

---

# Optimizing persistent homology based functions

---

Mathieu Carrière<sup>1</sup> Frédéric Chazal<sup>2</sup> Marc Glisse<sup>2</sup> Yuichi Ike<sup>3</sup> Hariprasad Kannan<sup>2</sup> Yuhei Umeda<sup>3</sup>

## Abstract

Solving optimization tasks based on functions and losses with a topological flavor is a very active, growing field of research in data science and Topological Data Analysis, with applications in non-convex optimization, statistics and machine learning. However, the approaches proposed in the literature are usually anchored to a specific application and/or topological construction, and do not come with theoretical guarantees. To address this issue, we study the differentiability of a general map associated with the most common topological construction, that is, the persistence map. Building on real analytic geometry arguments, we propose a general framework that allows us to define and compute gradients for persistence-based functions in a very simple way. We also provide a simple, explicit and sufficient condition for convergence of stochastic subgradient methods for such functions. This result encompasses all the constructions and applications of topological optimization in the literature. Finally, we provide associated code, that is easy to handle and to mix with other non-topological methods and constraints, as well as some experiments showcasing the versatility of our approach.

## 1. Introduction

Persistent homology is a central tool in Topological Data Analysis that allows to efficiently infer relevant topological features of complex data in a descriptor called persistence diagram. It has found many applications in Machine Learning (ML) where it initially played the role of a feature engineering tool, either through the direct use of persistence diagrams or through dedicated ML architectures that handle them—see, e.g., (Hofer et al., 2017; Umeda, 2017; Carrière et al., 2020; Dindin et al., 2020; Kim et al., 2020). For

<sup>1</sup>Université Côte d’Azur, Inria, France <sup>2</sup>Université Paris-Saclay, CNRS, Inria, Laboratoire de Mathématiques d’Orsay, France

<sup>3</sup>Fujitsu Ltd., Kanagawa, Japan. Correspondence to: Mathieu Carrière <[mathieu.carrriere@inria.fr](mailto:mathieu.carrriere@inria.fr)>.

the last few years, a growing number of works have successfully been using persistence theory in different ways, focusing on, for instance, better understanding, designing and improvement of neural network architectures—see, e.g., (Rieck et al., 2019; Moor et al., 2020; Carlsson & Gabrilsson, 2020; Gabrielson & Carlsson, 2019)—or design regularization and loss functions incorporating topological terms and penalties for various ML tasks—see, e.g., (Chen et al., 2019; Hofer et al., 2019; 2020; Clough et al., 2020). These new use cases of persistence generally involve minimizing functions that depend on persistence diagrams. Such functions are in general non-convex and not differentiable, and thus their theoretical and practical minimization can be difficult. In some specific cases, persistence-based functions can be designed to be differentiable and/or some effort have to be made to compute their gradient, so that standard gradient descent techniques can be used to minimize them—see e.g., (Wang et al., 2020; Poulenard et al., 2018; Brüel-Gabrielsson et al., 2020). In the general case, recent attempts have been made to better understand their differential structures (Leygonie et al., 2021). Moreover, building on powerful tools provided by software libraries such as PyTorch or TensorFlow, practical methods allowing to encode and optimize a large family of persistence-based functions have been proposed and experimented with (Brüel-Gabrielsson et al., 2020; Solomon et al., 2021). However, in all these cases, the algorithms used to minimize these functions do not come with theoretical guarantees of convergence to a global or local minimum.

**Contributions and organization of the article.** The aim of this article is to provide a general framework that includes almost all persistence-based functions from the literature, and for which stochastic subgradient descent algorithms are easy to implement and come with convergence guarantees.

More precisely, we first observe that the persistence map, converting a filtration over a given simplicial complex<sup>1</sup> into a persistence diagram, can be thought of as a map between Euclidean spaces (Section 2). This observation allows us to prove that the persistence map is semi-algebraic and, using classical arguments from o-minimal geometry, to study the

<sup>1</sup>The presentation is restricted to simplicial complexes for simplicity, but this generalizes to other complexes as well. We present an example with cubical complexes in Supplementary Material.

differentiability of the persistence of parametrized families of filtrations (Section 3). Then, building on the recent work of (Davis et al., 2020), we consider the minimization problem of persistence-based functions and show that under mild assumptions, stochastic subgradient descent algorithms applied to such functions converge almost surely to a critical point (Section 4). We also provide a simple corresponding Python implementation for minimizing functions of persistence<sup>2</sup>, and we illustrate it with several examples from the literature (Section 5).

## 2. Filtrations and persistence diagrams

In this section, we show that the persistence map is nothing but a permutation of the coordinates of a vector containing the filtration values.

### 2.1. Simplicial complexes and filtrations

Recall that given a set  $V$ , a (finite) *simplicial complex*  $K$  is a collection of finite subsets of  $V$  that satisfies (1)  $\{v\} \in K$  for any  $v \in V$ , and (2) if  $\sigma \in K$  and  $\tau \subseteq \sigma$  then  $\tau \in K$ . An element  $\sigma \in K$  with  $|\sigma| = k + 1$  is called a  $k$ -simplex.

Given a simplicial complex  $K$  and a subset  $R$  of  $\mathbb{R}$ , a *filtration* of  $K$  is an increasing sequence  $(K_r)_{r \in R}$  of subcomplexes of  $K$  with respect to the inclusion, i.e.,  $K_r \subseteq K_s$  for any  $r \leq s$ , and such that  $\bigcup_{r \in R} K_r = K$ .

To each simplex  $\sigma \in K$ , one can associate its *filtering index*  $\Phi_\sigma = \inf\{r \in R : \sigma \in K_r\}$ . Thus, when  $K$  is finite, a filtration of  $K$  can be conveniently represented as a filtering function  $\Phi: K \rightarrow \mathbb{R}$ . Equivalently, it can be represented as a  $|K|$ -dimensional vector  $\Phi = (\Phi_\sigma)_{\sigma \in K}$  in  $\mathbb{R}^{|K|}$  whose coordinates are the indices of the simplices of  $K$  and that satisfies the following condition: if  $\sigma, \tau \in K$  and  $\tau \subseteq \sigma$ , then  $\Phi_\tau \leq \Phi_\sigma$ . As a consequence, if the vectorized filtration  $\Phi$  depends on a parameter, the corresponding family of filtrations can be represented as a map from the space of parameters to  $\mathbb{R}^{|K|}$  in the following way.

**Definition 2.1.** Let  $K$  be a simplicial complex and  $A$  a set. A map  $\Phi: A \rightarrow \mathbb{R}^{|K|}$  is said to be a *parametrized family of filtrations* if for any  $x \in A$  and  $\sigma, \tau \in K$  with  $\tau \subseteq \sigma$ , one has  $\Phi_\tau(x) \leq \Phi_\sigma(x)$ .

### 2.2. Persistence computation from filtrations

We briefly recall how the computation of the persistence diagram of a filtered simplicial complex decomposes into: (i) a purely combinatorial part only relying on the order on the simplices induced by the filtration, and (ii) a part relying on the filtration values. A detailed introduction to persistent homology and its computation can be found in,

<sup>2</sup>It is publicly available at <https://github.com/MathieuCarriere/difftda>

e.g., (Edelsbrunner & Harer, 2010; Boissonnat et al., 2018).

Let  $K$  be a simplicial complex endowed with a filtration and corresponding filtering function  $\Phi \in \mathbb{R}^{|K|}$ , where  $|K|$  is the number of non-empty simplices of  $K$ .

**First part: combinatorial part (persistence pairs).** The filtering function  $\Phi$  induces a total preorder on the elements of  $K$  as follows:  $\tau \preceq \sigma$  if  $\Phi_\tau \leq \Phi_\sigma$ . This preorder can be refined into a total order by breaking ties in some fairly arbitrary way, as long as it is consistent with the face relation, i.e., if  $\tau \subseteq \sigma$ , then  $\tau \preceq \sigma$ . One way to break ties is to sort simplices that have the same filtration value by dimension, and then order the ones that are still equivalent according to some arbitrary indexing of the simplices. For instance, one can represent simplices by their decreasing list of vertices, and sort equivalent simplices using the lexicographic order on those lists. In the following, we will assume that the total order is a function of the preorder, in particular it is deterministic and does not depend on the exact values of  $\Phi$ . Note that while different orders may yield different pairings, they all translate to the same persistence diagram in the second part. The basic algorithm to compute persistence iterates over the ordered set of simplices  $\sigma_1 \preceq \dots \preceq \sigma_{|K|}$  as according to Algorithm 1 below—see Section 11.5.2 in (Boissonnat et al., 2018) for a detailed description of the algorithm.

---

#### Algorithm 1 Persistence pairs computation (sketch)

```

Input: Ordered sequence of simplices  $\sigma_1 \preceq \dots \preceq \sigma_{|K|}$ 
 $K_0 = \emptyset$ 
 $\text{Pairs}_0 = \text{Pairs}_1 = \dots = \text{Pairs}_{d-1} = \emptyset$ 
for  $j = 1$  to  $|K|$  do
     $k = \dim \sigma_j$ 
     $K_j = K_{j-1} \cup \sigma_j$ 
    if  $\sigma_j$  does not create a new  $k$ -dimensional homology
        class in  $K_j$  then
            A  $(k - 1)$ -dimensional homology class created in
             $K_{l(j)}$  by  $\sigma_{l(j)}$  for some  $l(j) < j$  becomes homolo-
            gous to 0 in  $K_j$ .
             $\text{Pairs}_{k-1} \leftarrow \text{Pairs}_{k-1} \cup \{(\sigma_{l(j)}, \sigma_j)\}$ ;
        end if
    end for
Output: Persistence pairs in each dimension
 $\text{Pairs}_0, \text{Pairs}_1, \dots, \text{Pairs}_{d-1}$ 

```

---

Note that for each dimension  $k$ , some  $k$ -dimensional simplices may remain unpaired at the end of the algorithm; their number is equal to the  $k$ -dimensional Betti number of  $K$ .

**Second part: associated filtration values.** The persistence diagram of the filter function  $\Phi$  is now obtained by associating to each persistent pair  $(\sigma_{l(j)}, \sigma_j)$  the point  $(\Phi_{\sigma_{l(j)}}, \Phi_{\sigma_j})$ . Moreover, to each unpaired simplex  $\sigma_l$  is

associated the point  $(\Phi_{\sigma_l}, +\infty)$ .

If  $p$  is the number of persistence pairs and  $q$  is the number of unpaired simplices, then  $|K| = 2p + q$  and the persistence diagram  $D(\Phi)$  of the filtration  $\Phi$  of  $K$  is made of  $p$  points in  $\mathbb{R}^2$  (counted with multiplicity) and  $q$  points (also counted with multiplicity) with infinite second coordinate. Choosing the lexicographical order on  $\mathbb{R} \times (\mathbb{R} \cup \{+\infty\})$ , the persistence diagram  $D(\Phi)$  can be represented as a vector in  $\mathbb{R}^{|K|}$  and the output of the persistence algorithm can be simply seen as a permutation of the coordinates of the input vector  $\Phi$ . Moreover, this permutation only depends on the order on the simplices of  $K$  induced by  $\Phi$ .

**Definition 2.2.** The subset of points of a persistence diagram  $D$  with finite coordinates (resp. infinite second coordinate) is called the *regular part* (resp. *essential part*) of  $D$  and denoted by  $D_{\text{reg}}$  (resp.  $D_{\text{ess}}$ ).

With the notations defined above,  $D_{\text{reg}}$  and  $D_{\text{ess}}$  can be represented as vectors in  $\mathbb{R}^{2p}$  and  $\mathbb{R}^q$ , respectively.

Note that, in practice, the above construction is usually done dimension by dimension to get a persistence diagram for each dimension in homology, by restricting to the subset of simplices of dimension  $k$  and  $k + 1$ . Without loss of generality, and to avoid unnecessary heavy notation, in the following we consider the whole persistence diagram, made of the union of the persistence diagrams in all dimensions  $k$ .

### 3. Differentiability of functions of persistence

$\text{o-minimal}$  geometry provides a well-suited setting to describe the parametrized families of filtrations encountered in practice and to exhibit interesting differentiability properties of their composition with the persistence map.

#### 3.1. Background on $\text{o-minimal}$ geometry

In this section, we recall some elements of  $\text{o-minimal}$  geometry, which are needed in the next sections—see, e.g., (Coste, 2000) for a more detailed introduction.

**Definition 3.1** ( $\text{o-minimal}$  structure). An  *$\text{o-minimal}$  structure* on the field of real numbers  $\mathbb{R}$  is a collection  $(S_n)_{n \in \mathbb{N}}$ , where each  $S_n$  is a set of subsets of  $\mathbb{R}^n$  such that:

1.  $S_1$  is exactly the collection of finite unions of points and intervals;
2. all algebraic subsets<sup>3</sup> of  $\mathbb{R}^n$  are in  $S_n$ ;
3.  $S_n$  is a Boolean subalgebra of  $\mathbb{R}^n$  for any  $n \in \mathbb{N}$ ;
4. if  $A \in S_n$  and  $B \in S_m$ , then  $A \times B \in S_{n+m}$ ;
5. if  $\pi: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  is the linear projection onto the first  $n$  coordinates and  $A \in S_{n+1}$ , then  $\pi(A) \in S_n$ .

<sup>3</sup>Recall that an algebraic set is the 0-level set of a polynomial.

An element  $A \in S_n$  for some  $n \in \mathbb{N}$  is called a *definable set* in the  $\text{o-minimal}$  structure. For a definable set  $A \subseteq \mathbb{R}^n$ , a map  $f: A \rightarrow \mathbb{R}^m$  is said to be *definable* if its graph is a definable set in  $\mathbb{R}^{n+m}$ .

Definable sets are stable under various geometric operations. The complement, closure and interior of a definable set are definable sets. The finite unions and intersections of definable sets are definable. The image of a definable set by a definable map is itself definable. Sums and products of definable functions as well as compositions of definable functions are definable—see Section 1.3 in (Coste, 2000). In particular, the max and min of finite sets of real-valued definable functions are also definable. An important property of definable sets and definable maps is that they admit a finite Whitney stratification (van den Dries & Miller, 1996). This implies that (i) any definable set  $A \subseteq \mathbb{R}^n$  can be decomposed into a finite disjoint union of smooth submanifolds of  $\mathbb{R}^n$  and (ii) for any definable map  $\Phi: A \rightarrow \mathbb{R}^m$ ,  $A$  can also be decomposed into a finite union of smooth manifolds such that the restriction of  $\Phi$  on each of these manifolds is a differentiable function.

The simplest example of  $\text{o-minimal}$  structures is given by the family of semi-algebraic subsets<sup>4</sup> of  $\mathbb{R}^n$  ( $n \in \mathbb{N}$ ). Although most of the classical parametrized families of filtrations are semi-algebraic, the  $\text{o-minimal}$  framework actually allows to consider larger families. In particular, the result of (Wilkie, 1996) says that the family of images of the sublevel sets of functions in  $\mathbb{R}[x_1, \dots, x_N, \exp(x_1), \dots, \exp(x_N)]$  for some  $N \in \mathbb{N}$  under linear projections is an  $\text{o-minimal}$  structure, which allows us to mix exponential functions with semi-algebraic functions.

#### 3.2. Persistence diagrams of definable parametrized families of filtrations

Let  $K$  be a simplicial complex and  $\Phi: A \rightarrow \mathbb{R}^{|K|}$  be a parametrized family of filtrations that is definable in a given  $\text{o-minimal}$  structure. If for any  $x, x' \in A$ , the preorders induced by  $\Phi(x)$  and  $\Phi(x')$  on the simplices of  $K$  are the same, i.e., for any  $\sigma_1, \sigma_2 \in K$ ,  $\Phi_{\sigma_1}(x) \leq \Phi_{\sigma_2}(x)$  if and only if  $\Phi_{\sigma_1}(x') \leq \Phi_{\sigma_2}(x')$ , then the pairs of simplices  $(\sigma_{i_1}, \sigma_{j_1}), \dots, (\sigma_{i_p}, \sigma_{j_p})$ , and the unpaired simplices  $\sigma_{i_{p+1}}, \dots, \sigma_{i_{p+q}}$  that are computed by the persistence Algorithm 1 are independent of  $x$ . Then,  $\forall x \in A$ , the persistence diagram  $D = D(\Phi(x))$  of  $\Phi(x)$  is

$$D = \bigcup_{k=1}^p (\Phi_{\sigma_{i_k}}(x), \Phi_{\sigma_{j_k}}(x)) \cup \bigcup_{k=1}^q (\Phi_{\sigma_{i_{p+k}}}(x), +\infty), \quad (1)$$

where  $|K| = 2p + q$ .

<sup>4</sup>It is the family of all finite unions and intersections of level sets and sublevel sets of polynomials (Benedetti & Risler, 1991).

Given the lexicographic order on  $\mathbb{R} \times (\mathbb{R} \cup \{+\infty\})$ , the points of any finite multi-set  $D \subseteq \mathbb{R} \times (\mathbb{R} \cup \{+\infty\})$  with  $p$  points in  $\mathbb{R}^2$  and  $q$  points in  $\mathbb{R} \times \{+\infty\}$  can be ordered in non-decreasing order, and  $D$  can be represented as a vector in  $\mathbb{R}^{2p+q}$ . As a consequence, denoting by  $\text{Filt}_K$  the set of vectors in  $\mathbb{R}^{|K|}$  that define a filtration on  $K$ , the persistence map  $\text{Pers}: \text{Filt}_K \rightarrow \mathbb{R}^{|K|}$  that assigns to each filtration of  $K$  its persistence diagram consists of a permutation of the coordinates of  $\mathbb{R}^{|K|}$ . This permutation is constant on the set of filtrations that define the same preorder on the simplices of  $K$ . This leads to the following statement.

**Proposition 3.2.** *Given a simplicial complex  $K$ , the map  $\text{Pers}: \text{Filt}_K \subseteq \mathbb{R}^{|K|} \rightarrow \mathbb{R}^{|K|}$  is semi-algebraic, and thus definable in any o-minimal structure. Moreover, there exists a semi-algebraic partition of  $\text{Filt}_K$  such that the restriction of  $\text{Pers}$  to each element of this partition is a Lipschitz map.*

*Proof.* See Supplementary Material.  $\square$

Since there exists a finite semi-algebraic partition of  $\text{Filt}_K$  on which  $\text{Pers}$  is a locally constant permutation, the subdifferential (see Section 4 for the definition) of  $\text{Pers}$  is well-defined and obvious to compute: each coordinate in the output (i.e., the persistence diagram) is a copy of a coordinate in the input (i.e., the filtration values of the simplices). This implies that every partial derivative is either 1 or 0. The output can be seen as a reindexing of the input, and this is indeed how we implement it in our code, so that automatic differentiation frameworks (PyTorch, TensorFlow, etc.) can process the function  $\text{Pers}$  directly and do not need explicit gradient formulas—see Section 5. Note that the subdifferential depends on the arbitrary refinement of the preorder in Subsection 2.2.

**Corollary 3.3.** *Let  $K$  be a simplicial complex and  $\Phi: A \rightarrow \mathbb{R}^{|K|}$  be a definable (in a given o-minimal structure) parametrized family of filtrations. The map  $\text{Pers} \circ \Phi: A \rightarrow \mathbb{R}^{|K|}$  is definable.*

Note that according to the remark following Proposition 3.2, if  $\Phi$  is differentiable, the subdifferential of  $\text{Pers} \circ \Phi$  can be easily computed in terms of the partial derivatives of  $\Phi$  using, for example, Equation (1).

It also follows from standard finiteness and stratifiability properties of definable sets and maps that  $\text{Pers} \circ \Phi$  is differentiable almost everywhere. More precisely:

**Proposition 3.4.** *Let  $K$  be a simplicial complex and  $\Phi: A \rightarrow \mathbb{R}^{|K|}$  a definable parametrized family of filtrations, where  $\dim A = m$ . Then there exists a finite definable partition of  $A$ ,  $A = S \sqcup O_1 \sqcup \dots \sqcup O_k$  such that  $\dim S < \dim A := m$  and, for any  $i = 1, \dots, k$ ,  $O_i$  is a definable manifold of dimension  $m$  and  $\text{Pers} \circ \Phi: O_i \rightarrow \mathcal{D}$  is differentiable.*

### 3.3. Examples of definable families of filtrations

**Vietoris-Rips filtrations.** The family of Vietoris-Rips filtrations built on top of sets of  $n$  points  $x_1, \dots, x_n \in \mathbb{R}^d$  is the semi-algebraic parametrized family of filtrations

$$\Phi: A = (\mathbb{R}^d)^n \rightarrow \mathbb{R}^{|\Delta_n|} = \mathbb{R}^{2^n - 1},$$

where  $\Delta_n$  is the simplicial complex made of all the faces of the  $(n - 1)$ -dimensional simplex. It is defined, for any  $x = (x_1, \dots, x_n) \in A$  and any simplex  $\sigma \subseteq \{1, \dots, n\}$ , by

$$\Phi_\sigma(x) = \max_{i, j \in \sigma} \|x_i - x_j\|.$$

One easily checks that the permutation induced by  $\text{Pers}$  is constant on the connected components of the complement of the union of the subspaces  $S_{i, j, k, l} = \{(x_1, \dots, x_n) : \|x_i - x_j\| = \|x_k - x_l\|\}$  over all the 4-tuples  $(i, j, k, l)$  such that at least 3 of the 4 indices  $i, j, k, l$  are distinct. This example naturally extends to Vietoris-Rips-like filtrations in the following way. Let  $A \subset \mathcal{M}_n(\mathbb{R})$  be the set of  $n \times n$  symmetric matrices with non-negative entries and 0 on the diagonal. This is a semi-algebraic subset of the space of  $n$ -by- $n$  matrices  $\mathcal{M}_n(\mathbb{R}) \simeq \mathbb{R}^{n^2}$ , of dimension  $m = (n - 1)(n - 2)/2$ . The map  $\Phi: A \rightarrow \mathbb{R}^{|\Delta_n|} = \mathbb{R}^{2^n}$  defined by  $\Phi_\sigma(M) = \max_{i, j \in \sigma} m_{i, j}$  for any  $M = (m_{i, j})_{1 \leq i, j \leq n} \in A$ , is a semi-algebraic family of filtrations. Note that the set  $S$  of Proposition 3.4 can be chosen to be the set of matrices with at least 2 entries that are equal.

**Weighted Rips filtrations.** Given a function  $f: \mathbb{R}^d \rightarrow \mathbb{R}$ , the family of weighted Rips filtrations  $\Phi: A = (\mathbb{R}^d)^n \rightarrow \mathbb{R}^{|\Delta_n|} = \mathbb{R}^{2^n}$  associated with  $f$  is defined, for any  $x = (x_1, \dots, x_n) \in A$  and any simplex  $\sigma \subseteq \{1, \dots, n\}$ , by

- $\Phi_\sigma(x) = 2f(x_j)$  if  $\sigma = [j]$ ;
- $\Phi_\sigma(x) = \max(2f(x_i), 2f(x_j), \|x_i - x_j\| + f(x_i) + f(x_j))$ , if  $\sigma = [i, j]$ ,  $i \neq j$ ;
- $\Phi_\sigma(x) = \max(\Phi_{[i, j]}(x), i, j \in \sigma)$  if  $|\sigma| \geq 3$ .

Since Euclidean distances and max function are semi-algebraic, this family of filtrations is definable as soon as the weight function  $f$  is definable.

This example easily extends to the case where the weight function depends on the set of points  $x = (x_1, \dots, x_n)$ : the weight at vertex  $y$  is defined by  $f(x, y)$  with  $f: (\mathbb{R}^d)^n \times \mathbb{R}^d \rightarrow \mathbb{R}$ . A particular example of such a family is given by the so-called DTM filtration (Anai et al., 2020), where  $f(x, y)$  is the average distance from  $y$  to its  $k$ -nearest neighbors in  $x$ . In this case,  $f$  is semi-algebraic, and the family of DTM filtrations is semi-algebraic.

The o-minimal framework also allows us to consider weight functions involving exponential functions (Wilkie, 1996), such as, for instance, kernel-based density estimates with Gaussian kernels.

**Sublevel sets filtrations.** Let  $K$  be a simplicial complex with  $n$  vertices  $v_1, \dots, v_n$ . Any real-valued function  $f$  defined on the vertices of  $K$  can be represented as a vector  $(f(v_1), \dots, f(v_n)) \in \mathbb{R}^n$ . The family of sublevel sets filtrations  $\Phi: A = \mathbb{R}^n \rightarrow \mathbb{R}^{|K|}$  of functions on the vertices of  $K$  is defined by  $\Phi_\sigma(f) = \max_{i \in \sigma} f_i$  for any  $f = (f_1, \dots, f_n) \in A$  and any simplex  $\sigma \subseteq \{1, \dots, n\}$ . This filtration is also known as the *lower-star filtration* of  $f$ . The function  $\Phi$  is obviously semi-algebraic, and for Proposition 3.4 to hold it is sufficient to choose  $S = \bigcup_{1 \leq i < j \leq n} \{f = (f_1, \dots, f_n) \in A : f_i = f_j\}$ .

## 4. Minimization of functions of persistence

Using the same notation as in the previous section, recall that the space of persistence diagrams associated with a filtration of  $K$  is identified with  $\mathbb{R}^{|K|} = (\mathbb{R}^2)^p \times \mathbb{R}^q$ , where each point in the  $p$  copies of  $\mathbb{R}^2$  is a point with finite coordinates in the persistence diagram and each coordinate in  $\mathbb{R}^q$  is the  $x$ -coordinate of a point with infinite persistence.

**Definition 4.1.** A function  $E: \mathbb{R}^{|K|} = (\mathbb{R}^2)^p \times \mathbb{R}^q \rightarrow \mathbb{R}$  is said to be a *function of persistence* if it is invariant to permutations of the points of the persistence diagram, i.e., for any  $(p_1, \dots, p_p, e_1, \dots, e_q) \in (\mathbb{R}^2)^p \times \mathbb{R}^q$  and any permutations  $\alpha, \beta$  of the sets  $\{1, \dots, p\}$  and  $\{1, \dots, q\}$ , respectively, one has

$$\begin{aligned} E(p_{\alpha(1)}, \dots, p_{\alpha(p)}, e_{\beta(1)}, \dots, e_{\beta(q)}) \\ = E(p_1, \dots, p_p, e_1, \dots, e_q). \end{aligned}$$

It follows from this permutation invariance and Proposition 3.2 that if a function of persistence  $E: \mathbb{R}^{2p+q} = \mathbb{R}^{|K|} \rightarrow \mathbb{R}$  is locally Lipschitz, then the composition  $E \circ \text{Pers}$  is also locally Lipschitz. Moreover, if  $E$  is definable in an o-minimal structure, then for any definable parametrized family of filtrations  $\Phi: A \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^{|K|}$ , the composition  $\mathcal{L} = E \circ \text{Pers} \circ \Phi: A \rightarrow \mathbb{R}$  is also definable. As a consequence,  $\mathcal{L}$  has a well-defined Clarke subdifferential  $\partial \mathcal{L}(z) := \text{Conv}\{\lim_{z_i \rightarrow z} \nabla \mathcal{L}(z_i) : \mathcal{L} \text{ is differentiable at } z_i\}$ , since it is differentiable almost everywhere thanks to Proposition 3.4.

### 4.1. Stochastic gradient descent

To minimize  $\mathcal{L}$ , we consider the differential inclusion

$$\frac{dz}{dt} \in -\partial \mathcal{L}(z(t)) \quad \text{for almost every } t,$$

whose solutions  $z(t)$  are the trajectories of the subgradient of  $\mathcal{L}$ . They can be approximated by the standard stochastic

subgradient algorithm given by the iterations of

$$x_{k+1} = x_k - \alpha_k(y_k + \zeta_k), \quad y_k \in \partial \mathcal{L}(x_k), \quad (2)$$

where the sequence  $(\alpha_k)_k$  is the learning rate and  $(\zeta_k)_k$  is a sequence of random variables. In (Davis et al., 2020), the authors prove that under mild technical conditions on these two sequences, the stochastic subgradient algorithm converges almost surely to a critical point of  $\mathcal{L}$  as soon as  $\mathcal{L}$  is locally Lipschitz.

More precisely, consider the following assumptions, which correspond to Assumption C in (Davis et al., 2020):

1. for any  $k$ ,  $\alpha_k \geq 0$ ,  $\sum_{k=1}^{\infty} \alpha_k = +\infty$  and,  $\sum_{k=1}^{\infty} \alpha_k^2 < +\infty$ ;
2.  $\sup_k \|x_k\| < +\infty$ , almost surely;
3. denoting by  $\mathcal{F}_k$  the increasing sequence of  $\sigma$ -algebras  $\mathcal{F}_k = \sigma(x_j, y_j, \zeta_j, j < k)$ , there exists a function  $p: \mathbb{R}^d \rightarrow \mathbb{R}$  which is bounded on bounded sets such that almost surely, for any  $k$ ,

$$\mathbb{E}[\zeta_k | \mathcal{F}_k] = 0 \text{ and } \mathbb{E}[\|\zeta_k\|^2 | \mathcal{F}_k] < p(x_k).$$

These assumptions are standard and not very restrictive. Assumption 1 depends on the choice of the learning rate by the user and is easily satisfied, e.g., taking  $\alpha_k = 1/k$ . Assumption 2 is usually easy to check for most of the functions  $\mathcal{L}$  encountered in practice. Assumption 3 is a standard condition, which states that, conditioned upon the past, the variables  $\zeta_k$  have zero mean and controlled moments; e.g., this can be achieved by taking a sequence of independent and centered variables with bounded variance that are also independent of the  $x_k$ 's and  $y_k$ 's.

Under these assumptions, the following result is an immediate consequence of Corollary 5.9 in (Davis et al., 2020).

**Theorem 4.2.** Let  $K$  be a simplicial complex,  $A \subseteq \mathbb{R}^d$ , and  $\Phi: A \rightarrow \mathbb{R}^{|K|}$  a parametrized family of filtrations of  $K$  that is definable in an o-minimal structure. Let  $E: \mathbb{R}^{|K|} \rightarrow \mathbb{R}$  be a definable function of persistence such that  $\mathcal{L} = E \circ \text{Pers} \circ \Phi$  is locally Lipschitz. Then, under the above assumptions 1, 2, and 3, almost surely the limit points of the sequence  $(x_k)_k$  obtained from the iterations of Equation (2) are critical points of  $\mathcal{L}$  and the sequence  $(\mathcal{L}(x_k))_k$  converges.

The above theorem provides explicit conditions ensuring the convergence of stochastic subgradient descent for functions of persistence. The main criterion to be checked is the locally Lipschitz condition for  $\mathcal{L}$ . From the remark following Definition 4.1, it is sufficient to check that  $\Phi$  and  $E$  are Lipschitz. Regarding  $\Phi$ , it is obvious for the examples of Subsection 3.3. However, this is not the case for some other examples, such as the so-called alpha-complex filtration that can be made locally Lipschitz using a simple technical trick—see Supplementary Material.

#### 4.2. Examples of definable locally Lipschitz functions of persistence

**Total persistence.** Let  $E$  be the sum of the distances to the diagonal of each point of a persistence diagram with finite coordinates: given a persistence diagram represented as a vector in  $\mathbb{R}^{2p+q}$ ,  $D = ((b_1, d_1), \dots, (b_p, d_p), e_1, \dots, e_q)$ ,

$$E(D) = \sum_{i=1}^p |d_i - b_i|.$$

Then  $E$  is obviously semi-algebraic, and thus definable in any o-minimal structure. It is also Lipschitz.

**Wasserstein and bottleneck distance** Given a persistence diagram  $D$ , and another target persistence diagram  $D^*$ , the bottleneck distance between the regular part of  $D$  and that of  $D^*$  (see Definition 2.2) is given by

$$E(D) = d_B(D_{\text{reg}}, D_{\text{reg}}^*) = \min_m \max_{(p, p^*) \in m} \|p - p^*\|_\infty,$$

where  $m$  is a partial matching between  $D_{\text{reg}}$  and  $D_{\text{reg}}^*$ , i.e., a subset of  $(D_{\text{reg}} \cup \Delta) \times (D_{\text{reg}}^* \cup \Delta)$ , with  $\Delta = \{(x, x) : x \in \mathbb{R}\}$  being the diagonal in  $\mathbb{R}^2$ , such that every point of  $D_{\text{reg}} \setminus \Delta$  and  $D_{\text{reg}}^* \setminus \Delta$  appears exactly once in  $m$ . One can easily check that the map  $E$  is semi-algebraic, and thus definable in any o-minimal structure. It is also Lipschitz. This property also extends to the case where the bottleneck distance is replaced by the so-called Wasserstein distance  $W_p$  with  $p \in \mathbb{N}$  (Cohen-Steiner et al., 2010), or its approximation, the Sliced Wasserstein distance (Carrière et al., 2017). Optimization of these functions and other functions of bottleneck and Wasserstein distances have been used, for example, in shape matching (Poulenard et al., 2018). See also the example on 3D shape in Supplementary Material.

**Persistence landscapes (Bubenik, 2015)** To any given point  $p = (x, y) \in \mathbb{R}^2$  with  $x = \frac{b+d}{2}$  and  $y = \frac{d-b}{2}$ , associate the function  $\Lambda_p : \mathbb{R} \rightarrow \mathbb{R}$  defined by

$$\Lambda_p(t) = \begin{cases} t - b & (t \in [b, \frac{b+d}{2}]) \\ d - t & (t \in (\frac{b+d}{2}, d]) \\ 0 & (\text{otherwise}). \end{cases}$$

Given a persistence diagram  $D$ , the persistence landscape of  $D$  is a summary of the arrangement of the graphs of the functions  $\Lambda_p$ ,  $p \in D$ :

$$\lambda_D(k, t) = k\text{-}\max_{p \in D} \Lambda_p(t), \quad t \in [0, T], \quad k \in \mathbb{Z}^+,$$

where  $k$ -max is the  $k$ th largest value in the set, or 0 when the set contains less than  $k$  points. Given a positive integer  $k$ , a finite set  $\{t_1, \dots, t_n\} \subset \mathbb{R}$ , and a simplicial complex  $K$ , the map that associates the vector

$(\lambda_D(k, t_1), \dots, \lambda_D(k, t_n))$  to each persistence diagram  $D$  of a filtration of  $K$  is Lipschitz (Bubenik, 2015) and clearly semi-algebraic.

Other classical ways to vectorize persistence diagrams are the *linear representations* (Chazal & Divol, 2018) which are also definable in o-minimal structures, such as, e.g., persistence images (Adams et al., 2017)—see Supplementary Material. In (Divol & Lacombe, 2020), the authors give explicit conditions for such representations to be locally Lipschitz.

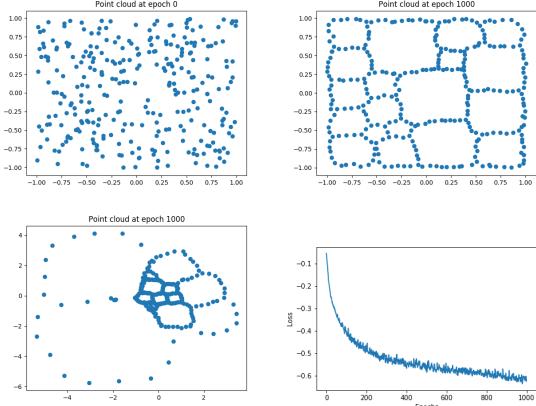
#### 5. Numerical illustrations

We showed in Sections 3 and 4 that the usual stochastic gradient descent procedure of Equation (2) enjoys some convergence properties for persistence-based functions. This means in particular that the algorithms available in standard libraries such as TensorFlow and PyTorch, which implement stochastic gradient descent (among other optimization methods), can be leveraged and used as is for differentiating persistence diagrams, while still ensuring convergence. The purpose of this section is to illustrate that our code, which implements the general gradient defined in Proposition 3.4 for persistence-based functions, and which is based on Gudhi<sup>5</sup> and TensorFlow, can be readily used for studying several different persistence optimization tasks. Along the way, we also suggest regularization terms that one can add to topological losses in order to avoid unwanted behaviors. We only present a few applications due to lack of space, and we refer the interested reader to Supplementary Material and the publicly available code for more examples.

**Point cloud optimization.** A toy example in persistence optimization is to modify the positions of the points in a point cloud so that its homology is maximized (Brüel-Gabrielsson et al., 2020; Gameiro et al., 2016). In this experiment, we start with a point cloud  $X$  sampled uniformly from the unit square  $S = [0, 1]^2$ , and then optimize the point coordinates so that the loss  $\mathcal{L}(X) = P(X) + T(X)$  is minimized. Here  $T(X) := -\sum_{p \in D} \|p - \pi_\Delta(p)\|_\infty^2$  is a topological penalty,  $D$  is the 1-dimensional persistence diagram associated with the Vietoris-Rips filtration of  $X$ ,  $\pi_\Delta$  stands for the projection onto the diagonal  $\Delta$ , and  $P(X) := \sum_{x \in X} d(x, S)$  is a penalty term ensuring that the point coordinates stay in the unit square. The topological penalty  $T(X)$  was used in (Brüel-Gabrielsson et al., 2020), and ensures that points in the persistence diagram  $D$  are as far away from the diagonal as possible, which in turns means that the corresponding holes in the point cloud are as large as possible. However, we point out that if one uses  $T(X)$  alone without the penalty  $P(X)$ , as in (Brüel-Gabrielsson et al., 2020), then convergence is very difficult to reach since

<sup>5</sup>See <https://gudhi.inria.fr/>

inflating the point cloud with dilations can make the topological penalty  $T(X)$  arbitrarily small. In contrast, using our second term  $P(X)$  in addition to  $T(X)$  constrains the points to stay in a fixed region  $S$  of the Euclidean plane. Another effect of the penalty  $P(X)$  is to flatten the boundary of the created holes along the boundary of  $S$ . See Figure 1 for an illustration.



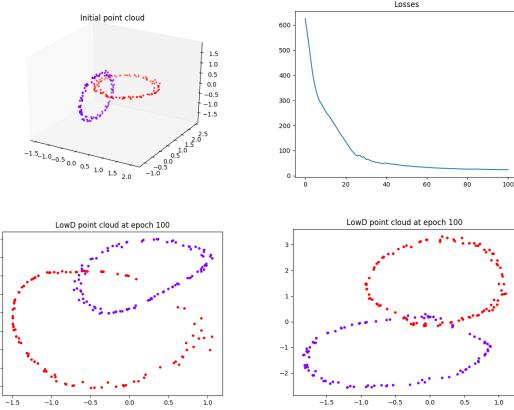
**Figure 1.** Illustration of point cloud optimization. We initialize with a random point cloud (upper left), and we show the optimized point cloud (upper right) when optimization is done with topological and regularization losses. We also show the convergence of the total loss (lower right). When only topological loss is used, the optimized point cloud inflated some loops to minimize the loss (lower left). Note how the coordinates are now much larger.

**Dimensionality reduction.** In this experiment, we show how our general setup can be used to reduce dimension with the so-called *topological autoencoders* introduced in (Moor et al., 2020). In this family of autoencoders, a topological loss  $T(X, Z)$  between the input space  $X$  and latent space  $Z$  is used in addition to the usual loss  $\mathcal{D}(X, Z) = \sum_i \|x_i - z_i\|_2^2$ . This topological loss was computed in (Moor et al., 2020) by (i) computing the permutations induced by the persistence map (see Subsection 3.2) of the Vietoris-Rips complexes built from the input space  $X$  and the latent space  $Z$ , (ii) computing, for each simplex in these permutations, the corresponding edge that induces its filtration value, and (iii) measuring, for all those edges, the differences between the edge lengths in  $X$  and the same edge lengths in  $Z$ . To sum up, the loss function is defined as

$$\begin{aligned} \mathcal{L}(X, Z) = & \|M_X[\pi_X] - M_Z[\pi_X]\|_2^2 \\ & + \|M_X[\pi_Z] - M_Z[\pi_Z]\|_2^2, \end{aligned}$$

where  $M_X, M_Z$  are the distance matrices of the input and the latent spaces respectively, and where  $\pi_X, \pi_Z$  denote the indices of the entries in  $M_X, M_Z$  that are picked by

the permutation induced by the persistence map to generate the Vietoris-Rips persistence diagrams of  $X$  and  $Z$ . Note that  $\mathcal{L}$  is obviously semi-algebraic and thus fits in our framework. Moreover, in our setup we can directly use the bottleneck and Wasserstein distances between the Vietoris-Rips persistence diagrams of the input and latent spaces as the topological loss. This is relevant since in (Moor et al., 2020) the authors pointed out that looking at homology in dimension larger than 1 was not adding anything for their loss, and stuck to 0-dimensional homology. We show in Figure 2 an example in which 1-dimensional homology is also important, that is, a point cloud in  $\mathbb{R}^3$  that is comprised of two nested circles, which is then non-linearly embedded in  $\mathbb{R}^9$  by converting each point  $p = (x, y, z)$  into the exponential of the  $3 \times 3$  anti-symmetric matrix whose coefficients are  $x, y$  and  $z$ . We then train an autoencoder made of four fully-connected layers with 32 neurons and ReLU activations, using the usual loss, the usual plus the topological loss described above, and the usual plus a topological loss computed as  $\mathcal{L}(X, Z) = W_1(D_X, D_Z)$ , i.e., the 1-Wasserstein distance between the 1-dimensional Vietoris-Rips persistence diagrams of the input and latent spaces. It can be seen from Figure 2 that autoencoders without the Wasserstein loss cannot embed the point cloud in the plane perfectly, while using the Wasserstein loss between the 1-dimensional Vietoris-Rips persistence diagrams of the input and latent spaces improves on the result by separating better the two intrinsic circles.



**Figure 2.** Example of dimension reduction with autoencoders. An initial point cloud made of two circles (upper left) is embedded in  $\mathbb{R}^9$ , and then fed to autoencoders that either do not use topology, or only use the distances induced by the persistence maps in dimension 0. The resulting embeddings (lower left, we only show one but the two are similar) cannot separate the circles, while using 1-dimensional topology induces a better embedding (lower right). Convergence of the loss function is also provided (upper right).

**Filter selection.** In this experiment, we address a very common issue in Topological Data Analysis, filter selection. Indeed, when computing persistence diagrams in order to

### Optimizing persistent homology based functions

Dataset	Baseline	Before	After	Difference	Dataset	Baseline	Before	After	Difference
vs01	100.0	61.3	99.0	<b>+37.6</b>	vs26	99.7	98.8	98.2	-0.6
vs02	99.4	98.8	97.2	-1.6	vs28	99.1	96.8	96.8	0.0
vs06	99.4	87.3	98.2	<b>+10.9</b>	vs29	99.1	91.6	98.6	<b>+7.0</b>
vs09	99.4	86.8	98.3	<b>+11.5</b>	vs34	99.8	99.4	99.1	-0.3
vs16	99.7	89.0	97.3	<b>+8.3</b>	vs36	99.7	99.3	99.3	-0.1
vs19	99.6	84.8	98.0	<b>+13.2</b>	vs37	98.9	94.9	97.5	<b>+2.6</b>
vs24	99.4	98.7	98.7	0.0	vs57	99.7	90.5	97.2	<b>+6.7</b>
vs25	99.4	80.6	97.2	<b>+16.6</b>	vs79	99.1	85.3	96.9	<b>+11.5</b>

**Table 1.** Accuracy scores obtained from persistence diagrams before and after performing our optimization over the image filtration. Note that the difference between the scores is almost always positive, i.e., there is almost always improvement after our optimization process. Scores do not have standard deviations since we use the train/test splits of the `mnist.load_data` function in TensorFlow 2.

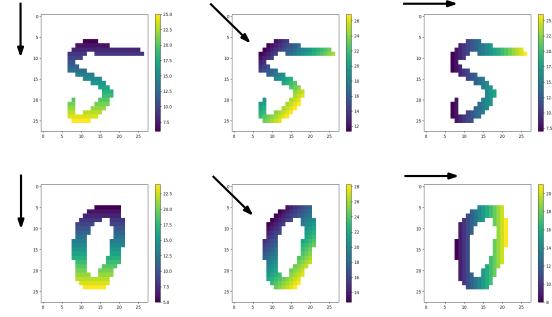
generate topological features from a data set for further data analysis, the filter function that is being used to filter the data set always has to be specified a priori. Here, we provide a very simple heuristic to tune it if it comes from a parametrized family  $\mathcal{F}$  of filters and if the learning task is supervised, which is the case in, e.g., classification. We simply start from a random guess in  $\mathcal{F}$  and then optimize the following criterion, inspired from (Zhao & Wang, 2019):

$$\mathcal{L}(f) = \sum_{l=1}^N \frac{\sum_{i,j:y_i=y_j=l} W_p(D_i(f), D_j(f))}{\sum_{i,j:y_i=l} W_p(D_i(f), D_j(f))}, \quad (3)$$

which amounts to minimizing the distances between persistence diagrams that share the same label, and increasing the distances between persistence diagrams with different labels. Note that the batch size that we use in this optimization process has a big influence on the computation time, since the larger the batch size, the more Wasserstein distances we will have to compute in our cost. To cope with this issue, we actually used the Sliced Wasserstein distance  $SW$  (Carrière et al., 2017) instead of  $W_p$ , which, since it is computed with projections onto lines, can be defined entirely with matrix operations that are usually available in any library with autodifferentiation. This drastically improves on computation time, while remaining in our framework since the Sliced Wasserstein distance is also a semi-algebraic function.

We classify images from the MNIST data set. We assign values to the pixels using a height function given by a direction (parametrized by an angle in the Euclidean plane), and we use 0-dimensional persistence diagrams computed after optimizing this direction using loss (3). See Figure 3.

We then compute the accuracy scores obtained with a random forest classifier for the (binary) classification tasks digit  $x$  vs. digit  $y$  for all pairs  $0 \leq x, y \leq 9$ , using the first five persistence landscapes with resolution 100 associated with the persistence diagrams before and after optimization. Even though our primary goal is to demonstrate that optimizing the filter almost always lead to an improvement, we also add a baseline score obtained by training a random



**Figure 3.** Example of images and directions inducing different height functions. Different directions generate different height functions and filtrations and thus different persistence diagrams. In this experiment, we optimize over the direction so that the persistence diagrams are the most efficient for image classification.

forest classifier directly on the images for proper comparison. Some of the scores are displayed in Table 1 (the full table can be found in Supplementary Material). Interestingly, when starting with a random direction, scores can be much worse than the baseline, but our optimization process is then able to select the best direction that induces the best persistence diagrams (with respect to the classification task) without prior knowledge on the data set.

## 6. Conclusion

In this article we introduced a theoretical framework that encompasses most of the previous methods for optimizing topology-based functions. In particular, we obtained convergence results for very general classes of functions with topological flavor computed with persistence theory, and provided corresponding code that one can use to reproduce previously introduced topological optimization tasks. For future work, we are planning to further investigate tasks related to classifier regularization in ML (Chen et al., 2019), and to improve on computation time using, e.g., vineyards (Cohen-Steiner et al., 2006).

## References

- Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(1):218–252, 2017.
- Anai, H., Chazal, F., Glisse, M., Ike, Y., Inakoshi, H., Tinarrage, R., and Umeda, Y. DTM-based Filtrations. In *Topological Data Analysis*, pp. 33–66. Springer, 2020.
- Benedetti, R. and Risler, J.-J. *Real algebraic and semialgebraic sets*. Hermann, 1991.
- Boissonnat, J.-D., Chazal, F., and Yvinec, M. *Geometric and topological inference*, volume 57. Cambridge University Press, 2018.
- Brüel-Gabrielsson, R., Ganapathi-Subramanian, V., Skraba, P., and Guibas, L. J. Topology-aware surface reconstruction for point clouds. In *Computer Graphics Forum*, volume 39, pp. 197–207. Wiley Online Library, 2020.
- Brüel-Gabrielsson, R., Nelson, B., Dwaraknath, A., Skraba, P., Guibas, L., and Carlsson, G. A topology layer for machine learning. In *23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, pp. 1553–1563. PMLR, 2020.
- Bubenik, P. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.
- Carlsson, G. and Gabrielsson, R. B. Topological approaches to deep learning. In *Topological Data Analysis*, pp. 119–146. Springer, 2020.
- Carrière, M., Cuturi, M., and Oudot, S. Sliced Wasserstein kernel for persistence diagrams. In *34th International Conference on Machine Learning (ICML 2017)*, volume 70, pp. 664–673. JMLR.org, 2017.
- Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., and Umeda, Y. PersLay: a neural network layer for persistence diagrams and new graph topological signatures. In *23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, pp. 2786–2796. PMLR, 2020.
- Chazal, F. and Divol, V. The density of expected persistence diagrams and its kernel based estimation. In *34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- Chen, C., Ni, X., Bai, Q., and Wang, Y. A topological regularizer for classifiers via persistent homology. In *22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)*, volume 89, pp. 2573–2582. PMLR, 2019.
- Clough, J., Byrne, N., Oksuz, I., Zimmer, V. A., Schnabel, J. A., and King, A. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. Vines and vineyards by updating persistence in linear time. In Amenta, N. and Cheong, O. (eds.), *22nd Annual Symposium on Computational Geometry (SoCG 2006)*, pp. 119–126. Association for Computing Machinery, 2006.
- Cohen-Steiner, D., Edelsbrunner, H., Harer, J., and Mileyko, Y. Lipschitz functions have  $L_p$ -stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010.
- Coste, M. *An introduction to o-minimal geometry*. Istituti editoriali e poligrafici internazionali Pisa, 2000.
- Davis, D., Drusvyatskiy, D., Kakade, S. M., and Lee, J. D. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, 20(1): 119–154, 2020.
- Dindin, M., Umeda, Y., and Chazal, F. Topological data analysis for arrhythmia detection through modular neural networks. In *Canadian Conference on Artificial Intelligence*, pp. 177–188. Springer, 2020.
- Divol, V. and Lacombe, T. Understanding the topology and the geometry of the persistence diagram space via optimal partial transport. *Journal of Applied and Computational Topology*, 5:1–53, 2020.
- Edelsbrunner, H. and Harer, J. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- Gabrielsson, R. B. and Carlsson, G. Exposition and interpretation of the topology of neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1069–1076. IEEE, 2019.
- Gameiro, M., Hiraoka, Y., and Obayashi, I. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334:118–132, 2016.
- Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 1634–1644. Curran Associates, Inc., 2017.
- Hofer, C., Kwitt, R., Niethammer, M., and Dixit, M. Connectivity-optimized representation learning via persistent homology. In *36th International Conference on*

- Machine Learning (ICML 2019)*, pp. 2751–2760. PMLR, 2019.
- Hofer, C., Graf, F., Niethammer, M., and Kwitt, R. Topologically densified distributions. In *37th International Conference on Machine Learning (ICML 2020)*, volume 119, pp. 4304–4313. PMLR, 2020.
- Kim, K., Kim, J., Zaheer, M., Kim, J., Chazal, F., and Wasserman, L. Efficient topological layer based on persistent landscapes. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.
- Leygonie, J., Oudot, S., and Tillmann, U. A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics (to appear)*, 2021.
- Moor, M., Horn, M., Rieck, B., and Borgwardt, K. Topological autoencoders. In *37th International Conference on Machine Learning (ICML 2020)*, volume 119, pp. 7045–7054, 2020.
- Poulenard, A., Skraba, P., and Ovsjanikov, M. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, volume 37, pp. 13–25. Wiley Online Library, 2018.
- Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumsch, T., and Borgwardt, K. Neural persistence: a complexity measure for deep neural networks using algebraic topology. In *7th International Conference on Learning Representations (ICLR 2019)*. OpenReviews.net, 2019.
- Solomon, Y., Wagner, A., and Bendich, P. A fast and robust method for global topological functional optimization. In *24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021)*, volume 130, pp. 109–117. PMLR, 2021.
- Umeda, Y. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.
- van den Dries, L. and Miller, C. Geometric categories and o-minimal structures. *Duke Mathematical Journal*, 84(2):497–540, 1996.
- Wang, F., Liu, H., Samaras, D., and Chen, C. Topogan: A topology-aware generative adversarial network. In *European Conference on Computer Vision (ECCV)*, 2020.
- Wilkie, A. J. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996.
- Zhao, Q. and Wang, Y. Learning metrics for persistence-based summaries and applications for graph classification. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pp. 9855–9866. Curran Associates, Inc., 2019.