{D}$, we require $s_\theta(\mathcal{D})$, $\theta \in \Theta$, to be differentiable on the parameter space $\Theta$

As shown in the previous section, conditions (P1) and (P2) suffice to guarantee that mapping $\mathcal{V}_{s,\theta}$, stated in Definition 14, is stable[6] with respect to $\mathrm{w}_1^q$. To develop structure elements and to verify the conditions, it will be helpful to rearrange and transform the coordinate axes of the barcode. We introduce this next.

### 5.1. Birth-lifetime coordinates

The intention of rearranging the coordinate axes to a *birth-lifetime* coordinate frame is to facilitate a straightforward analytic construction of structure elements that fulfill requirements (P1) to (P3). First, note that in the traditional notation we have $(x_0, x_1) \in \Delta \Leftrightarrow x_0 = x_1$. Hence, the predicate for $(x_0, x_1)$ to be contained in $\Delta$ depends on both $x_0$ and $x_1$. Shifting the coordinate system such that $(x_0', x_1')$ lies in the shifted version of $\Delta \Leftrightarrow x_1' = x_1 - x_0 = 0$

6. For $p > 1$, to guarantee continuity with respect to $\mathrm{w}_p^q$, the growth condition from Eq. (8) needs to hold.




(a) (b)

Figure 2: Stretching the *lifetime* axis of a birth-lifetime transformed (cf. Def. 15) barcode via (a) the transform proposed in Eq. (10) and (b) the log-stretching previously proposed by Hofer et al. (2017a). Different values of $\nu$ are marked by dashed lines. While, the differences appear subtle, the stretching as in (a) allows to guarantee Lipschitz continuity of *all* proposed structure elements (Section 5.4), whereas (b) only works for exponential structure elements (Section 5.4.1).

results in a notable simplification. We summarize this step in the following definition of a *birth-lifetime* coordinate transform, similar to (Adams et al., 2017).

**Definition 15 (Birth-lifetime coordinate transform)** *We define*

$$\rho : \Omega \cup \Delta \to \mathbb{R} \times [0, \infty) \qquad (b, d) \mapsto (b, d - b) \ .$$

*as the birth-lifetime coordinate transform of a barcode.*

The transform $\rho$ enables representing barcodes as multi sets over the upper half-plane of $\mathbb{R}^2$.

### 5.2. Rationally stretched birth-lifetime coordinates

Second, we introduce a transform, $\tau$, on top of $\rho$, such that we can represent barcodes as multi sets over $\mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$. We use the convention that for some function $f : \mathbb{R} \to \mathbb{R}$, $f(-\infty) = \lim_{x \to -\infty} f(x)$, if the limit exists and it is not defined otherwise.
The advantage of splitting the structure element into a composition of $\rho$, $\tau$ and $t_\theta$ is that is allows to use parametrized functionals $t_\theta$ defined on $\mathbb{R} \times \mathbb{R}$ as a starting point and then require that $t_\theta$ vanishes on $\mathbb{R} \times \{-\infty\} = \tau \circ \rho(\Delta)$. For example, most continuous probability density functions are subject to this constraint. As a consequence, we can use the rich pool of such functions to find possible candidates for $t$.

**Definition 16 (Rationally stretched birth-lifetime transform)** *For $0 < \nu$, we define the birth-lifetime transform*

$$\tau_\nu : \mathbb{R} \times [0, \infty) \to \mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$$

*by*

$$\tau_\nu((x_0,x_1)) = \begin{cases} (x_0,x_1), & x_1 \in [\nu,\infty), \\ (x_0, 2\nu - \frac{\nu^2}{x_1}), & x_1 \in (0,\nu), \\ (x_0,-\infty), & x_1 = 0 \ . \end{cases} \tag{10}$$

Note that $\tau_\nu$ is continuous on $\mathbb{R}^2$ as

$$\lim_{x\to\nu} x = \nu = \lim_{x\to\nu} 2\nu - \frac{\nu^2}{x} \ .$$

As a consequence, we can define a parametrized functional, say

$$t : \Theta \to \{f : \mathbb{R}\times(\mathbb{R}\cup\{-\infty\}) \to \mathbb{R}\} \qquad \theta \mapsto t_\theta \ ,$$

on $\mathbb{R}\times(\mathbb{R}\cup\{-\infty\})$ such that

$$s = t\circ\tau_\nu\circ\rho : \Theta \to \{f : \Omega\cup\Delta \to \mathbb{R}\} \qquad \theta \mapsto s_\theta = t_\theta\circ\tau_\nu\circ\rho$$

is a structure element iff $t\big((x_0,-\infty)\big) = 0$. A visualization of the proposed birth-lifetime transform is shown in Figure 2b, in comparison to the original (but limited) variant introduced in Hofer et al. (2017a).

### 5.3. Lipschitz continuity

As we are especially interested in a stability preserving vectorization, we want to construct structure elements which are Lipschitz continuous, see Section 4. While the transform $\tau_\nu\circ\rho$ facilitates the selection of $t$, it also comes at a price. A closer examination reveals that $\tau_\nu$ is not continuous on $\Delta$ and hence not Lipschitz continuous. As a consequence, Lipschitz continuity of some functional $t$ is not enough to guarantee that $t\circ\tau_\nu\circ\rho$ is Lipschitz continuous. The core of the problem is that $\tau_\nu\circ\rho$ diverges for $(x_0,x_1)\to(x_0,0)$. In order to preserve Lipschitz continuity, a chosen functional $t$ has to *converge rapidly enough* to dominate this divergence. In the following, we provide the essential argumentation to verify the Lipschitz condition for $t\circ\tau_\nu\circ\rho$. We start with two remarks to recall some properties of Lipschitz continuous functions.

**Remark 17** *To show that a multivariate function $f : \mathbb{R}^N \to \mathbb{R}$ is Lipschitz continuous with respect to $\|\cdot\|_q$ , $1 \le q \le \infty$, it is sufficient that Lipschitz continuity with respect to $\|\cdot\|_q$ holds in each coordinate. The reason is that coordinate-wise Lipschitz continuity with respect to $\|\cdot\|_q$ leads to Lipschitz continuity with respect to $\|\cdot\|_1$ by taking the maximum of the coordinate-wise Lipschitz constants. As all $q$-norms are equivalent, that is, $c\|\cdot\|_{q'} \le \|\cdot\|_q \le C\|\cdot\|_{q'}$ for $1 \le q \le q' \le \infty$ and $0 < c \le C$, this is sufficient. Additionally, concatenation preserves Lipschitz continuity, that is, if $f, g$ are Lipschitz continuous with respect to to some metric, so is $f\circ g$.*

Our strategy to verify that a construction of the form $t \circ \rho_\nu \circ \tau$ is Lipschitz continuous will be as follows: *First*, let $\nu > 0$ and $t : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be Lipschitz continuous. Then, it suffices to verify Lipschitz continuity of $t \circ \tau_\nu$ on $\mathbb{R} \times [0, \infty)$, as $\rho$ is trivially Lipschitz continuous. *Second*, for all following variants of $t$ in Section 5.4, it will be clear that $\mathbb{R} \times [\varepsilon, \infty), \varepsilon > 0$ poses no problem and it remains to study the behavior of $t \circ \tau_\nu$ on $\mathbb{R} \times [0, \varepsilon)$. Now, as all variants of $t$ will also be Lipschitz continuous in the first coordinate, $x_0$, only the second coordinate, $x_1$, needs to be considered (Remark 17) and our strategy boils down to show

$$\left| \frac{\partial}{\partial x_1} t \circ \tau_\nu \right| < C \in \mathbb{R} \text{ on } [0, \varepsilon) \ .$$

In all of the following cases, this will be equivalent to

$$\lim_{x \to 0} \left| \left( \frac{\partial}{\partial x_1} t \circ \tau_\nu \right)(x) \right| < C \in \mathbb{R} \ ,$$

as the partial derivative of $t \circ \tau_\nu$ will be continuous on $(0, \varepsilon)$.

### 5.4. Structure elements

Next, we use the previous ideas to introduce (radial) structure elements which result in $\mathrm{w}_1^q$ stable vectorizations. In particular, we start with a structure element based on the exponential function, analyze its properties, and then successively refine this element by switching to rational functions for better numerical stability and compatibility of the parameter gradients. Our main motivation is, to build structure elements that integrate well into a deep learning framework.

**Remark 18** *This section iteratively refines the structure element originally proposed by Hofer et al. (2017a), eventually resulting in the rational hat structure element of Section 5, Definition 23. We choose this type of presentation, as it elucidates certain design choices. However, those (intermediate) steps are not necessary to understand the final form of the refined version and may be skipped.*

#### 5.4.1. Exponential structure element(s)

Exponential functions have been previously used by Reininghaus et al. (2015), Kusano et al. (2016) and Adams et al. (2017) to weight points in barcodes. In (Reininghaus et al., 2015), the exponential function arises as the solution of a heat-diffusion problem, whereas in (Kusano et al., 2016) this is motivated by the idea of embedding measures into a reproducing kernel Hilbert space. The crucial difference to these approaches is that our structure elements *are not* at *fixed* locations (that is, one element per point of the particular barcode), but their locations and scales are adjustable.

**Definition 19 (Exponential structure element)** *We set* $\Theta = \mathbb{R}^2 \times \mathbb{R}^2$ *with* $((\mu_0, \mu_1), (\sigma_0, \sigma_1)) = (\mu, \sigma) \in \Theta$ *and define*

$$\begin{aligned} t^{exp} : \Theta &\to \{ f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+ \} \\ (\mu, \sigma) &\mapsto t^{exp}_{(\mu,\sigma)} \end{aligned}$$

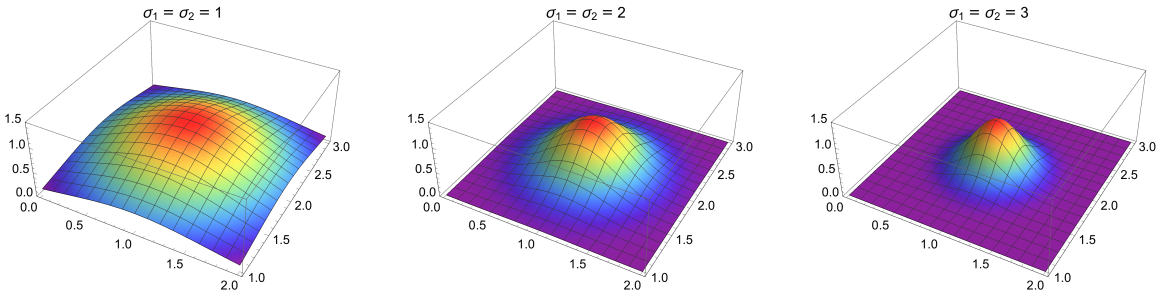



Figure 3: Illustration of an *exponential structure element*, centered at $\mu = (1, 2)$.

*with*

$$t^{exp}_{(\mu,\sigma)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+ \\ (x_0, x_1) \mapsto e^{-(\sigma_0^2(\mu_0 - x_0)^2 + \sigma_1^2(\mu_1 - x_1)^2)} \quad .$$

*We call* $s^{exp}_{\mu,\sigma} = t^{exp}_{\mu,\sigma} \circ \tau_\nu \circ \rho$ *an exponential structure element.*

Note that $s^{\texttt{exp}}_{\mu,\sigma}$ is well-defined, that is, it is a structure element in the sense of Definition 9. This is obvious, as

$$s^{\texttt{exp}}_{\mu,\sigma}(\Delta) = t^{\texttt{exp}}_{\mu,\sigma} \circ \tau_\nu \circ \rho(\Delta) = t^{\texttt{exp}}_{\mu,\sigma} \circ \tau_\nu(\mathbb{R} \times \{0\}) = t^{\texttt{exp}}_{\mu,\sigma}(\mathbb{R} \times \{-\infty\}) = \{0\}$$

holds by design. An illustration of an exponential structure element, centered at $\mu = (1, 2)$ is shown in Figure 3. The following lemma establishes Lipschitz continuity of this element.

**Lemma 20** $s^{exp}_{(\mu,\sigma)}$ *is Lipschitz continuous with respect to* $\|\cdot\|_q$ *on* $\Omega \cup \Delta$.

**Proof** see Appendix B.2

With respect to the differentiability of $s^{\texttt{exp}}_{(\mu,\sigma)}$, it is easy to see that $t^{\texttt{exp}}_{(\mu,\sigma)}(x)$ is differentiable in $\Theta$ for $x \in \mathbb{R}^2$. Therefore, so is $s^{\texttt{exp}}_{(\mu,\sigma)}(x)$ for $x \in \Omega \cup \Delta$. As the differential operator is additive, it follows that the induced barcode mapping, $\sum_{x \in \mathcal{D}} s^{\texttt{exp}}_{(\mu,\sigma)}(x)$, is differentiable.

### 5.4.2. Rational structure element(s)

The second structure element is inspired by rational functions of the form $(1 + x)^{-n}$. Our main intention is to simplify $s^{\texttt{exp}}$ such that we circumvent the exponential dependency of the diagram points. This is desirable in terms of numerical stability.

**Definition 21 (Rational structure element)** *We set* $\Theta = \mathbb{R}^2 \times \mathbb{R}^2 \times [1, \infty)$ *with* $((\mu_0, \mu_1), (\sigma_0, \sigma_1), \alpha) = (\mu, \sigma, \alpha) \in \Theta$ *and define*

$$t^{rat} : \Theta \to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+\} \\ (\mu, \sigma) \mapsto t^{rat}_{(\mu,\sigma,\alpha)}$$

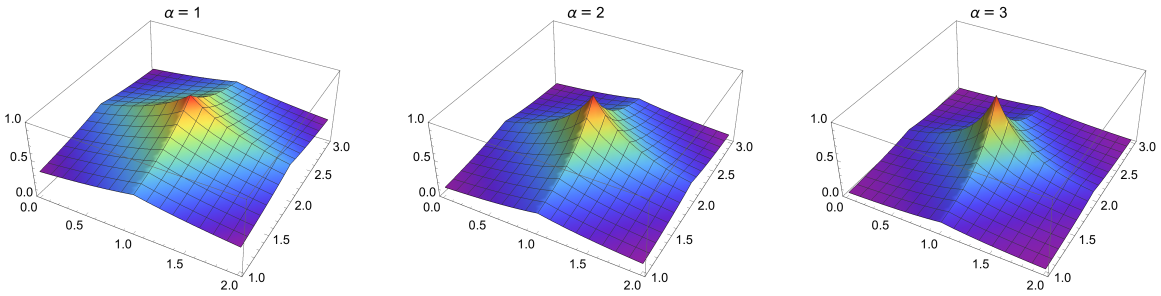



Figure 4: Illustration of a *rational structure element* for $\sigma_{0,1} = 1$ and various choices of $\alpha$.

*with*

$$t^{rat}_{(\mu,\sigma,\alpha)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}$$
$$(x_0, x_1) \mapsto \frac{1}{\left(1 + |\sigma_0| \cdot |x_0 - \mu_0| + |\sigma_1| \cdot |x_1 - \mu_1|\right)^{\alpha}} \ .$$

*We call* $s^{rat}_{(\mu,\sigma,\alpha)} = t^{rat}_{(\mu,\sigma,\alpha)} \circ \tau_\nu \circ \rho$ *a rational structure element.*

**Lemma 22** $s^{rat}_{(\mu,\sigma,\alpha)}$ *is Lipschitz continuous with respect to* $\|\cdot\|_q$.

**Proof** see Appendix B.3

An illustration of a rational structure element, centered at location $\mu = (1, 2)$ is shown in Figure 4. Similar to the exponential structure element, we see that $s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}$ is differentiable in $\mu_i$ and $\sigma_i$ for $i = 1, 2$. However, in case of $\alpha$, we have to introduce a slight restriction as $s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}$ is only differentiable for $\alpha \in (1, \infty)$. Regarding the partial derivatives, it is worth pointing out that

$$\frac{\partial}{\partial\alpha} s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}(x) \quad \text{and} \quad \frac{\partial}{\partial\theta} s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}(x) \quad \text{for } \theta \in \{\sigma_0, \sigma_1, \mu_0, \mu_1\}$$

are scaled differently. More precisely, let $f(\mu, \sigma, x)$ denote the denominator, that is,

$$s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}(x) = \frac{1}{f(\mu, \sigma, x)^{\alpha}} \ .$$

Then, we see that

$$\frac{\partial}{\partial\alpha} s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}(x) = \ln\left(\frac{1}{f(\mu, \sigma, x)^{\alpha}}\right) \cdot \frac{1}{f(\mu, \sigma, x)^{\alpha}}$$

differs from

$$\frac{\partial}{\partial\theta} s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}(x) = -\alpha \cdot \frac{1}{f(\mu, \sigma, x)^{\alpha+1}} \cdot \frac{\partial}{\partial\theta} f(\mu, \sigma, x) \quad \text{for } \theta \in \{\sigma_0, \sigma_1, \mu_0, \mu_1\}$$

with respect to the order of growth, depending on $f$. This yields the conjecture that it may be difficult to optimize both parameter groups $\{\alpha\}$ and $\{\sigma_0, \sigma_1, \mu_0, \mu_1\}$ simultaneously. A similar scaling of the parameter gradients is preferable, as it simplifies the optimization process.

#### 5.4.3. Rational hat structure element(s)

A closer look at $s^{\texttt{rat}}$ reveals that the purpose of the parameters $\sigma$ and $\alpha$ is to control the slope of $s^{\texttt{rat}}_{(\mu,\sigma,\alpha)}$ around $\mu$. To obtain a similar effect and to reduce the number of parameters, we gather this functionality into a single parameter $r$. We will see that this strategy yields a more balanced gradient scaling behavior. In detail, the idea is to use a uniformly scaled rational structure element and to subtract an antipode whose maximum is reached when the distance of a point to $\mu$ is exactly $r$. This leads to the following definition.

**Definition 23 (Rational hat structure element)** *We set $\Theta = \mathbb{R}^2 \times \mathbb{R} \times \mathbb{N}$ with $((\mu_0, \mu_1), r, q) = (\mu, r, q) \in \Theta$, $x = (x_0, x_1) \in \mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$ and define*

$$\begin{aligned} t^{\texttt{rat\_hat}} : \Theta &\to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}\} \\ (\mu, r, q) &\mapsto t^{\texttt{rat\_hat}}_{(\mu,r,q)} \end{aligned}$$

*with*

$$\begin{aligned} t^{\texttt{rat\_hat}}_{(\mu,r,q)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) &\to \mathbb{R} \\ x &\mapsto \frac{1}{1 + \|x - \mu\|_q} - \frac{1}{1 + |\,|r| - \|x - \mu\|_q|} \ . \end{aligned}$$

*We call $s^{\texttt{rat\_hat}}_{(\mu,\sigma,\alpha)} = t^{\texttt{rat\_hat}}_{(\mu,\sigma,\alpha)} \circ \tau_\nu \circ \rho$ rational hat structure element.*

To see that this structure element has the desired gradient behavior, lets consider the partial derivatives with respect to $r$ and $\mu_i$, that is,

$$\frac{\partial}{\partial r}\left(t^{\texttt{rat\_hat}}_{(\mu,r,q)}(x)\right)(r) = \frac{\operatorname{sgn}(|r| - \|x - \mu\|_q|)}{(1 + |\,|r| - \|x - \mu\|_q|)^2} \cdot \operatorname{sgn}(r)$$

and

$$\tfrac{\partial}{\partial \mu_i}\left(t^{\texttt{rat\_hat}}_{(\mu,r,q)}(x)\right)(\mu_i) = (-1) \cdot \left(\tfrac{1}{(1+\|x-\mu\|_q)^2} - \tfrac{\operatorname{sgn}(|r|-\|x-\mu\|_q|)}{(1+|\,|r|-\|x-\mu\|_q|)^2}\right) \cdot \tfrac{\partial}{\partial \mu_i}(\|x - \mu\|_q)(\mu_i) \ .$$

We observe that both terms have similar order of growth in $r$ and $\mu_i$, respectively. However, this comes at the price that we no longer have $s^{\texttt{rat\_hat}}_{(\mu,\sigma,\alpha)}(\mu) = 1$. While this could be repaired by a multiplicative scaling factor

$$c_r = \frac{1 + |r|}{|r|} \ ,$$

a closer look at

$$\frac{\partial}{\partial r} c_r(r) = -\frac{\operatorname{sgn}(r)}{|r|^2}$$

reveals that this should be avoided, as it would lead to an undesirable quadratic dependency of the denominator with respect to $r$.

**Lemma 24** *$t^{\texttt{rat\_hat}}_{(\mu,\sigma,\alpha)}$ is Lipschitz continuous with respect to $\|\cdot\|_q$.*

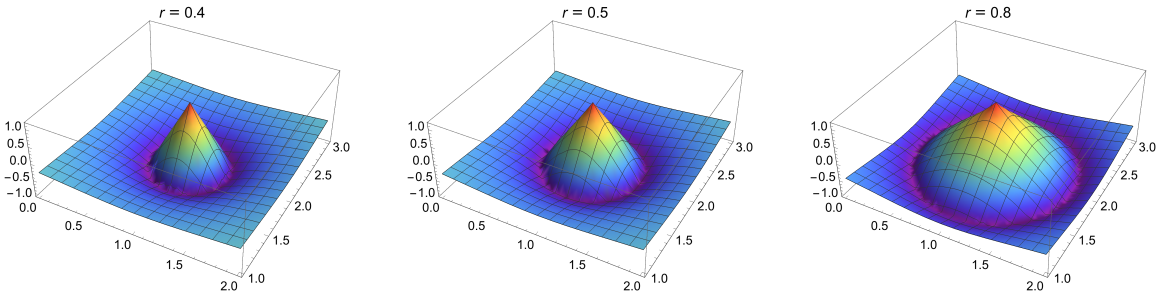



Figure 5: Illustration of a *rational hat structure element* for $\|\cdot\|_2$ and various choices of $r$.

We omit the proof for brevity, but note that it is similar to the proof of Lemma 22. Furthermore, $s^{\texttt{rat\_hat}}_{(\mu,\sigma,\alpha)}(\mathcal{D})$ is differentiable in its parameters. An illustration of this structure element, centered at $\mu = (1, 2)$, for different parameter settings is shown in Figure 5.

Overall, the advantages of $s^{\texttt{rat\_hat}}$ over $s^{\texttt{exp}}$ and $s^{\texttt{rat}}$ in terms of (1) the scaling of the derivatives and (2) its straightforward implementation, are particularly appealing for learning in the context of neural networks. Consequently, we use this element in all experiments that follow.

## 6. Experiments

We present a diverse set of experiments for supervised learning with different types of data objects. In particular, we show results for 2D/3D shape recognition and the classification of social network graphs, evaluated on standard benchmark data sets. Additionally, we present two *exploratory* experiments for the problems of predicting the eigenvalue distribution of normalized graph Laplacian matrices and activity recognition from EEG signals. These experiments demonstrate the versatility and predictive power of persistent homology. Our goal is not to necessarily outperform approaches that are specifically tailored to a problem, but to demonstrate that persistent homology combined with deep learning delivers competitive performance in a very generic setup.

All experiments were implemented in `PyTorch`[7], using `DIPHA`[8] (Bauer et al., 2014) and `Perseus`[9] (Mischaikow and Nanda, 2013) for persistent homology computations. Source code to fully reproduce the experiments is publicly-available[10].

### 6.1. Network architecture

We focus on a generic neural network architecture, shown in Figure 6, that only needs to be slightly adjusted per experiment.

7. https://github.com/pytorch/pytorch
8. https://bitbucket.org/dipha/dipha
9. http://people.maths.ox.ac.uk/nanda/perseus
10. https://github.com/c-hofer/jmlr_2019.git

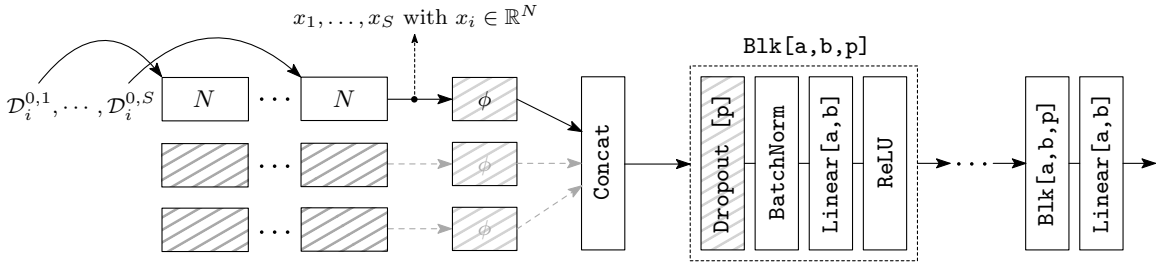



Figure 6: Illustration of our generic *network architecture*. Building blocks that are hatched are optional. One processing path (solid arrows) for handling $S$ barcodes $\mathcal{D}_i^{0,1}, \ldots, \mathcal{D}_i^{0,S}$ from *one* input object $o_i$ are shown. In particular, $\mathcal{D}_i^{0,j}$ denotes the $S$-th 0-dimensional barcode for $o_i$. Handling homology groups of higher dimension simply requires more input layers. The hyperparameter $N$ denotes the number of structure elements per input layer.

For each filtration and homology dimension, we need one input layer. In Figure 6, an example with $S$ 0-dimensional barcodes, $\mathcal{D}_i^{0,1}, \ldots, \mathcal{D}_i^{0,S}$, obtained from one input object $o_i$, is illustrated. Each input layer can have a varying number, $N$, of structure elements. We use the rational hat structure element (see Definition 23) for the following reasons: *First*, we only need to learn two parameters per structure element, that is, $\mu$ and $r$. *Second*, the gradients with respect to $(\mu, r)$ only differ by a linear term which we consider beneficial for learning. *Third*, commonly used building blocks of neural networks, for instance, batch normalization or rectified linear units (ReLU), expect both negative and positive inputs which is a property that only the rational hat element possesses. Overall, the output of one input layer is a vectorization in $\mathbb{R}^N$ that is then concatenated with the output of other input layers and fed into a multilayer perceptron (MLP). For optimization, we use stochastic gradient descent (SGD) with momentum and cross-entropy loss (if not mentioned otherwise). We additionally include a non-linearity $\phi : \mathbb{R} \to \mathbb{R}$ (in our case $\phi = \tanh$) after the input layer(s) to squash values in situations where points occur with high multiplicity; otherwise, $\phi = \mathrm{id}$. The parameter $\nu$ for the birth-lifetime transform of Definition 16 is fixed to 0.01 over all experiments[11]. We refer the reader to Table 6 for a full specification of the network configurations. Regarding initialization of the structure elements, we run $k$-means++ (Arthur and Vassilvitskii, 2007) clustering on all points from diagrams in the training portion of each dataset with $k = N$. The $N$ cluster centers are then used to initialize the position of the structure elements. The parameter $r$ is set to $1/4$ of the maximum lifetime of barcode elements in the training corpus. Different initializations of $r$ (within a reasonable range) only had a negligible impact on our results.

### 6.2. Baselines & Competitors

In all experiments on *benchmark data* (Sections 6.3 to 6.5), we compare against two baselines. *First*, we include a straightforward vectorization approach, following Bendich et al. (2016).

11. In principle, $\nu$ could be learned as well, but we did not explore that direction in this work.

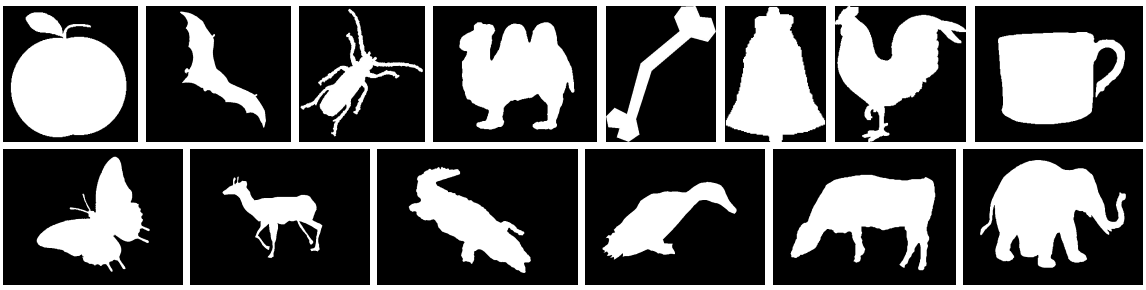


Figure 7: Some examples from the `MPEG-7` (*top*) and `Animal` (*bottom*) 2D shape datasets.

For each point $(b, d)$ in a barcode $\mathcal{D}$, we calculate its *persistence* $d - b$, sort the calculated persistences by magnitude from high to low and take the first $N$ values. Hence, for each barcode, a vector of dimension $N$ (if $|\mathcal{D} \setminus \Delta| < N$, we pad with zero) is obtained. Then, a linear SVM (denoted as $\texttt{SVM}^{\texttt{lin}}$) is trained with $N$ determined by cross-validation on the training portion of each dataset (we denote this as *Baseline* in all experiments). *Second*, we include a baseline-like approach where the parametrization of the structure elements in our approach is *only initialized* via $k$-means++ and then frozen during optimization. This allows to assess the impact of learning the parametrization of the structure elements.

We additionally compare against the *persistence images* approach from Adams et al. (2017) in two different variants. In its first incarnation, persistence images are computed on a $20 \times 20$ grid, vectorized (as originally suggested) and fed to a linear SVM. We use Gaussians centered at each point with the standard deviation determined via cross-validation (over $\sigma \in \{0.1, 0.5, 1.0\}$). In its second incarnation, we compute persistence images for the three choices of $\sigma$ listed above and then use these images as a *multi-channel* input (that is, each $\sigma$ yields an input channel) to a convolutional neural network (`NN`) with one convolution layer (using $3 \times 3$ kernels and a stride of 2), followed by two fully-connected layers (interleaved with ReLUs). This setup is similar to Cang and Wei (2017) where barcodes are vectorized and fed to a 1D convolutional neural network. We use Adam (Kingma and Ba, 2014) for optimization, train for 100 epochs with a batch size of 32, and use a learning rate of 0.01 (dropping to 0.001 after 50 epochs). This establishes a strong baseline and closely resembles our approach. For fair comparison, the number of (trainable) parameters in this network is comparable to the number of parameters in our architecture. We also found that including additional convolution layers did not further improve the results (in fact, unlike natural images, persistence images do not possess a compositional structure which would justify multiple convolution layers).

**Remark 25** *It is interesting to note that the construction of persistence images, in particular the construction of the persistence surface, is conceptually similar to the feature map constructed in the PSS kernel of Reininghaus et al. (2015). The key difference is that the feature map in the PSS kernel evaluates to zero on the diagonal by mirroring the Gaussians (placed on top of each point in the barcode), whereas persistence images use a suitable weighting function to achieve this. From this perspective, feeding persistence images to a convolutional neural network is akin to an approach that would feed a discretized (on a*

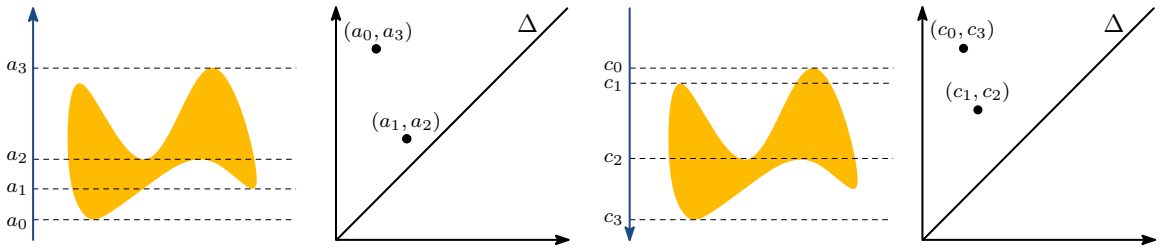



Figure 8: *Height function filtration* of a 2D shape from two directions, that is, $\xi = (0, 1)$ (*left*) and $\xi' = (0, -1)$ (*right*), with their corresponding 0-dimensional barcodes (visualized as persistence diagrams).

*grid) version of the PSS kernel feature map (or multiple feature maps, computed with varying PSS kernel scale parameter).*

### 6.3. Recognition of 2D shapes

We first test our approach on two different datasets of 2D object shape recognition: (1) the `Animal` dataset, introduced by Bai et al. (2009), which consists of 20 different animal classes, 100 samples each, and (2) the `MPEG-7` dataset which consists of 70 classes of different object/animal shapes, 20 samples each; see Latecki et al. (2000) for more details. For illustration, Figure 7 shows a selection of 2D object shapes from both datasets.

**Filtration.** The requirements to use persistent homology on 2D shapes are twofold: *First*, we need to assign a simplicial complex to each shape; *second*, we need to appropriately filter the complex. While, in principle, we could analyze contour features, such as curvature, and compute a sub level set filtration, such a strategy requires substantial preprocessing of the discrete data (for example, smoothing). Instead, we choose to work with the raw pixel data and leverage the *persistent homology transform*, introduced by Turner et al. (2014b). The filtration in that case is based on sub level sets of the *height function*, computed from multiple directions, illustrated in Figure 8. Practically, this means that we *directly construct a simplicial complex from the binary image*. We set $K_0$ as the set of all pixels which are contained in the object. A 1-simplex $[p_0, p_1]$ is in the 1-skeleton $K_1$ iff $p_0$ and $p_1$ are 4–neighbors on the pixel grid. To filter the complex, we denote by $\beta$ the barycenter of the object and by $r$ the radius of its bounding circle around $\beta$. For $[p] \in K_0$ and $\xi \in \mathbb{S}^1$, the filtration function is then given by

$$f([p]) = \frac{1}{r} \cdot \langle p - \beta, \xi \rangle \ ,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^2$. The filtration is lifted to $K_1$ by taking the maximum over the two vertices of each edge. Letting $\xi_i$ denote 16 equidistantly distributed directions in $\mathbb{S}^1$, starting from $(1, 0)$, we get a vector of barcodes $(\mathcal{D}^{0,j})_{j=1}^{16}$. Here, $\mathcal{D}^{0,j}$ is the barcode for 0-dimensional homology groups, obtained by filtration along $\xi_j$. As most objects do not differ in homology groups of *higher* dimensions ($> 0$), we did not use the corresponding barcodes. Furthermore, we only obtain *one* essential feature per direction, as

| | | MPEG-7 | Animal |
|---|---|---|---|
| ‡Skeleton paths (Bai et al., 2009) | | 86.7 | 67.9 |
| ‡Class segment set (Sun and Super, 2005) | | 90.9 | 69.7 |
| †ICS (Bai et al., 2009) | | 96.6 | 78.4 |
| †BCF (Wang et al., 2014) | | **97.2** | **83.4** |
| *Baseline* (Bendich et al., 2016) | $\texttt{SVM}^{\texttt{lin}}$ | 82.3 | 50.0 |
| PSS kernel (Reininghaus et al., 2015) | $\texttt{SVM}^{\texttt{kernel}}$ | 94.5 | 73.6 |
| Persistence images (Adams et al., 2017) | $\texttt{SVM}^{\texttt{lin}}$ | 85.1 | 61.1 |
| Persistence images (Adams et al., 2017) | NN | 92.3 | 69.4 |
| **Ours** (init. only, via $k$-means++) | NN | 92.2 | 72.9 |
| **Ours** (learned) | NN | 92.7 | 74.6 |

Table 1: Average cross-validation accuracies (in %) on the Animal and MPEG-7 benchmark datasets, compared to the two best (†) and two worst (‡) results reported in Wang et al. (2014), as well as different approaches that use persistent homology: our baselines from Section 6.2 and the PSS kernel from Reininghaus et al. (2015).

all objects only have one connected component. For practical reasons, the death time of those essential points is set to the maximum filtration value of the corresponding direction. This has a natural geometric interpretation, as the persistence of these essential points is the diameter of the object along a direction.

**Network configuration/training.** We use the architecture of Figure 6 with $\phi = \mathrm{id}$ and train with an initial learning rate of 0.01, a momentum of 0.9 and a batch size of 100 for 200 epochs. The learning rate is halved every 40-th epoch.

**Results.** Table 1 lists the average cross-validation accuracies over 10 random 90/10 splits. Regarding approaches that also leverage information from persistent homology, we include the approaches of Section 6.2, as well as the *PSS kernel* from Reininghaus et al. (2015). For the PSS kernel, we train a kernel SVM using the averaged kernel matrices obtained for each filtration direction. In case of persistence images, multiple filtration directions simply increase the dimensionality of the vectorized input to the linear SVM, or the number of input channels in the neural network variant, respectively. When using SVMs, the cost parameter $C$ is also determined via cross-validation on the training portion of each random split. Regarding non-topological strategies, we report the two best (†) and two worst (‡) results from Wang et al. (2014). While using topological signatures is below the top result on both datasets, we remark that *no* specific data preprocessing is required. This is in contrast to BCF or ICS which require contour extraction and are tailored to this type of data. Notably, in case of MPEG-7, the PSS kernel actually performs better than our approach. This can be attributed to the fact that the dataset is relatively small and we have to learn the parameters of our structure elements for all 16 input layers. In comparison, on the larger Animal dataset, learning a representation of barcodes already exhibits superior performance. We highlight that the proposed architecture readily generalizes to 3D with the only difference that, in this case, the filtration directions $\xi_i$ are in $\mathbb{S}^2$. Finally, we observe that persistence images

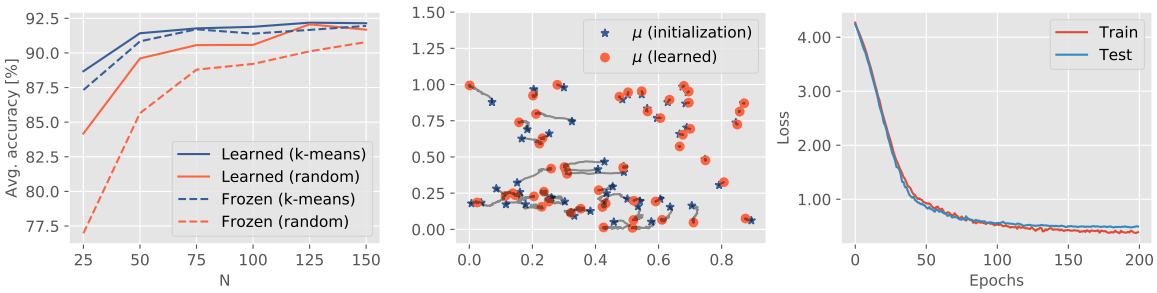



Figure 9: *Left*: Comparison, in terms of average cross-validation accuracy, of different initialization schemes, as a function of the number of structure elements, $N$, for two scenarios: (1) freezing the initial parameters and (2) learning the parametrization. *Middle*: Example of the change in location of the rational hat structure elements during training (for $N = 25$) and *random initialization*. *Right*: Training *vs.* testing loss over the training epochs for the same setup.

combined with a convolutional neural network, perform better than simple vectorization combined with a linear SVM. Also, initialization via $k$-means++ exhibits good performance on `MPEG-7`, but is inferior to learning the parametrization of the input layer on the `Animal` dataset.

### 6.3.1. Effect of learning a task-specific vectorization

On `MPEG-7`, we conduct a more detailed study on the impact of *learning* the parametrization of the proposed input layer *vs.* different initialization schemes and no learning. This means that we compare against a parametrization that is *only initialized*, frozen during training, and only the classification part (that is, everything after the `Concat` layer in Figure 6) of the network is trained. Notably, by construction, even a random initialization outputs a vectorization that is 1-Wasserstein stable, however it is not optimized for the learning task. We experimented with different numbers of structure elements $N$, as well as two initialization schemes: (1) random initialization and (2) initialization via $k$-means++ clustering (our default choice with $k = N$) of the points from all training diagrams. Figure 9 (*left*) shows the average cross-validation accuracy on `MPEG-7` as a function of the number of structure elements. *First*, we observe that initialization via $k$-means++ is superior to random initialization, no matter if the input layer is frozen or not. *Second*, we see that learning consistently improves performance, although, on this dataset and for small $N$, $k$-means++ initialization exhibits performance that is superior to random initialization and learning. This can be explained by the observation that random initialization typically requires a substantial change with respect to location which might not be achievable due to local minima in the energy landscape. Figure 9 (*middle*) shows an example of how the centers of each rational hat structure element change over the training epochs for $N = 25$ and randomly initialized locations. Finally, Figure 9 (*right*) shows an example of training vs. testing loss for the same parameter settings, revealing good generalization ability, as the testing loss closely tracks the training loss.

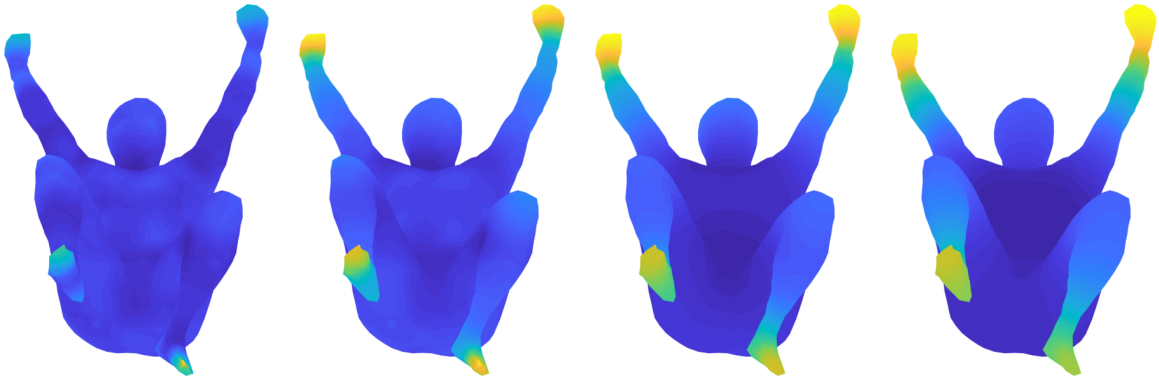


Figure 10: 3D surface meshes for one example object from the `SHREC14` evaluation benchmark (Pickup, D. et al., 2014). Vertices $x$ are colored by the heat-distribution $h(x,t)$ at increasing values of $t$ (from *left* to *right*). This relates to the scalar curvature of $x$ at scale $t$.

### 6.4. Recognition of 3D shapes

In this experiment, we consider the problem of 3D object recognition, based on features of the object's surface mesh. We use the real watertight 3D surface meshes that are part of the `SHREC14` benchmark and replicate the evaluation setup of Reininghaus et al. (2015). In particular, this data set contains 400 watertight meshes, split into 40 classes with 10 samples per class. Each class represents one human in different poses.

**Filtration**. For each mesh, we compute the heat-kernel signature (HKS) of Sun et al. (2009) at 10 discrete time points $t_i$, which captures the local heat distribution $h(x,t_i)$ at mesh vertex $x$ at scale $t_i$. Figure 10 illustrates the heat-distribution for one example object of the `SHREC14` benchmark as $t_i$ increases (from *left* to *right*). Similar to Section 6.5, we lift the heat-distribution function to higher-dimensional simplices by taking the maximum over all faces and compute persistent homology of the sub level set filtration. Notably, in the kernel-based approach of Reininghaus et al. (2015), the optimal choice of $t_i$ is determined via cross-validation and only diagrams for homology groups of dimension 0 are used[12]. In contrast, we use 0- *and* 1-dimensional homology and process the diagrams for each $t_i$ directly via 20 parallel input layers (that is, 10 for dimension 0 and 10 for dimension 1). This is a natural choice and allows the network to focus on information relevant to the learning task.

**Network configuration/training**. We use the generic architecture of Figure 6 with $\phi = \mathrm{id}$ and 20 input branches, that is, two input branches per HKS scale $t_i$. The network is trained for 300 epochs with a batch size of 20 and an initial learning rate of 0.5. The learning rate is halved every 20-th epoch.

**Results**. Table 2 lists cross-validation accuracies in comparison to specifically-tailored approaches, reported in Pickup, D. et al. (2014). We remark that results listed as `APT`, `supDLtrainR` and `R-BiHDM-s` are obtained in a leave-one-out cross-validation setup using a 1-nearest neighbor classifier, due to the fact that Pickup, D. et al. (2014) assess *retrieval*

12. In particular, it is mentioned that 0- and 1-dimensional homology produced comparable results.

| | | SHREC14 (*Real*) |
|---|---|---|
| ‡APT | | **84.5** |
| ‡supDLtrainR | | 79.3 |
| ‡R-BiHDM-s | | 68.5 |
| *Baseline* (Bendich et al., 2016) | $\texttt{SVM}^{\texttt{lin}}$ | 41.3 |
| †Persistence landscapes (Bubenik, 2015) | $\texttt{SVM}^{\texttt{kernel}}$ | 51.7 |
| †PSS kernel (Reininghaus et al., 2015) | $\texttt{SVM}^{\texttt{kernel}}$ | 62.7 |
| Sparse-TDA/Persistence images (Guo et al., 2018) | $\texttt{SVM}^{\texttt{lin}}$ | 68.8 |
| Persistence images Adams et al. (2017) | NN | 69.0 |
| **Ours** (init. only, via $k$-means++) | NN | 70.5 |
| **Ours** (learned) | NN | 72.2 |

Table 2: Average cross-validation accuracies (in %) on SHREC14 (*Real*), compared to the approaches discussed in Section 6.2, the best results (†) reported by Reininghaus et al. (2015) and a recent Sparse-TDA approach by Guo et al. (2018). For reference, we also list the top-3 approaches (‡) of Pickup, D. et al. (2014).

*performance*. This is obviously the extreme case of making maximum use of the training data, as only one sample is left out per cross-validation run. In comparison, all other results (using persistent homology) are averaged over 10 random 90/10 splits. While, leave-one-out evaluation would be computationally impractical in a neural network regime, we expect that our results could be better in such a setup due to an increased amount of available training data. As we replicate the evaluation setup from Reininghaus et al. (2015), we list the (best) results for the PSS kernel and persistence landscapes reported in this work. Additionally, we compare against the approaches listed in Section 6.2 and the Sparse-TDA approach of Guo et al. (2018) which is based on persistence images (Adams et al., 2017). Table 2 clearly shows that learning a task-specific vectorization of barcodes is beneficial.

### 6.5. Classification of social network graphs

Next, we consider the problem of classifying unlabeled and undirected graphs. A graph $\mathcal{G}$ is given by $\mathcal{G} = (V, E)$, where $V$ denotes the set of vertices and $E$ denotes the set of edges. This is already a valid simplicial complex. We evaluate our approach on the challenging task of classifying social networks, using the two largest benchmark datasets from Yanardag and Vishwanathan (2015), that is, `reddit-5k` (5 classes, ≈5k graphs) and `reddit-12k` (11 classes, ≈12k graphs). Each sample in these datasets represents a Reddit discussion graph and the classes indicate *subreddits* (for example, *worldnews*, *video*, etc.).

**Filtration.** The construction of a simplicial complex from $\mathcal{G} = (V, E)$ is straightforward: we set $K_0 = \{[v] \in V\}$ and $K_1 = \{[v_0, v_1] : \{v_0, v_1\} \in E\}$. We choose a simple filtration based on the *vertex degree*, that is, the number of incident edges to a vertex $v \in V$. Hence, for $[v_0] \in K_0$, we get $f([v_0]) = \deg(v_0)$ and again lift $f$ to $K_1$ by taking the maximum. Note

that chain groups are trivial[13] for dimension $> 1$, hence, all features in dimension 1 are *essential*. Figure 11 (*top*) shows an illustration of the basic filtration principle and three exemplary sub level sets of a randomly selected sample from `reddit-5k` (*bottom*).

**Network configuration/training.** In this experiment, the network of Figure 6 has three input branches: two for 0-dimensional features (non-essential and essential) and one for essential 1-dimensional features. We use $\phi = \tanh$, as we observe many points with high multiplicity, due to the fact that the filtration values are natural numbers. Empirically, using tanh exhibits beneficial optimization behavior compared to, for example, normalizing the filtration function by the maximum degree as in Hofer et al. (2017a). We train the network for 300 epochs with a batch size of 100 and an initial learning rate of 0.5. The learning rate is halved every 20-th epoch.

**Results.** Table 3 compares our strategy to various state-of-the-art approaches from the literature. In particular, we compare against (1) the graphlet kernel (GK) and deep graphlet kernel (DGK) results from Yanardag and Vishwanathan (2015), (2) the Weisfeiler-Lehmann (WL) graph kernel (Shervashidze et al., 2011), (3) the Patchy-SAN (PSCN) results from Niepert et al. (2016) and (4) a graph-feature + random forest approach (RF) from Barnett et al. (2016). As we can see, using information from persistent homology in our proposed setting considerably outperforms the current state-of-the-art on both datasets by a large margin. This is an interesting observation, as PSCN (Niepert et al., 2016), for instance, also relies on vertex degrees and an extension of the convolution operation to graphs. Regarding approaches that use persistent homology, we compare against the PSS kernel and the approaches of Section 6.2. Additionally, we evaluate our approach *with* and *without* using essential features. The reason for this is to establish a fair comparison, as essential features are typically discarded and it is unclear how to properly include them in other methods. From a computational perspective, the PSS kernel did not terminate within a reasonable time frame ($< 1$ day), due to the large number of points per barcode and the fact that the computing *one entry* of the PSS kernel matrix scales with $\mathcal{O}(N \cdot M)$, where $M, N$ denote the cardinalities of the two multi sets involved. As multiple kernel matrices need to be computed to cross-validate the PSS scale parameter, this becomes even more of a bottleneck. We remark that the results reported in Table 3 for our approach *without* essential features are different to (Hofer et al., 2017a); in fact, they are lower by $\approx 5$ points. As, in practice, one would use *all* available information (that is, *including* essential features), we focused on this case in terms of network design. The same architecture was then used for the comparative experiments (that is, *without* essential features), but could be adjusted to obtain results similar to (Hofer et al., 2017a), in particular, 49% (`reddit-5k`) and 38.5% (`reddit-12k`).

### 6.6. Predicting the eigenvalue distribution of normalized graph Laplacians

As a first exploratory experiment, we investigate the problem of predicting the distribution of eigenvalues from normalized graph Laplacian matrices. Technically, this problem is similar to the social network classification experiment of Section 6.5, but differs in the type of learning target. We consider this a practically relevant task, as for large graphs, exact computation

13. The reason for this is that there are no 2-simplices in this setting.

| | | reddit-5k | reddit-12k |
|---|---|---|---|
| GK (Yanardag and Vishwanathan, 2015) | | 41.0 | 31.8 |
| DGK (Yanardag and Vishwanathan, 2015) | | 41.3 | 32.2 |
| PSCN (Niepert et al., 2016) | | 49.1 | 41.3 |
| RF (Barnett et al., 2016) | | **50.9** | **42.7** |
| WL graph kernel (Shervashidze et al., 2011) | | 47.7 | 38.5 |
| *Baseline* (Bendich et al., 2016) | $\texttt{SVM}^{\texttt{lin}}$ | 45.2 | 31.6 |
| PSS kernel (Reininghaus et al., 2015) | $\texttt{SVM}^{\texttt{kernel}}$ | n/a | n/a |
| Persistence images (Adams et al., 2017) | $\texttt{SVM}^{\texttt{lin}}$ | 44.9 | 34.6 |
| Persistence images (Adams et al., 2017) | NN | 46.7 | 35.1 |
| **Ours** (w/o essential, learned) | NN | 44.5 | 34.1 |
| **Ours** (w/ essential, init. only, via $k$-means++) | NN | 53.5 | 45.8 |
| **Ours** (w/ essential, learned) | NN | 54.8 | 46.0 |

Table 3: Average cross-validation accuracies (in %) on the `reddit-5k` and `reddit-12k` benchmark datasets. The *middle/bottom* part of the table lists approaches that use persistent homology. For the PSS kernel, `n/a` denotes that computation of the Gram matrices (required to cross-validate the PSS scale parameter) did not terminate within a reasonable time window ($< 1$ day).

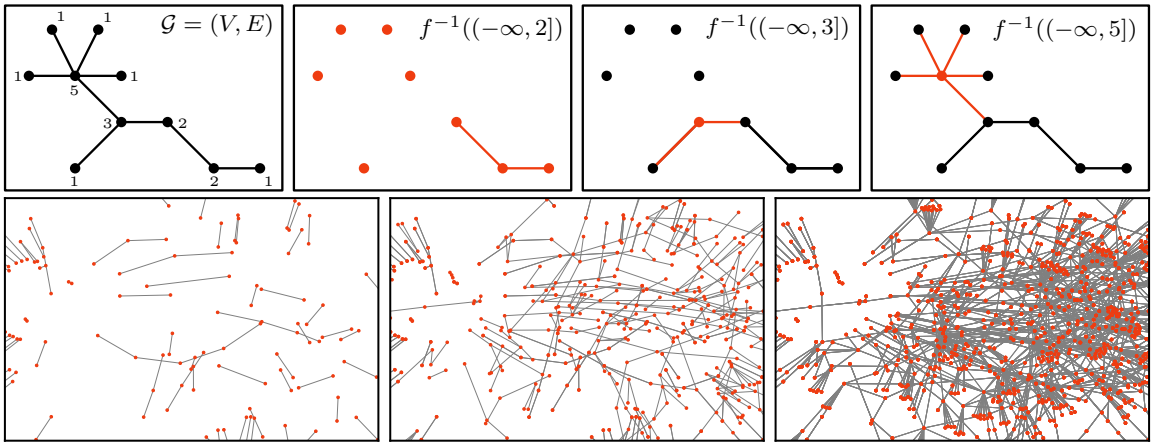



Figure 11: *Top:* Principle of graph filtration by vertex degree on a toy graph. Simplices that are added as the filtration value $a_i$ increases are marked in red. *Bottom:* one real `reddit_5k` graph for filtration values $a_i \in \{2, 5, 100\}$ (from *left* to *right*).

of all eigenvalues is computationally intensive and time-consuming. A relevant question in this context is, whether it is possible to learn on small graphs and predict on larger graphs. Accurate prediction of the eigenvalue spectrum could constitute a lightweight approach to assess the conformity of graph generation approaches with respect to real world data.

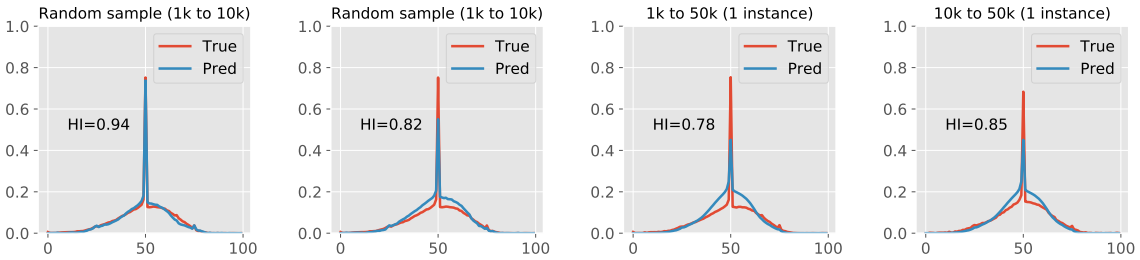



Figure 12: *Left*: two exemplary results for predicting the distribution (discretized to 100 bins) of normalized graph Laplacian eigenvalues. *Right*: prediction results for training on graphs of with $|V| = 10^3$ and $|V| = 10^4$ vertices and testing on a graph with $|V| = 5 \cdot 10^4$ vertices. HI denotes the value of the (symmetric) histogram intersection metric.

**Dataset**. To explore this problem, we use the `Pokec` (Takak and Zabovsky, 2012) social network graph that is part of the `SNAP` repository[14]. The full `Pokec` graph $\mathcal{G}(V, E)$ contains $|V| = 1,632,803$ vertices with $|E| = 30,622,564$ edges. We extracted two collections of subgraphs using breadth-first search (BFS), starting from randomly selected vertices of $V$ with the constraint that 1-hop neighbors of vertices are not picked as starting points for BFS to reduce subgraph overlap. The first collection, $S_1$ contains 5000 subgraphs with $10^3$ vertices, the second collection, $S_2$, contains 3000 subgraphs with $10^4$ vertices. For both, $S_1$ and $S_2$ exact eigenvalue computation is time-consuming, but still possible. For exploration, we also computed the eigenvalue spectrum for *one* subgraph, $\mathcal{G}^*$, with $5 \cdot 10^4$ vertices in the same manner (with computation time $\approx$ 1-2 days). Our goal is to predict a histogram of normalized graph Laplacian eigenvalues $\lambda_i$ (with $\lambda_i \in [0, 2]$). We chose a fixed binning of the eigenvalues into 100 bins, based on the eigenvalues of the largest graph and the Freedman-Diaconis rule for bin size (Freedman and Diaconis, 1981).

**Filtration**. The filtration is done as in Section 6.5, that is, by vertex degree. We only consider 0-dimensional persistent homology without essential features. The reason for this is that, due to the construction of the subgraphs (via BFS), we would only have one essential feature in $H_0$. Also, we were unable to handle 1-dimensional features, due to the vast number of cycles produced and the memory limitations of our GPU(s). Also, we did not want to introduce arbitrary thresholds for points in the barcode.

**Network training/configuration**. In this setup, we only have one input branch in the network of Figure 6 and $\phi = \tanh$. Different to previous sections, the loss function needs to be chosen differently. In particular, we decided to use the (symmetrized) histogram intersection (HI) metric (with range $[0, 1]$) as a loss function. A value of 0 indicates no overlap, a value of 1 indicates perfect overlap of two histograms.

**Results**. Three scenarios were evaluated: *First*, training on $S_1$ and testing on $S_2$, for which we obtain an average HI score of 0.85. This indicates remarkable performance, considering that the test set $S_2$ contains graphs that are one magnitude larger (in terms of vertices) than

14. https://snap.stanford.edu/data/soc-pokec.html

graphs in $S_1$. Figure 12 (*left*) shows two example prediction results. *Second*, for comparison, we trained on $S_2$ and tested on $S_2$ in a 90/10 cross-validation setup. In that case, the average HI increases to 0.94, indicating the larger graphs contain more structure that is relevant to the prediction task. When training on $S_1$, or $S_2$, respectively, and testing on $\mathcal{G}^*$, we obtain HI scores of 0.78 and 0.85, illustrated in Figure 12 (*right*). We consider this an encouraging result, as computing the filtration as well as one forward pass through the trained model only takes a fraction of the time required to compute the eigenvalues of $\mathcal{G}^*$.

### 6.7. Recognizing activity from EEG signals

As a second exploratory experiment, we investigate the problem of recognizing human activity, using recorded electroencephalography (EEG) signals. From an application point of view, this is a challenging problem and a lot of effort is put into developing brain-computer-interfaces (BCI) which rely solely on brain activity to, for example, facilitate robotic assistance systems for people with disabilities.

**Dataset**. The dataset we use in this experiment was collected by the Department of Neurology, Paracelsus Medical University Salzburg, Austria. It consists of 22 healthy participants as well as 7 participants in need of a wheelchair after spinal chord injury. The subjects were asked to execute 7 different tasks: *moving* of a hand/foot (2 tasks), *imagining to move* a hand/foot (2 tasks), *observing* a movie showing movement of a hand/foot (2 tasks) and *staying* in a resting condition, that is, no movement at all (1 task). In terms of recording the EEG signals, a geodesic sensor net with 256 channels was used. Consequently, each data point in the dataset is a collection of 256 time-intensity curves in combination with the corresponding task label. The EEG signal during each task was recorded over 25 trials where each trial was divided in 5 sub-runs, that is, repetitions of the given task. In total the dataset contains $\approx$ 31k such curves (some of the patients had more than one recording). Figure 13 shows example EEG signals of one sensor for three different actions.

**Filtration**. As persistent homology (partly) emerged from Morse theory, its application on time-intensity curves is straightforward. In our case, we interpreted the time-intensity sequences as a polyline, that is, edges are inserted between time-intensity points which are neighbors on the time axis. As a preprocessing step, intensities were mean and standard-deviation normalized. The resulting polylines were filtered along the intensity axis from top *and* bottom, leading to *two* barcodes for each channel of the EEG signal. For computational reasons, we used a sub-selection of 20 channels. This sub-selection represents a low-resolution sensor configuration. As only homology groups of dimension 0 are non-trivial for this data, we obtain two diagrams per sensor.

**Network configuration/training**. We use the architecture of Figure 6 with two input branches per sensor and $\phi = \mathrm{id}$. The network is trained for 100 epochs with a batch size of 300 and an initial learning rate of 0.1. The learning rate halved after every 10-th epoch.

**Results**. Table 4 lists the average cross-validation accuracies for 10 random 90/10 splits of the dataset. We chose to compare against our *Baseline*, as well as a linear SVM trained on mean and standard deviation normalized EEG sequences, reduced to 250 dimensions via

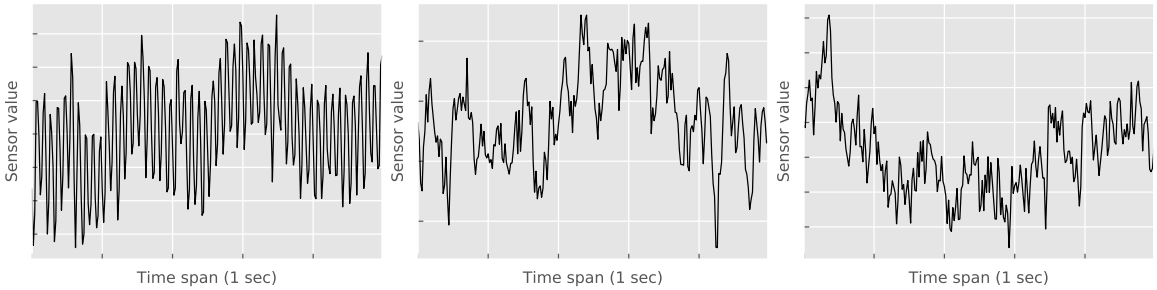



Figure 13: Exemplary EEG signals from one (out of 256) sensor for three different actions.

PCA (250 components already cover $> 99\%$ of the variance). The number of samples in this classification problem did not allow us to evaluate the PSS kernel within reasonable computation time ($< 1$ day). For persistence images, concatenation of the discretized persistence surface(s), even for a coarse $20 \times 20$ grid, leads to 16000 dimensional input vectors, which also posed computational problems. Hence, only persistence images in combination with a convolutional neural network (see Section 6.2) are included in Table 4 as a strong baseline. As we can see, the proposed approach exhibits considerably better performance than all competitors and, most importantly, easily handles datasets of this size with multiple parallel input paths.

| | | EEG sequences |
|---|---|---|
| PCA | $\texttt{SVM}^{\texttt{lin}}$ | **30.5** |
| *Baseline* (Bendich et al., 2016) | $\texttt{SVM}^{\texttt{lin}}$ | 21.1 |
| PSS kernel (Reininghaus et al., 2015) | $\texttt{SVM}^{\texttt{kernel}}$ | n/a |
| Persistence images (Adams et al., 2017) | $\texttt{SVM}^{\texttt{lin}}$ | n/a |
| Persistence images (Adams et al., 2017) | NN | 30.4 |
| **Ours** | NN | 37.4 |

Table 4: Average cross-validation accuracies (in %) on the EEG dataset, compared to a non persistent homology approach (PCA + linear SVM), our *Baseline*, as well as persistence images combined with a convolutional neural network, see Section 6.2. Approaches marked as n/a did not finish within 1 day of computation time.

## 7. Discussion

We presented an in-depth theoretical analysis of a recently proposed (Hofer et al., 2017a) approach to learn task-specific vectorizations of barcodes. This analysis has not only highlighted shortcomings of the original concept, but also revealed alternatives to mitigate these shortcomings. We summarize and discuss the main contributions next.

**Learnable vectorizations.** We started by studying the problem of how to vectorize multi sets over $\mathbb{R}^2$ in a parametrized manner. This led to the concept of structure elements which

induce a real-valued mapping of multi sets. In particular, $N$ structure elements lead to an $N$-dimensional vectorization upon concatenation. Different to existing approaches in the context of machine learning compatible representations of persistence diagrams, this mapping is not *fixed* a-priori, but can be optimized, for example, within a neural network regime. Our experimental study on datasets from different domains confirms that learning a vectorization of persistence diagrams is preferable over non-data driven approaches. Additionally, our results demonstrate that topological features of data can be beneficial in many learning tasks, not necessarily to replace existing inputs, but rather as a complementary source of discriminative information.

**Continuity properties.** We subsequently introduced conditions on the structure elements under which important continuity properties (such as stability) of persistence diagrams can be retained by the induced mapping. Our analysis also revealed (1) that we cannot achieve stability stronger than 1-Wasserstein ($\mathrm{w}_1^\delta$), but (2) the induced mapping is continuous with respect to $\mathrm{w}_p^\delta$ if a specific growth-condition is satisfied. We consider this particularly relevant, as it might open up the path to learn a filtration for a given simplicial complex. For example, in case of graphs, using the degree function is an ad-hoc choice. From a learning perspective, one would rather favor a strategy that learns a task-specific mapping from vertex meta-data $x \in \mathbb{R}^n$ to $\mathbb{R}$ which can then be used to compute sub level set filtration. This is certainly an interesting research direction for future work.

**Neural network integration.** From a practical perspective, it is imperative that (1) the required conditions on new structure elements can be verified easily and (2) that any new layer integrates well into existing network architectures. To address the first issue, we studied a decomposition of structure elements into a composition of two fixed coordinate transforms ($\tau_\nu$ and $\rho$) and a function $t$ defined on $\mathbb{R}^2$. This is beneficial, as new structure elements can be easily constructed through different choices of $t$. Further, verifying $\mathrm{w}_1^q$-stability of the induced mapping boils down to control the behavior of $t \circ \tau_\nu$ on a small part of the domain. Notably, this decomposition turned out to be equally beneficial for studying parameter gradients, as one can focus on the parameters of $t$. Starting from the popular Gaussian radial basis function (as our initial choice for $t$) and the observation that its parameter gradients greatly differ in terms of scaling, we introduced two radial alternatives with fewer parameters and gradients with comparable order of growth.

**Open problems.** While the proposed approach facilitates to leverage information from persistent homology in deep learning, it is unclear how to appropriately choose the number of structure elements. Although, empirically, we observe fairly stable performance with a reasonable number of elements, it would be desirable to make this decision in a more informed manner. Additionally, the particular functional form of the structure elements can most likely be further improved (or simplified), possibly with even more linear parameter gradients. Finally, we consider the problem of filtration learning to be of great interest, especially in cases where a fixed simplicial complex (for example, a graph or mesh) is given and the learner decides on an suitable filtration function for persistent homology.

## Acknowledgements

This work was partially funded by the Austrian Science Fund (FWF project KLI 00012 and FWF project P 31799), the Spinal Cord Injury and Tissue Regeneration Center Salzburg (SCI-TReCS), Paracelsus Medical University, Salzburg and NSF grant NSF EECS-1610762. We also thank Robert Elsässer and Gregor Bankhamer for providing the subgraphs for the experiments in Section 6.6, funded by the Austrian Science Fund (FWF project P 27613).

## Appendix A. Notation

| Symbol | Meaning |
|---|---|
| $\Delta$ | Diagonal $\Delta = \{(x_0, x_1) \in \mathbb{R}^2 : x_0 = x_1\}$ |
| $\Omega$ | Upper-diagonal part of $\mathbb{R}^2$, that is, $\Omega = \{(x_0, x_1) \in \mathbb{R}^2 : x_1 > x_0\}$ |
| $\mathcal{D}_i^{n,j}$ | Barcode of $n$-th homology group, $i$-th data object and $j$-th filtration |
| $\mathbb{D}$ | Set of barcodes with finite cardinality |
| $\mathrm{w}_p^q$, $\mathrm{w}_\infty$ | $p$-Wasserstein distance (using $\|\cdot\|_q$) and Bottleneck distance |
| $\langle x, y \rangle$ | Inner product in $\mathbb{R}^n$ for $x, y \in \mathbb{R}^n$ |
| $\mathbb{S}^n$ | $(n+1)$ dimensional unit sphere |
| $H_n$ | $n$-th homology group |
| $[v_0, \ldots, v_n]$ | $n$-dimensional simplex |
| $K_n$ | $n$-skeleton of a simplicial complex $K$ |
| $\mathcal{G}(V, E)$ | Undirected graph with vertex set $V$ and edge set $E$ |
| $[a, b], [a, b), (b, a], (a, b)$ | Closed, left-open, right-open and open interval for $a < b$, $a, b \in \mathbb{R}$ |

Table 5: Commonly used symbols used throughout the manuscript.

## Appendix B. Proofs

### B.1. Proof of Lemma 11

**Proof** To show this claim, we leverage the definition of a $\Delta$-relative bijective matching (cf. Definition 4) and a beneficial re-arrangements of the involved sums.

$$
\begin{aligned}
\sum_{x \in \mathcal{D}} f(x) - \sum_{y \in \mathcal{E}} f(y) &= \sum_{x \in \operatorname{supp}(\mathcal{D})} \operatorname{mult}_{\mathcal{D}}(x) f(x) - \sum_{y \in \operatorname{supp}(\mathcal{E})} \operatorname{mult}_{\mathcal{E}}(x) f(y) \\
&= \sum_{x \in \operatorname{supp}(\mathcal{D})} \left( \sum_{y \in \operatorname{supp}(\mathcal{E})} \varphi(x, y) \right) f(x) - \sum_{y \in \operatorname{supp}(\mathcal{E})} \left( \sum_{y \in \operatorname{supp}(\mathcal{D})} \varphi(x, y) \right) f(y) \\
&= \sum_{(x,y) \in \operatorname{supp}(\mathcal{D}) \times \operatorname{supp}(\mathcal{E})} \operatorname{mult}_{\varphi}\big((x, y)\big) f(x) - \sum_{(y,x) \in \operatorname{supp}(\mathcal{E}) \times \operatorname{supp}(\mathcal{D})} \operatorname{mult}_{\varphi}\big((x, y)\big) f(y) \\
&= \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x, y)\big) f(x) - \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x, y)\big) f(y) \\
&= \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_{\varphi}\big((x, y)\big) \big(f(x) - f(y)\big) \ . \qquad (\text{as } |\operatorname{supp}(\varphi)| < \infty)
\end{aligned}
$$

■

### B.2. Proof of Lemma 20

**Proof** As $t^{\texttt{exp}}_{(\sigma,\mu)}$ is Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$, we follow the argumentation of Section 5.3. That is, it suffices to show that

$$\lim_{x\to 0}\left|\left(\frac{\partial}{\partial x_1}t^{\texttt{exp}}_{(\sigma,\mu)}\circ\tau_\nu\right)(x)\right| < C \tag{11}$$

holds for $C\in\mathbb{R}$. Let $x<\varepsilon<\nu$. We get

$$\begin{aligned}
\left|\left(\frac{\partial}{\partial x_1}t^{\texttt{exp}}_{(\sigma,\mu)}\circ\tau_\nu\right)(x)\right| &= \left|\left(\frac{\partial}{\partial x_1}e^{-\sigma_0^2(x_0-\mu_0)^2-\sigma_1^2(2\nu-\nu^2/x_1-\mu_1)^2}\right)(x)\right| \\
&= |c_0|\cdot\left|\left(\frac{\partial}{\partial x_1}e^{-\sigma_1^2(2\nu-\nu^2/x_1-\mu_1)^2}\right)(x)\right| \\
&= |c_0|\cdot\left|e^{-\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2}\cdot 2\,(-\sigma_1^2)(2\nu-\nu^2/x-\mu_1)\cdot(-1)\frac{\nu^2}{x^2}\right| \\
&= |c_0|\cdot\left|c_1\,e^{(\cdots)}\cdot\frac{1}{x^2}+c_2\,e^{(\cdots)}\cdot\frac{1}{x^3}+c_3\,e^{(\cdots)}\cdot\frac{1}{x^2}\right| \\
&\leq 3\cdot|c_0|\cdot\max\{|c_1|,|c_2|,|c_3|\}\cdot\left|e^{-\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2}\cdot\frac{1}{x^3}\right| \\
&= c_4\cdot\underbrace{\left|\frac{\frac{1}{x^3}}{e^{\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2}}\right|}_{\text{I}}
\end{aligned}$$

for $x$ small enough (we aggregated constants into $c_i$, $0\leq i\leq 4$ for readability). Applying the de l'Hôpital's rule iteratively on I yields $\text{I}\to 0$. Consequently, Eq. (11) is satisfied. ■

### B.3. Proof of Lemma 22

**Proof** Similar to the proof of Lemma 20, we follow the argumentation of Section 5.3 and note that $t^{\texttt{rat}}_{(\sigma,\mu,\alpha)}$ is Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$.

Let $\varepsilon<\nu$ and $2\nu-\nu^2/\varepsilon<\mu_1$. Then, for $x<\varepsilon$

$$\left|2\nu-\frac{\nu^2}{x}-\mu_1\right| = \mu_1-2\nu+\frac{\nu^2}{x}$$

and we circumvent the fact that $|\cdot|$ is not differentiable at 0. Now consider

$$
\begin{aligned}
\left|\left(\frac{\partial}{\partial x_1} t^{\texttt{rat}}_{(\mu,\sigma,\alpha)}\right)(x)\right| &= \left|\left(\frac{\partial}{\partial x_1}\left(1+|\sigma_0|\cdot|x_0-\mu_0|+|\sigma_1|\cdot\left(\mu_1-2\nu+\frac{\nu^2}{x_1}\right)\right)^{-\alpha}\right)(x)\right| \\
&= \left|\left(\frac{\partial}{\partial x_1}\left(c_0+\frac{|\sigma_1|\nu^2}{x_1}\right)^{-\alpha}\right)(x)\right| \\
&= \left|-\alpha\cdot\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-\alpha-1}\cdot(-1)\frac{|\sigma_1|\nu^2}{x^2}\right| \\
&= c_1\cdot\left|\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-\alpha-1}\cdot\frac{1}{x^2}\right| \\
&\overset{*}{\leq} c_1\cdot\left|\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-1-1}\cdot\frac{1}{x^2}\right| \\
&= c_1\cdot\left|\frac{\frac{1}{x^2}}{\left(c_0^2+2\cdot c_0\cdot\frac{|\sigma_1|\nu^2}{x}+\frac{|\sigma_1|^2\nu^4}{x^2}\right)}\right| \xrightarrow{x\to 0} \frac{c_1}{|\sigma_1|^2\nu^4}
\end{aligned}
$$

where we collapse constant terms into $c_0, c_1$ and $(*)$ holds for $x \in (0, \varepsilon)$ small enough, as $y^{-(\alpha+1)} \leq y^{-2}$ for $\alpha \geq 1$ and $y$ large enough $\Rightarrow$ the partial derivative is bounded on $(0, \varepsilon)$. ■

## Appendix C. Neural network configurations

| **Dataset** | | $H_0$ | $H_0^e$ | $H_1^e$ | $\phi$ | **MLP** |
|---|---|---|---|---|---|---|
| MPEG-7 | N | 150 | - | - | id | Blk[2400,600,0.3] |
| | S | 16 | - | - | | Blk[600,150,0.2] |
| | | | | | | Lin[150,70,0.1] |
| Animal-7 | N | 100 | - | - | id | Blk[1600,400,0.3] |
| | S | 16 | - | - | | Blk[400,100,0.2] |
| | | | | | | Lin[100,20,0.1] |
| Reddit-5k | N | 150 | 5 | 50 | tanh | Blk[205,410,0.2] |
| | S | 1 | 1 | 1 | | Blk[410,104,0.2] |
| | | | | | | Blk[104,25,-] |
| | | | | | | Lin[25,5,-] |
| Reddit-12k | N | 150 | 5 | 50 | tanh | Blk[205,410,0.2] |
| | S | 1 | 1 | 1 | | Blk[410,104,0.2] |
| | | | | | | Blk[104,25,-] |
| | | | | | | Lin[25,12,-] |
| Pokec | N | 100 | - | - | tanh | Blk[100,100,-] |
| | S | 1 | - | - | | Blk[100,100,0.2] |
| | | | | | | Lin[100,100,-] |
| SHREC 2014 | N | 40 | - | - | id | Blk[800,800,-] |
| | S | 10 | - | - | | Blk[800,200,-] |
| | | | | | | Lin[200,40,-] |

Table 6: Configuration of the neural network in Figure 6 per experiment/dataset.

## References

H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *JMLR*, 18(8):1–35, 2017.

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

D. Arthur and S. Vassilvitskii. $k$-means++: The advantages of careful seeding. In *SODA*, 2007.

X. Bai, W. Liu, and Z. Tu. Integrating contour and skeleton for shape classification. In *ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, 2009.

I. Barnett, N. Malik, M.L. Kuijjer, P.J. Mucha, and J.-P. Onnela. Feature-based classification of networks. *CoRR*, 2016. https://arxiv.org/abs/1610.05868.

U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *ALENEX*, 2014.

P. Bendich, J.S. Marron, E. Miller, A. Pieloch, and S. Skwerer. Persistent homology analysis of brain artery trees. *Ann. Appl. Stat.*, 10(2), 2016.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, London, UK, 2004.

P. Bubenik. Statistical topological data analysis using persistence landscapes. *JMLR*, 16(1): 77–102, 2015.

Z. Cang and G.-W. Wei. Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.*, 13(7):e1005690, 2017.

G. Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46:255–308, 2009.

G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *IJCV*, 76:1–12, 2008.

M. Carriére, M. Cuturi, and S. Outot. Sliced Wasserstein kernel for persistence diagrams. In *ICML*, 2017.

O. Chapelle. Training a support vector machine in the primal. *Neural Comput.*, 19(5): 1155–78, 2007.

F. Chazal and V. Divol. The density of expected persistence diagrams and its kernel based estimation. In *SoCG*, 2018.

F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-Hausdorff stable signatures for shapes using persistence. *Comput. Graph. Forum*, 28(5):13931403, 2009.

F. Chazal, B.T. Fasy, F. Lecci, A. Rinaldo, A. Singh, and L. Wasserman. On the bootstrap for persistence diagrams and landscapes. *Model. Anal. Inform. Sist.*, 20(6):111–120, 2013a.

F. Chazal, L.J. Guibas, S.Y. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6):41–79, 2013b.

F. Chazal, B.T. Fasy, F. Lecci, A. Rinaldo, and L. Wassermann. Stochastic convergence of persistence landscapes and silhouettes. *JoCG*, 6(2):140161, 2014.

Y.-C. Chen, D. Wang, A. Rinaldo, and L. Wasserman. Statistical analysis of persistence intensity functions. *CoRR*, 2015. https://arxiv.org/abs/1510.02502.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007.

D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have $L_p$-stable persistence. *Found. Comput. Math.*, 10(2):127–139, 2010.

H. Edelsbrunner and J. L. Harer. *Computational Topology : An Introduction*. American Mathematical Society, 2010.

H. Edelsbrunner, D. Letcher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, 2002.

H. Edelsbrunner, A. Ivanov, and R. Karasev. Current open problems in discrete and computational geometry. *Modelirovanie i Analiz Informats. Sistem*, 19(5):5–17, 2012.

B. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence sets for persistence diagrams. *Ann. Statist.*, 42(6):2301–2339, 2014.

D. Freedman and P. Diaconis. On the histogram as a density estimator: $l_2$ theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57:453–476, 1981.

A. Graves. Generating sequences with recurrent neural networks. *CoRR*, 2013. https://arxiv.org/abs/1308.0850.

A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

W. Guo, K. Manohar, S.L. Brunton, and A.G. Banerjee. Sparse-TDA: Sparse realization of topological data analysis for multi-way classification. *IEEE Trans. Knowl. Data Eng.*, 2018.

A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. In *NIPS*, 2017a.

C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Constructing shape spaces from a topological perspective. In *IPMI*, 2017b.

G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

S. Kališnik Verovšek. Tropical coordinates on the space of persistent barcodes. *Found. Comput. Math.*, pages 1–29, 2018.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

G. Kusano, K. Fukumizu, and Y. Hiraoka. Persistence weighted Gaussian kernel for topological data analysis. In *ICML*, 2016.

R. Kwitt, S. Huber, M. Niethammer, W. Lin, and U. Bauer. Statistical topological data analysis - a kernel perspective. In *NIPS*, 2015.

L. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, 2000.

C. Li, M. Ovsjanikov, and F. Chazal. Persistence-based structural recognition. In *CVPR*, 2014.

Y. Mileyko, S. Mukherjee, and J. Harer. Probability measures on the space of persistence diagrams. *Inverse Probl.*, 27(12), 2011.

K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete Comput. Geom.*, 50(2):330–353, 2013.

A. Monod, S. Kališnik-Verovšek, J.A. Patiño Galindo, and L. Crawford. Tropical sufficient statistics for persistent homology. *CoRR*, 2017. https://arxiv.org/abs/1709.02647.

M. Nicolau, A.J. Levine, and G. Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *PNAS*, 108(17): 7265–7270, 2011.

M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016.

Pickup, D. et al. SHREC '14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR'14. Eurographics Association, 2014.

C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017a.

C.R. Qi, L. Yi, H. Su, and L.J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017b.

R. Reininghaus, U. Bauer, S. Huber, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *CVPR*, 2015.

B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

N. Shervashidze, P. Schweitzer, E.J. van Leeuwen, K. Mehlhorn, and K.M Borgwardt. Weisfeiler-lehmann graph kernels. *JMLR*, 12:2539–2561, 2011.

G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D.L. Ringach. Topological analysis of population activity in visual cortex. *J. Vis.*, 8(8), 2008.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 1st edition, 2008.

J. Sun, M. Ovsjanikov, and L. Guibas. A concise and probably informative multi-scale signature based on heat diffusion. In *SGP*, 2009.

K. Sun and B. Super. Classification of contour shapes using class segment sets. In *CVPR*, 2005.

I. Sutskever, O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

L. Takak and M. Zabovsky. Data analysis in public social networks. In *International Scientific Conference & International Workshop Present Day Trends of Innovations*, 2012.

K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet means for distributions of persistence diagrams. *Discrete Comput. Geom.*, 52(1):44–70, 2014a.

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

X. Wang, B. Feng, X. Bai, W. Liu, and L.J. Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognit.*, 47(6):2116–2125, 2014.

S. Welleck, K. Cho, and Z. Zhang. Saliency-based sequential image attention with multiset prediction. In *NIPS*, 2017.

P. Yanardag and S.V.N. Vishwanathan. Deep graph kernels. In *KDD*, 2015.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In *NIPS*, 2017.

A. Zomorodian and G. Carlsson. Computing persistent homology. In *SCG*, 2004.