# Mixup Barcodes: Quantifying Geometric-Topological Interactions Between Point Clouds

**Hubert Wagner**
University of Florida, Gainesville, FL, USA

**Nickolas Arustamyan**
University of Central Florida, Orlando, FL, USA

**Matthew Wheeler**
University of Florida, Gainesville, FL, USA

**Peter Bubenik**
University of Florida, Gainesville, FL, USA

## Abstract

We propose a novel geometric-topological descriptor called a mixup barcode. Intuitively, it characterizes the shape of a point cloud as well as its spatial relationship with another point cloud embedded in the same ambient space. More technically, it enriches a standard persistence barcode with information on the image persistent homology. In three dimensions it captures natural spatial relationships like overlap and surrounding; in higher dimensions more intricate spatial relationships are captured. We provide a theoretical setup and a simple algorithm for mixup barcodes.

As a proof of concept, we explore data arising in a geometric-topological problem from machine learning. Specifically, we take first steps towards verifying a hypothesis stating that geometric-topological relationships within intermediate point cloud representations in an artificial neural network can hinder its training. More broadly, our experiments suggest that mixup barcodes are useful for characterizing spatial relationships and spatial interactions (i.e. the evolution of spatial relationships) that are hard to directly visualize or capture using standard methods.

**2012 ACM Subject Classification** Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

**Keywords and phrases** mixup barcode, persistent homology, persistence barcode, persistence diagram, image persistent homology, image persistence, deep learning, multilayer perceptron, topology of neural network embeddings, disentanglement

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2026.94

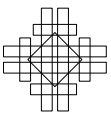
**Related Version** *Full Version*: `https://doi.org/10.48550/arXiv.2402.15058`

**Supplementary Material** *Software*: `https://github.com/hubwag/Mixup-SoCG26` [36]
archived at `swh:1:dir:9b75232380440dc7556b698dd1b7d07882a61fac`

**Funding** *Hubert Wagner*: Partially supported by the 2022 Google Research Scholar Award in Algorithms and Optimization.
*Matthew Wheeler*: Supported by The American Association of Immunologists through an Intersect Fellowship for Computational Scientists and Immunologists.

**Acknowledgements** The authors thank Ondřej Draganov, Teresa Heiss, Herbert Edelsbrunner and Yasu Hiraoka for helpful discussions.

© Hubert Wagner, Nickolas Arustamyan, Matthew Wheeler, and Peter Bubenik;
licensed under Creative Commons License CC-BY 4.0

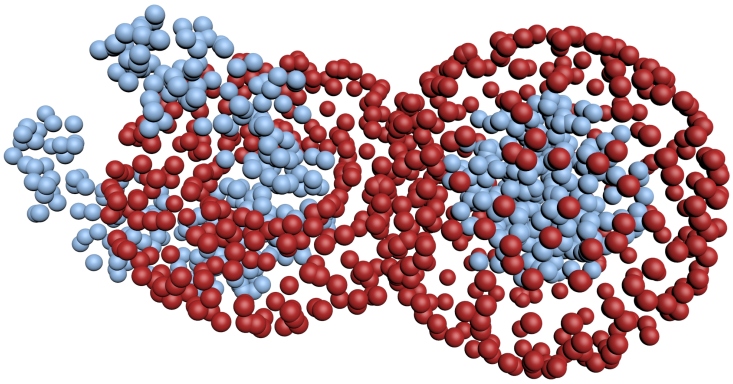


**Figure 1** TDA allows us to characterize the shapes of the two, red and blue, point clouds. We also ask: what are the spatial relationships between these point clouds? Intuitively, a portion of the blue point cloud is surrounded by the spherical part of the red one; similarly, another portion of the blue point cloud loops around the toroidal subset of the red points. These are natural questions and we currently lack the tools (and in some cases the language) to discuss them. High-dimensional generalizations play an important role in the machine learning problem we consider.

## 1 Introduction

Computational geometry delivers a variety of tools for characterizing the shape of a point cloud. One can compute the convex hull, the Delaunay triangulation and its variants, and related filtrations such as alpha filtrations. Analogs for high dimensional point clouds such as Rips and Čech filtrations are becoming better understood [20] and more computable [4, 14].

**The shape of data.** In recent years, it has become popular to summarize the geometric-topological structure of a point cloud using persistent homology. It involves computing a persistence diagram, or barcode, serving as a geometric-topological descriptor of a dataset.

**Beyond the shape of data.** There are important problems which require characterizing the **spatial relationships** among datasets. We provide a new tool, called a **mixup barcode**, which captures both the shape of data and its spatial relationships with another dataset embedded in the same ambient space. See Figure 1 for an example. We will also consider spatial interactions, by which we mean the evolution of spatial relationships over time.

In overview, mixup barcodes capture how the inclusion of another point cloud affects the persistent homology of an existing point cloud. Specifically, we ask how the lifetime of each topological feature shortens in the presence of new data. This way we quantify the spatial relationships between the two point clouds. While this task fits the general framework of multi-parameter persistent homology, we take a more pragmatic and computable approach. We frame this intuition in terms of persistent homology as well its variant, image persistent homology, which was recently shown to be efficiently computable [7]. We carefully combine these two pieces of information.

**A proof of concept application.** Our method is motivated by a problem arising in machine learning. We will refer to this problem as *the disentanglement problem*. In [29], it was hypothesized that training can be hindered by geometric-topological interactions within the intermediate representations (point clouds) produced by such models. Intuitively, a

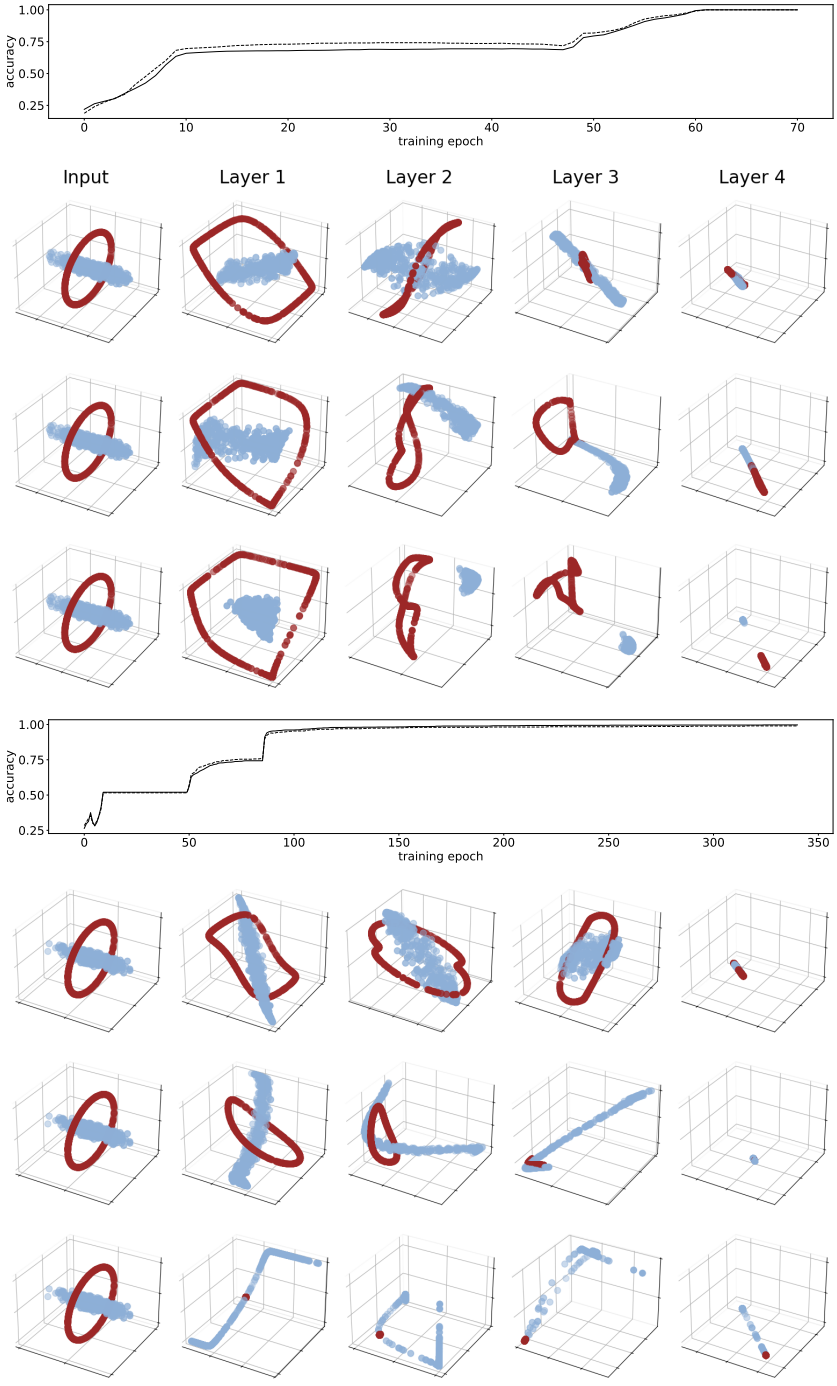



**Figure 2** Two training runs of the same MLP classifier. For each run, three selected training epochs are shown as rows. Each row shows the outputs of consecutive layers from left to right. Intuitively, at the end of the training, the blue and red points output by the *last* layer should be separated. In both cases (almost) perfect test accuracy is achieved. However, in the second run the model failed to *disentangle* the red and blue points, and a small portion of the blue points became *confused* with the red points. In particular, this behavior lowers the generalization power of the model (since certain new inputs are bound to be misclassified). We propose tools for detecting such situations.

model may fail to **disentangle** the representations corresponding to data with different labels – see Figure 2 for a concrete example. First steps towards testing this hypothesis were recently made using persistent homology [27, 39]. We argue that our method offers additional, useful information, and present experiments supporting this claim. We elaborate on the experimental setup in Section 7.

**Contributions.** The contributions of this paper include:

1. Mathematical foundations for a new topological descriptor called a *mixup barcode* characterizing spatial relationships between point clouds.
2. Summary statistics quantifying the robustness of these relationships.
3. Experiments showing the computability and usefulness of mixup barcodes in the above-mentioned machine-learning problem.
4. A simple practical algorithm for computing mixup barcodes.
5. This is one of the first applications of image persistence [13] and the first application we are aware of for high-dimensional data.

**Connections and potential impacts.** We expect this work to expand the standard paradigm of topological data analysis, namely from summarizing and learning from the shape of a point cloud, to summarizing and learning from the spatial relationships among point clouds. This goal is aligned with the emerging direction of Chromatic Topological Data Analysis outlined in [16]. Our work was developed independently, but both directions build on the same theoretical and algorithmic foundation [13]. The main novelty of our approach comes from providing a single, compound descriptor that merges two simpler barcodes. These barcodes are typically considered in isolation – which loses important information that our method retains (at a negligible computational cost). We will contrast these approaches further in Section 8.

Once the theoretical and computational aspects are fully developed, the proposed tool will likely open new applications of geometric-topological methods.

## 2 Related work

We briefly review related work in computational geometry and topology, as well as their relevant applications in machine learning.

**Image persistence.** Regarding computational methods for image persistence, an algorithm was proposed by Cohen-Steiner and collaborators [13]. An implementation is available in the first version of Morozov's Dionysus library, but not in the current version. Recently, Bauer and Schmahl proposed a version [7] of the Ripser library for image persistence of Vietoris–Rips complexes, which informs our implementation. One of the first practical applications of image persistence was done recently in the context of image segmentation [33].

Cultrera di Montesano and collaborators [17] have proposed an approach for analyzing the mingling of a small number of 3-dimensional point clouds using image (as well as kernel and cokernel) persistence. One crucial development is the chromatic alpha complex which allows for the efficient computation of these variants of persistence for data embedded in low dimensions. Their motivation is similar to ours, although both projects were started and developed independently. Our focus is on a single descriptor combining standard persistence with image persistence, while theirs is on different types of persistent homology and connections between them. Natarajan et al., study the Morse theory of chromatic Delaunay complexes [28] and provide an implementation.

**Induced matching.** Our results are also related to work on induced matchings originating in [6]. In particular, similar ideas were independently explored by Gonzalez-Diaz and coauthors [22, 23, 35].

**Persistent homology in machine learning.** Techniques and tools from topological data analysis have recently found a number of fruitful applications to machine learning and in particular deep neural networks [44]. Persistent homology has been used in developing topological network layers [21, 24, 26]. Another approach incorporates topological information about the data into the loss function of a neural network [11, 12, 25, 37]. Topological methods were also used to analyze other aspects of artificial neural networks [9, 30, 32, 38, 40, 42]. In [3] a variant of persistent homology was developed to compare neural networks.

**Disentanglement in deep neural networks.** In machine learning, disentanglement is a broad concept. We are interested in how embeddings corresponding to different classes are separated during the training of a classifier. A motivation for this work goes back to an influential 2014 blog post [29] by Olah on "Neural Networks, Manifolds, and Topology" – ten years later we have appropriate computational tools to address this challenge.

Perhaps the fist explicit mention of (manifold) disentanglement in the context of deep learning is in the work of Brahma, Wu and Yiyuan [8]. The work of Zhou and collaborators provides a topological view on disentanglement in the context of machine learning as well as a thorough overview [43]. We refer the reader interested in the machine learning details there, and focus on geometric-topological aspects of the problem. Our experiments are inspired by the work by Naitzat and collaborators [27] and the follow-up by Wheeler, Bouza and Bubenik [39]. We alleviate some of the shortcomings of these approaches that stem from the limitations of persistent homology applied to a single point cloud. Reani and Bobrowski proposed a data-analysis method [31] based on matching geometrically close cycles.

## 3 Mixup barcodes

In this technical section we propose the mathematical setup for mixup barcodes. We delay detailed interpretation to the next section. We will work with filtrations of simplicial complexes, chain complexes, persistence modules and their decompositions.

**Overview.** In the simple case, we study point clouds $A$ and $B$, via the inclusion $A \hookrightarrow A \cup B$. We ask how the persistence of each topological feature arising from $A$ is shortened by the inclusion of points in $B$. The constructed *mixup barcode* splits each bar in the standard persistence barcode of $A$ into two sub-bars. The first sub-bar quantifies this new, shortened lifetime, and is computed using a variant of persistent homology called image persistence. The second sub-bar is called a mixup sub-bar, and quantifies the shortening of the lifetime.

We follow up with technical definitions, but a reader interested in an intuitive explanation could jump to Section 5 and especially Figure 3.

**Technical definitions.** We will work with two filtrations $L$ and $K$ – which will be connected by an inclusion that we describe in a moment. Specifically, let $K$ denote a filtered finite simplicial complex $K_1 \hookrightarrow K_2 \hookrightarrow \cdots \hookrightarrow K_n$ with inclusion maps $K_i \hookrightarrow K_j$ for $1 \leq i \leq j \leq n$, and similarly for $L$.

We assume that both filtrations are simplex-wise, i.e. for each $i$, $L_{i+1}$ differs from $L_i$ by at most one simplex and similarly for $K$. We further assume that each $\sigma$ added in $L_i$, is not already present in $K$, namely $\sigma \notin K_{i-1}$, where $K_0 = \emptyset$. The index of a simplex is the $i$ at

which it appears in the filtration, which is uniquely defined under the above assumptions. Further, we will sometimes assume that we have a real-valued order-preserving function $f$ assigning a **filtration value** to each simplex.

The above assumptions may be met in the context of Vietoris–Rips filtrations, which will be our practical setup. In particular, while several simplices will typically appear at the same filtration value, it is easy to use an ordering of the simplices to obtain a finer simplex-wise filtration. We remark that everything in this section also generalizes to finite cell complexes.

**The studied inclusion map.** For each $k \geq 0$ we consider the inclusion map $\iota : L \hookrightarrow K$ between our filtrations $L$ and $K$. This gives rise to a variety of algebraic structures and induced maps between them. Of primary interest are the persistence modules $H_k(L)$ and $H_k(K)$ and an induced map between them $H_k(\iota) : H_k(L) \to H_k(K)$. This information is summarized in the following commutative diagram of vector spaces and linear maps.

$$\begin{array}{ccccccccc} H_k(L_1) & \longrightarrow & H_k(L_2) & \longrightarrow & \dots & \longrightarrow & H_k(L_i) & \longrightarrow & \dots & \longrightarrow & H_k(L_n) \\ {\scriptstyle H_k(\iota_1)}\downarrow & & {\scriptstyle H_k(\iota_2)}\downarrow & & & & {\scriptstyle H_k(\iota_i)}\downarrow & & & & {\scriptstyle H_k(\iota_n)}\downarrow \\ H_k(K_1) & \longrightarrow & H_k(K_2) & \longrightarrow & \dots & \longrightarrow & H_k(K_i) & \longrightarrow & \dots & \longrightarrow & H_k(K_n) \end{array} \tag{1}$$

The image of the map of persistence modules in (1) is the following persistence module:

$$\mathrm{im}(H_k(\iota)) = \mathrm{im}(H_k(\iota_1)) \to \mathrm{im}(H_k(\iota_2)) \to \cdots \to \mathrm{im}(H_k(\iota_i)) \to \cdots \to \mathrm{im}(H_k(\iota_n))$$

We are interested in the barcodes of the persistence module $H_k(L)$ (i.e. the persistent homology of $L$) and the persistence module $\mathrm{im}(H_k(\iota))$ (i.e. the image persistent homology of the inclusion). Specifically, we will combine the information contained in these barcodes in a meaningful way, to form a single richer descriptor.

To do so, we notice that under our assumptions for each index $i \in \{1, 2 \dots, n\}$ there is at most one interval in the barcode with minimum value $i$. We call such $i$ the **birth index** of the corresponding interval in the barcode. This allows us to match these barcodes by the birth *indices*; for brevity we refer to the following more general result.

▶ **Theorem 1** (Matching [6, Theorem 4.2, Proposition 5.7])**.** *There is a canonical surjective matching from the barcode of $H_k(L)$ to the barcode of $\mathrm{im}(H_k(\iota))$ which matches intervals with the same birth index.*

With this, we reorganize the information in this canonical matching of barcodes as follows.

▶ **Definition 2** (Mixup barcode)**.** *The **mixup barcode** in degree $k$ of the inclusion $L \hookrightarrow K$ is the collection of triples of indices consisting of a triple $(b, d', d)$ for each pair of matched intervals $[b, d)$ and $[b, d')$ in the barcodes of $H_k(L)$ and $\mathrm{im}(H_k(\iota))$; and a triple $(b, b, d)$ for each unmatched interval $[b, d)$ in the barcode of $H_k(L)$.*

The above definition is in terms of the filtration indices. In practice, we often replace them with the filtration values of the corresponding simplex. Typically the distinction will be clear from context, but sometimes we will explicitly talk about **index mixup barcodes**.

Viewing the mixup barcode as a refinement of the persistence barcode of $L$, we visualize it in a similar way. Specifically, we plot the image sub-bar in a light color, and its complement, the mixup sub-bar, in a darker color. We consider a separate mixup barcode for each degree; see Figure 3 for an example.

**Mixup barcodes for point clouds.** We consider two finite point clouds, $A, B \subset \mathbb{R}^d$, or more generally, a finite metric space $X$ with $A \subset X$ and $B = X \setminus A$, and construct Vietoris–Rips filtrations of $A$ and $A \cup B$, with a common sequence of radii, $r_1 \leq r_2 \leq \cdots \leq r_n$. The simplicial filtrations $L$ and $K$ are given by $L_i = \mathrm{VR}(A; r_i)$ and $K_i = \mathrm{VR}(A \cup B; r_i)$. By the mixup barcode of $A \hookrightarrow A \cup B$, we mean the mixup barcode induced by $L \hookrightarrow K$ as defined above. Unlike a persistence barcode, a mixup barcode is sensitive to the *ordering* of the simplices, since it may affect the matching.

## 4 Related concepts and statistics

With the main technical definition in place, we explain the rationale behind it, and mention some properties. We start with defining an analog of a **persistence pair** [18] that can be expressed either in terms of the indices or filtration values.

▶ **Definition 3** (Mixup triple). *Each triple $(b, d', d)$ in the mixup barcode of $L \hookrightarrow K$ is called an **mixup triple**.*

We reiterate what information is stored in each mixup triple.

▶ **Definition 4** (Persistence bar and image and mixup sub-bars). *For each mixup triple $(b, d', d)$, we have a **persistence bar** $[b, d)$, which splits into what we call:*
- i *an **image sub-bar** $[b,\ d')$, and*
- ii *a **mixup sub-bar** $[d',\ d)$,*

*noting that degenerate sub-bars of the form $[x, x)$ are allowed.*

▶ **Observation 5.** *The mixup barcode of $L \hookrightarrow K$ contains the persistence bars of $L$ and its image sub-bars correspond to the image persistence barcode. However, the mixup sub-bars will generally* not *coincide with any bars of kernel persistence barcode. We provide an explicit example in the full version of the paper.*

▶ **Definition 6** (Premature death). *Given an index mixup triple $(b, d', d)$ we interpret the index $d'$ as the index of premature death of the unique homology class $\gamma$ born at $L_b$. Specifically, we mean the first index at which $\gamma$ becomes trivial or merges with an older homology class in $H_k(L)$.*

▶ **Observation 7.** *We stress that a premature death arises from the image persistent homology vector spaces of our inclusion, namely $\frac{Z_k(L_i)}{B_k(K_j) \cap Z_k(L_i)}$. (Contrast this with the standard persistence: $\frac{Z_k(L_i)}{B_k(L_j) \cap Z_k(L_i)}$.) This premature death is therefore caused by the extra boundaries coming from $K$ – but not by merging with another homology class present in $H_k(K)$ but absent from $H_k(L)$.*

### 4.1 Statistics

It is often useful to extract a single number from a persistence barcode. For example, the total persistence – the sum of the lengths of all bars – is used to quantify the topological complexity of a point cloud. In a similar vein, we define simple ways of extracting a number to quantify the strength of interactions between filtrations $L$ and $K$.

▶ **Definition 8** (Mixup). *The mixup of a mixup triple $(b, d', d)$ is the value $d - d'$.*

In other words, it is the length of the sub-bar $(d', d)$ which quantifies the prematurity of the death of a single homology class.

▶ **Definition 9** (Total mixup). *The **total mixup** of $L \hookrightarrow K$ is the sum of the mixups for the bars in the mixup barcode.*

In our visualization, it is simply the sum of the lengths of the dark bars. We refer to this sum as total-mixup$(A, B)$. Equivalently, it is the difference between the total persistence of $A$ and the total image persistence of $A \hookrightarrow A \cup B$.

**Scale invariant statistics.** The next two definitions allow us to make scale-invariant measurements. Firstly, this allows us to compare data embedded in different dimensions. Secondly, the geometric scale of topological features is irrelevant for our application.

▶ **Definition 10** (Mixup percentage). *The **mixup percentage** of mixup triple $t = (b, d', d)$ is*

$$mixup_{\%}(t) = \frac{mixup(t)}{pers(t)} = \frac{d - d'}{d - b}. \tag{2}$$

▶ **Definition 11** (Total mixup percentage). *The **total mixup percentage** (**mean mixup percentage**) of a mixup barcode is the sum (mean) of the mixup percentages for all its bars.*

Note that the latter variant is between 0 and 1, but is *not* the ratio of the total mixup and the total persistence. The mean mixup percentage has the added benefit of allowing for the comparison between datasets of differing total persistence, which will be crucial in our application.

## 5 Geometric interpretation

In this section we provide intuitive interpretations of the information captured by mixup barcodes in various degrees. It turns out that they often capture natural spatial relationships.

**Holes.** We recall that for a shape embedded in three dimensional space, homology groups in **degree** 0, 1, 2 are often interpreted as various types of holes: the gaps between distinct parts of the object (or dually the connected components); tunnels (or handles); voids enclosed by the object. Note that in general voids correspond to $H_{n-1}$ for objects in $\mathbb{R}^n$. Persistent homology captures the births and deaths of such features along the filtration.

**Spatial relationships for point clouds.** Going back to the case of a pair of point clouds $A$ and $B$ in $\mathbb{R}^3$, the shortening of the lifetime of a homological feature conveys how points from $B$ fill the holes of different types.

Specifically, in degree 0, mixup quantifies to what extent the connected components in $A$ *merge quicker* via points in $B$ – which has natural interpretations as $A$ and $B$ **overlapping** or $A$ being **separated** by $B$. In degree 1, it quantifies how much quicker points in $B$ fill up a loop or tunnel formed by $A$ – for example when a portion of $A$ encircles a portion of $B$ (like in a Hopf link). In degree 2, it quantifies how points in $B$ contribute to filling up a void inside $A$ – which we interpret as a portion of $A$ **surrounding** a portion of $B$. Figure 3 illustrates these spatial relationships and the corresponding mixup barcode.

For the machine learning example illustrated in Figure 2, mixup in degree 1 is most relevant as it captures the red points initially *encircling* the blue points, and then either *tightening* around them, or *separating* from them. For the second case, mixup in degree 0 is also useful.

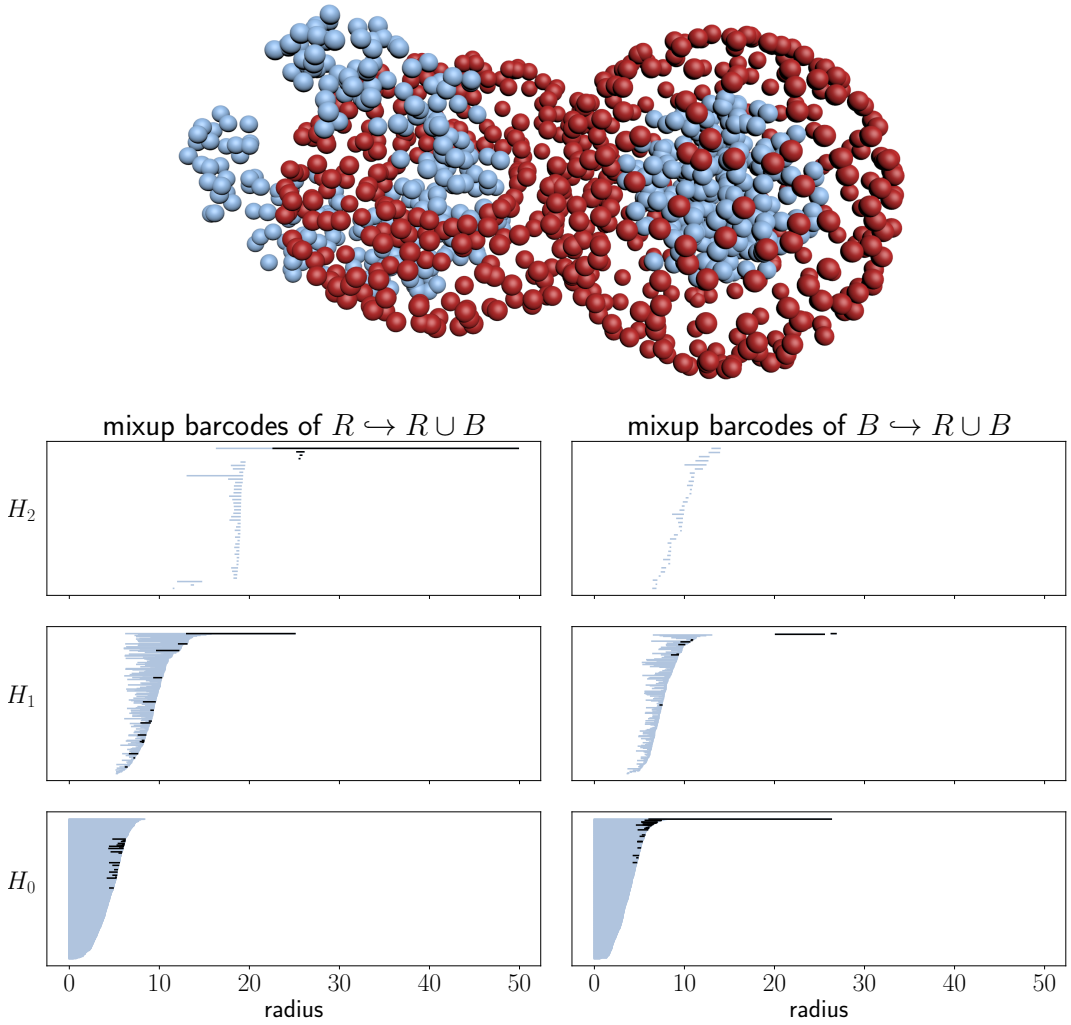



**Figure 3 Top:** Two point clouds: $R$ red (dark) and $B$ blue (light). **Bottom:** Mixup barcodes of $R$ and $B$ included in their union, in degrees $2, 1, 0$. Recall that each persistence bar is split into an image sub-bar and a mixup sub-bar (marked in thick black). **Bottom left:** In $H_2$, we see a prominent persistence bar with a long mixup bar. It detects the red **void** partially filled by blue points, interpreted as the red object **surrounding** a part of the blue object. Similarly in $H_1$ the prominent mixup bar detects the red handle (tunnel) that **encircles** the blue points. No significant mixup occurs in $H_0$, although the many shorter mixup bars detect the **overlap** between the two shapes. **Bottom right:** In $H_1$ the mixup barcode detects the handle **encircled** by the red object, with the late birth indicating that a large part is missing. The prominent bar for $H_0$ detects that the two blue connected components connect quicker via the red points. We interpret this as **separation** of the two blue parts by the red object. As expected, there is no prominent persistence in $H_2$.

In summary, for point clouds in $\mathbb{R}^n$, the mixup barcode captures: **overlap** or **separation** in degree 0 and **surrounding** in degree $n-1$. In remaining degrees it captures how holes in degree $k$ present in $L$ are filled by boundaries in $K$, which corresponds to more subtle and harder to name interactions.

## 6 Algorithm

In this section we describe a simple algorithm for the decomposition of a persistence barcode into a mixup barcode and prove its correctness under the assumptions in Section 3. We start by recalling the algorithms for standard and image persistent homology, which will be part of our construction.

**Standard reduction algorithm.** Persistent homology can be computed using a version of Gaussian elimination on the boundary matrix representing the filtration [18]. We call such an algorithm REDUCE.

The ordering of columns and rows corresponds to the order in which the elements (typically simplices) appear in the filtration. By adding the columns (only from left to right) modulo 2, the algorithm removes conflicts between columns, by which we mean columns having the same pivot. The index persistence pairs are formed by the indices of nonzero columns and their pivots in the final (reduced) matrix. This information does not depend on the choice of column additions [19].

**Image persistent homology computations.** Image persistent homology can be computed using a variant of the above algorithm, as described in [13, pp. 5-6]. Perhaps surprisingly, this algorithm reorders the *rows* of the boundary matrix, so that the elements from $L$ come first (ordered by the filtration index), followed by the elements from $K \setminus L$ (also ordered by the filtration index). The image index persistence pairs are of the form $(\sigma, \tau')$, where the $\tau'$ is the index of a nonempty column whose pivot $\sigma$ corresponds to a simplex in $L$ [13, Observations (ii),(iii)]. This yields a modified Elder rule: The youngest class from $K \setminus L$ is destroyed whenever available; otherwise, the youngest class from $L$ is destroyed, as usual [18]. A cohomological version is obtained by working on anti-transposed matrices [5].

**Mixup barcodes algorithm.** Algorithm 1 outlines the algorithm that decomposes the bars in the standard persistence diagram into the mixup barcode. The key new idea is to coordinate the two matrix reductions, but preparing two filtrations with consistent indexing. These reductions yield the standard and image persistent homology. The latter computation involves the reordering of the rows, as mentioned above. This coordinated reduction allows us to merge the two pieces of information. In contrast, computing the standard and image persistence on two unrelated filtrations loses the correspondence between these two pieces of information. Additionally, we reiterate that mixup bars generally do not coincide with kernel persistence, necessitating the use of our algorithm.

**Computational complexity and implementation.** Due to the use of Gaussian elimination, the algorithm runs in $O(n^3)$ worst-case time, where $n$ is the number of simplices. As with standard persistence [5], the running time is significantly better in practice.

**Algorithm 1** Algorithm for Mixup Barcode in degree $k$.

**Require:** Simplex-wise filtrations $L$ and $K$ as in Section 3; $f$ assigning the filtration value to each simplex; degree $k$

$BK$ = matrix with columns containing the boundaries of the $k$- and $(k+1)$-simplices of $K$ with rows and columns ordered by the filtration index
reorder the rows of $BK$ such that the simplices from $L$ have smaller indices than the simplices of $K \setminus L$ while retaining the relative order of simplices in $L$ and also in $K \setminus L$
form matrix $BL$ by zeroing out the entries in $BK$ involving the simplices of $K \setminus L$
REDUCE matrices $BL$ and $BK$.
**for** each index $\sigma$ of a $k$-simplex in $L$ such that $BL[\sigma] = 0$ **do**
    $birth = f(\sigma)$; $death = death' = \infty$
    **if** $\exists\ \tau$ such that $\text{pivot}(BL[\tau]) = \sigma$ **then** $death = f(\tau)$
    **if** $\exists\ \tau'$ such that $\text{pivot}(BK[\tau']) = \sigma$ **then** $death' = f(\tau')$
    record mixup triple $(birth, death', death)$
return all the recorded mixup triples

▶ **Theorem 12** (Mixup Decomposition Theorem)**.** *The algorithm in Algorithm 1 returns a correct mixup barcode of $L \hookrightarrow K$.*

Proof is in the Appendix of the full version of the paper.

## 7 Experiments

We apply our method to analyzing high-dimensional point clouds. They come from a geometric problem arising in machine learning. The goal is to gain insights about the training of an artificial neural network [2], the **multilayer perceptron** (MLP). It is the simplest *deep learning* method, often used for classification. We remark that models based solely on MLPs can achieve state of the art performance on challenging tasks [34].

**Model parameters.** We use a 5-layer MLP, with embedding dimensions $512, 256, 128, 10$ and $L$, where $L$ depends on the number of considered labels (in our case 3 or 10). The model has $565,248 + 10L$ trainable parameters (weights).

### 7.1 Datasets

We use two well known datasets with different levels of complexity. Note that while performing classification on these datasets is considered solved – understanding how machine learning models achieve their good performance is an open problem.

**MNIST.** This is a standard dataset, containing 60,000 grayscale images of size $28 \times 28$ depicting handwritten decimal digits. It is a heavily preprocessed version of real-world images, so that the Euclidean distance between two images closely corresponds to their perceptual dissimilarity. For this reason, representing the images as a (labeled) point cloud in $\mathbb{R}^{784}$ yields a faithful representation of the data. It comes split into 50,000 training images, and 10,000 test images. We normalize the values to be between 0 and 1. The model achieves 99% test accuracy when trained and tested on the images with labels 0,1 and 2, which we then use for our experiments. These are images of handwritten zeros, ones and twos. It is known to be an easy problem so we expect to see low mixup at the end of training. We are however curious about the higher-dimensional embeddings coming from the intermediate layers.

**CIFAR10.** Another standard dataset, containing 50,000 color photos of size $32 \times 32$ depicting various objects, animals etc. We treat this data as a (labeled) point cloud in $\mathbb{R}^{3072}$, but the Euclidean distance fails to capture the perceptual similarity. We expect to see more significant entanglement between different classes. This dataset is quite challenging for a basic MLP which achieves only 76% test accuracy when trained and tested the images with labels 0, 1 and 2. These are images of airplanes, cars and birds. We expect it to fail to disentangle the data, especially differentiating between airplanes and birds may be hard due to similar backgrounds.

## 7.2 Experiments

We use the new method to analyze geometric data arising from a machine learning model. The goal is to quantify geometric-topological difficulties during training – as theoretically outlined by Olah [29]. The results highlight the usefulness of the new methodology – especially in comparison to existing applications of persistent homology [27,39] that focused on quantifying the shape of a single point cloud.

In geometric terms, such a model learns a sequence of linear transformations each followed by a projection onto the positive orthant of $\mathbb{R}^m$, where $m$ may be different at each step. This is done in an iterative training process, whose goal is to finally embed each input point with label number $q$ near the $q$-th standard basis vector.

**Goals.** We consider the intermediate representation of the data produced by the $i$ initial layers of the model, namely $M_k = \ell_k \circ \ell_{k-1} \circ ... \circ \ell_1$. Given a single input vector $v$, we consider $(M_k(v))_k$, namely the sequence of embeddings of $v$ produced by each $M_k$. Extending this to the entire input point cloud, we get a sequence of point clouds – which additionally change as training progresses. Figure 2 illustrates this for a simplified three-dimensional case.

> We analyze how the representation of input data disentangles passing through the layers – and track this process during training. We aim to verify the conjecture that complicated spatial interactions can hinder training.

We return to the training depicted in Figure 2. In one of the two cases, the model failed to fully disentangle the two point clouds, even though both training and test accuracy reach almost 100%. Such a model is problematic. In this particular case, consider new examples corresponding to the blue region around which the red loop was constricted. These examples are likely to be misclassified, and can for example be exploited in an adversarial attack. More generally, this example shows that geometric-topological information about the embeddings can reveal information about the quality of the training, even when the accuracy (and by analogy the value of the loss function) suggest no such problems.

This situation is of course idealized: the intermediate embedding dimensions can be arbitrarily high, which simplifies the disentanglement problem. Still, complicated entanglement is hypothesized [29, 43] to correlate with training difficulties such as over-fitting and lack of generalization. Indeed, even if disentanglement is theoretically possible, it does not mean that the simplistic training process (oblivious to the global geometric-topological relationships) can realize it. Case in point: in our example, the problem was solvable in this dimension, but one of the training runs failed.

**Datasets.** To train the model, we will be using two datasets MNIST (easier, 99% accuracy) and CIFAR10 (harder, 76% accuracy). See Section 7.1 for more details. After training, we consider the embeddings produced from the test data. We treat the datasets fa finite labeled point cloud in $X \subset \mathbb{R}^d$. By $X_q$ we mean **class** $q$ of dataset $X$, namely the subset of points with label number $q$.

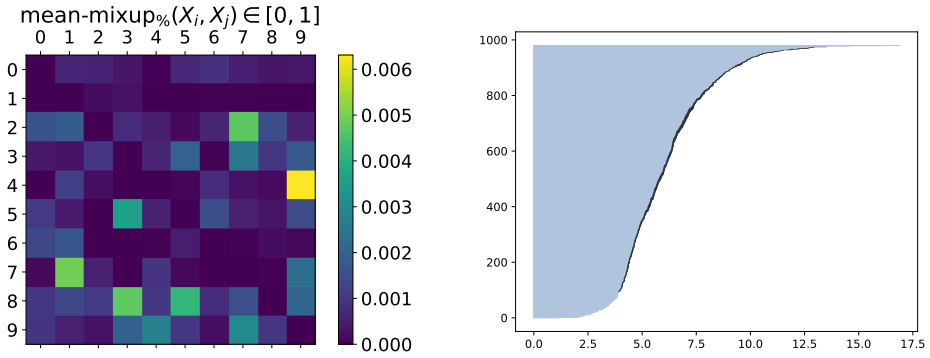



**Figure 4 Left:** The mean mixup percentage (in degree 0) between all pairs of MNIST classes. Most values are low, as expected due to the simplicity of this data. **Right**: The mixup barcode between the images of fours and nines, which have the greatest mixup. Many bars with small – but positive – mixup suggest that the two point clouds overlap along a broad but shallow interface.

**Computations and subsampling.** Due to space restrictions we elaborate on these aspects in the full version of the paper.

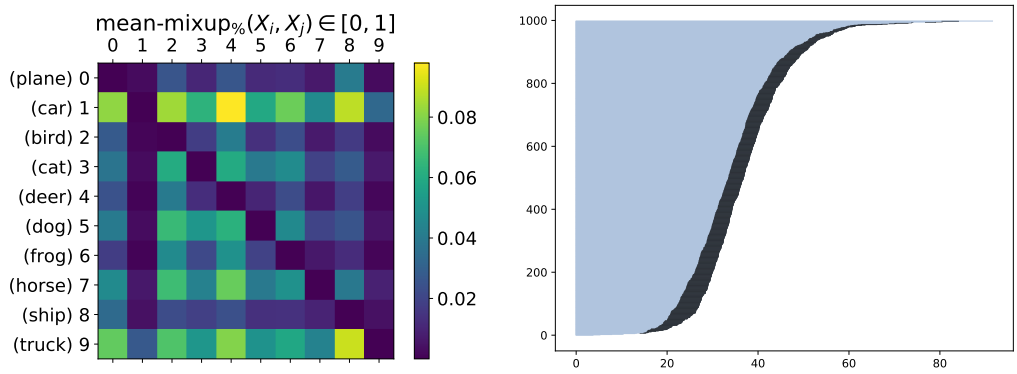



**Figure 5 Left:** The mean mixup percentage (in degree 0) between all pairs of CIFAR10 classes. Note that the values are generally an order of magnitude higher than for the other dataset, with the highest value of 0.09 between classes 1 (car) and 4 (deer). **Right:** The mixup barcode between classes 1 and 4, suggesting a significantly more robust overlap than we saw in the other dataset.

## 7.3 Warmup: mixup barcodes on raw inputs

While our primary focus is to analyze the model (and not the input data), we focus on the raw MNIST data for a moment. We do this because previous studies considered the disentanglement of different classes of this dataset [39] through the lens of persistent homology.

However, strong performance of (non-kernel) linear classifiers on MNIST [15, 41] suggests that pairs of point clouds belonging to different classes are easy to separate using an affine hyperplane. This suggests that they are unlikely to be strongly entangled.

As a sanity check, we verified that despite complicated topology, the mixup (which we use as a proxy for entanglement) is typically low. Specifically, we compute the mean percentage mixup in degree 0 for all pairs of classes of examples, namely for $A = X_i$ and $B = X_j$ for all $0 \leq i, j < 10$. The results for MNIST and CIFAR10 are in Figures 4 and 5.

As expected, CIFAR10 exhibits greater mixup, often by an order of magnitude. It is easy to see that persistence barcodes do not distinguish between the two cases. (The only difference comes from different dimensions of the two datasets.)

We emphasize that in general persistent homology of the data may be complicated, even if subsets of point clouds do not exhibit complex spatial relationships. Using persistence directly may overestimate the amount of entanglement. The extra information present in the mixup barcodes is therefore crucial for robust characterization of such phenomena.

## 7.4 Characterizing the disentanglement across layers

Our goal is to track disentanglement across layers during training. Specifically, we will analyze a labeled point cloud $X_{kqt} \subset \mathbb{R}^n$ for each layer $k$ ($1 \leq k \leq 5$), training epoch $t$, and label $q$ ($1 \leq q \leq 3$). The information about disentanglement will be characterized using the following concept.

▶ **Definition 13** (Mixup profile). *At each layer $k$ and training epoch $t$, the mixup profile is computed as: $P_{kt} = \max_j\{\textit{total-mixup}_{\%}(X_{kjt}, \bigcup_{r\neq j} X_{krt})\}$, in homological degrees 0 and 1.*

The **mixup profile**, $P_{kt}$, measures complexity and strength of interactions between data representations with different labels, serving as a proxy for entanglement. It is visualized as a surface plot, with $k$ (layer) and $t$ (time) on the horizontal axes and $P_{kt}$ (strength of spatial relationship) as the height.

**Outcomes.** See Figures 6 and 7 for a direct visualization of the mixup profiles for the MNIST and CIFAR datasets.

The main observation is that the mixup profiles are significantly different for the intermediate representations of the two datasets, suggesting that the interactions captured by the mixup barcodes indeed correlate with training difficulties.

**Outcomes summary.** The mixup profiles successfully distinguish between the training on the CIFAR and MNIST datasets. In particular, the mixup quickly drops for the easier dataset (MNIST), and not for the harder dataset (CIFAR). Overall, the statistics derived from the mixup barcodes do correlate with difficulties during training. They therefore support the conjecture stated in [29]. We stress that while MNIST and CIFAR10 are simple datasets, we analyze the training process on these datasets – which is poorly understood, even for these datasets. We discuss this further momentarily.

# 8 Discussion

We presented a novel geometric-topological descriptor called a mixup barcode, and its summary statistics. Unlike persistence barcodes, mixup barcodes can be used to characterize not only the shape of a dataset but also its spatial relationships with other datasets. While developed independently, it complements recent development of **Chromatic TDA**, an emerging branch of computational geometry and topology [16]. One advantage of our approach is that a single barcode is produced, which is often simpler to explain and analyze, especially compared to the six barcodes in [16]. Additionally, the mixup barcode retains the

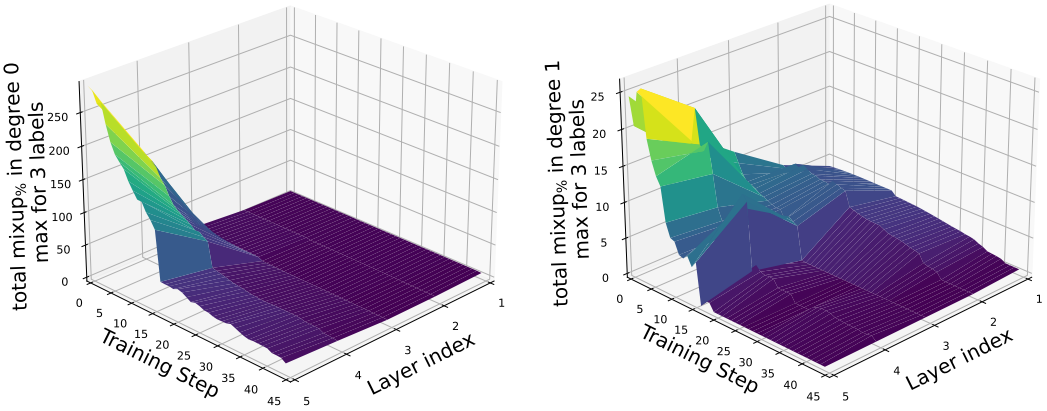



**Figure 6** Mixup profiles characterizing the training on the MNIST dataset, in degree 0 (**left**) and degree 1 (**right**). We recall that the height corresponds to the mixup – the complexity of geometric-topological relationships – between intermediate embeddings of data with different labels at a given training step and layer of the model. The point closest to the viewer corresponds to the last layer at the end of training. Here, the mixup initially *grows* with consecutive layers (which makes sense since the input data is simple, but the matrix in each layer is initially random). On the other hand, mixup drops throughout with successive training iterations. This reflects that the network successfully *disentangles* all the embeddings.

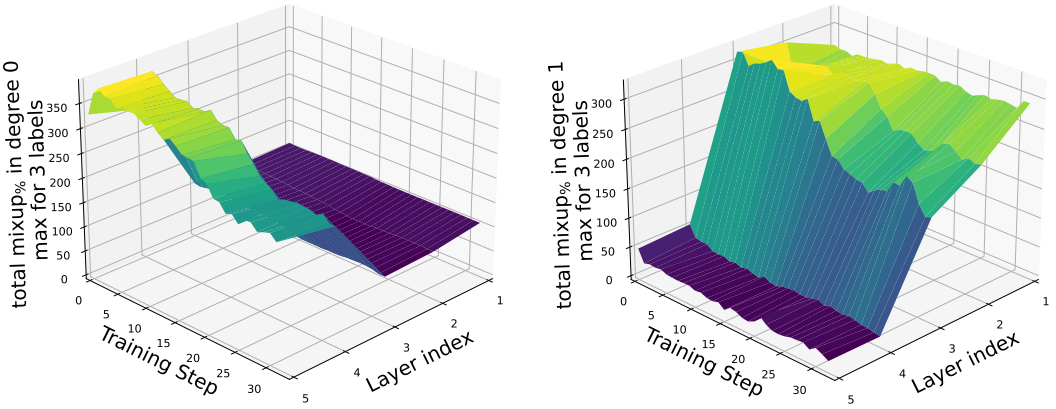



**Figure 7** Mixup profiles characterizing the training on the more challenging CIFAR dataset. While the total mixup percentage generally drops during training, it remains generally much higher for CIFAR compared to MNIST. This is consistent with the worse performance of this model on the CIFAR dataset. In particular, for the last layer, mixup in degree 0 initially drops but stabilizes at a relatively high value (which is consistent with the partially successful training); in degree 1 it remains constant (and much higher than for MNIST) – this hints at more complicated spatial relationships between inputs with different classes that were hard to disentangle. Overall, the graphs suggest that the information captured in the mixup barcodes does indeed correlate with training difficulties and the model's failure to disentangle the embeddings.

crucial matching (between the two relevant barcodes), which is otherwise lost. Note that this information cannot in general be reconstructed using individual barcodes, expressed in terms of filtration values. Indeed, the proposed algorithm, working at the level of *index* persistence of two *consistently indexed* filtrations, is necessary.

We also reiterate that kernel persistence bars capture different information than our mixup sub-bars. For our application, it was crucial to use the total mixup percentage, which relies on the matching between persistence and image bars.

Our experiments mark the first application of Chromatic TDA ideas to analyzing high-dimensional data – as well as the first practical application of image persistence to such data. We believe image persistence (and related variants) are powerful tools, but currently underdeveloped and underutilized. We expect that other pairs (or perhaps larger subsets) of barcodes can be combined in a similar way.

On the applied side, this new tool sheds light on the complicated geometry and topology of the training process of machine learning models. The experiments suggest that the theoretical considerations proposed in [29] (in terms of smooth manifolds, point-set topology and knot theory) can now be verified computationally (on finite point clouds). The main outcome is that mixup barcodes capture spatial interactions that were conjectured to hinder training [29].

We expect that this new tool could help detect problems – such as overfitting – already during training, and regularize (or otherwise better control) the training to avoid such problems altogether.

However, further progress, including more detailed experiments, is limited by underdeveloped computational tools and theory. In particular a vectorization, for example generalizing persistence landscapes [10] or images [1], would allow us to feed the mixup barcodes into a statistical or machine learning pipeline. Relatively slow computation [7] of image persistence is another bottleneck. For now we are limited to simple descriptive statistics on small data.

Still, we do believe that our preliminary experiments hint at the usefulness of mixup barcodes, in particular in the context of high-dimensional data.

## References


1 Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017. URL: `https://jmlr.org/papers/v18/16-337.html`.

2 Shunichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, 1967. `doi:10.1109/PGEC.1967.264666`.

3 Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations. *arXiv preprint arXiv:2201.00058*, 2021.

4 Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021. `doi:10.1007/S41468-021-00071-5`.

5 Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat: Persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017. Algorithms and Software for Computational Topology. `doi:10.1016/j.jsc.2016.03.008`.

6 Ulrich Bauer and Michael Lesnick. Induced matchings and the algebraic stability of persistence barcodes. *Journal of Computational Geometry*, 6(2):162–191, 2015. `doi:10.20382/JOCG.V6I2A9`.

7 Ulrich Bauer and Maximilian Schmahl. Efficient computation of image persistence. In *39th International Symposium on Computational Geometry (SoCG 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.SoCG.2023.14`.

8 Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. Why deep learning works: A manifold disentanglement perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10):1997–2008, 2016. `doi:10.1109/TNNLS.2015.2496947`.

9 Rickard Bruel Gabrielsson and Gunnar Carlsson. Exposition and interpretation of the topology of neural networks. *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pages 1069–1076, 2019. ISBN: 9781728145495. arXiv:1810.03234, doi:10.1109/ICMLA.2019.00180.

10 Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015. doi:10.5555/2789272.2789275.

11 Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2573–2582. PMLR, 16–18 April 2019. URL: https://proceedings.mlr.press/v89/chen19g.html.

12 James R. Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A. Zimmer, Julia A. Schnabel, and Andrew P. King. A Topological Loss Function for Deep-Learning Based Image Segmentation Using Persistent Homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8766–8778, December 2022. doi:10.1109/TPAMI.2020.3013679.

13 David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Persistent homology for kernels, images, and cokernels. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1011–1020. SIAM, 2009. doi:10.1137/1.9781611973068.110.

14 Matija Čufar. Ripserer. jl: flexible and efficient persistent homology computation in julia. *Journal of Open Source Software*, 5(54):2614, 2020. doi:10.21105/JOSS.02614.

15 Dennis DeCoste and Dominic Mazzoni. Fast query-optimized kernel machine classification via incremental approximate nearest support vectors. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 115–122, Washington, DC, USA, 2003. AAAI Press. Includes experiments on all 45 pairwise MNIST digit classification tasks; Table 1 reports linear SVM error rates on raw 784-D pixels for several pairs (e.g., 0 vs 1, 8 vs 3, 2 vs 5). URL: http://www.aaai.org/Library/ICML/2003/icml03-018.php.

16 Sebastiano Cultrera di Montesano, Ondrej Draganov, Herbert Edelsbrunner, and Morteza Saghafian. Chromatic topological data analysis. *arXiv preprint arXiv:2406.04102*, 2024. doi:10.48550/arXiv.2406.04102.

17 Sebastiano Cultrera di Montesano, Ondřej Draganov, Herbert Edelsbrunner, and Morteza Saghafian. Chromatic alpha complexes. *Foundations of Data Science*, 8:30–62, 2026.

18 Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

19 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000. doi:10.1109/SFCS.2000.892133.

20 Herbert Edelsbrunner and János Pach. Maximum betti numbers of čech complexes. In *40th International Symposium on Computational Geometry (SoCG 2024)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SoCG.2024.53.

21 Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A Topology Layer for Machine Learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1553–1563. PMLR, 26–28 August 2020. URL: https://proceedings.mlr.press/v108/gabrielsson20a.html.

22 R. Gonzalez-Diaz, M. Soriano-Trigueros, and A. Torras-Casas. Partial matchings induced by morphisms between persistence modules. *Computational Geometry*, 112:101985, 2023. doi:10.1016/j.comgeo.2023.101985.

23 Rocío González-Díaz, Manuel Soriano-Trigueros, and Álvaro Torras-Casas. Additive partial matchings induced by persistence morphisms. *arXiv preprint*, 2020. arXiv:2006.11100.

24 Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 1633–1643, Red Hook, NY, USA, 2017. Curran Associates Inc.

25 Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in neural information processing systems*, 32, 2019.

26 Kwangho Kim, Jisu Kim, Manzil Zaheer, Joon Sik Kim, Frederic Chazal, and Larry Wasserman. Pllay: Efficient topological layer based on persistence landscapes. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

27 Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *The Journal of Machine Learning Research*, 21(1):7503–7542, 2020.

28 Abhinav Natarajan, Thomas Chaplin, Adam Brown, and Maria-Jose Jimenez. Morse theory for chromatic delaunay triangulations. *arXiv preprint*, 2024. arXiv:2405.19303.

29 Chris Olah. Neural Networks, Manifolds, and Topology, 2014. Accessed: 2024-06-28. URL: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/.

30 Archit Rathore, Nithin Chalapathi, Sourabh Palande, and Bei Wang. TopoAct: Exploring the Shape of Activations in Deep Learning. *arXiv: 1912.06332*, pages 1–16, 2019. arXiv:1912.06332.

31 Yohai Reani and Omer Bobrowski. Cycle registration in persistent homology with applications in topological bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5579–5593, 2022. doi:10.1109/TPAMI.2022.3217443.

32 Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*, 2018.

33 Nico Stucki, Johannes C Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. In *International Conference on Machine Learning*, pages 32698–32727. PMLR, 2023. URL: https://proceedings.mlr.press/v202/stucki23a.html.

34 Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. MLP-mixer: An all-MLP architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

35 Álvaro Torras-Casas, Eduardo Paluzo-Hidalgo, and Rocío González-Díaz. Topological data quality via 0-dimensional persistence matchings. *arXiv preprint*, 2023. arXiv:2306.02411.

36 Hubert Wagner, Nickolas Arustamyan, Matthew Wheeler, and Peter Bubenik. hubwag/Mixup-SoCG26. Software, swhId: swh:1:dir:9b75232380440dc7556b698dd1b7d07882a61fac (visited on 2026-05-18). URL: https://github.com/hubwag/Mixup-SoCG26, doi:10.4230/artifacts.26095.

37 Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. TopoGAN : A Topology-Aware Generative Adversarial Network. In *European Conference on Computer Vision*, pages 118–136. Springer, 2020. doi:10.1007/978-3-030-58580-8_8.

38 Satoru Watanabe and Hayato Yamana. Topological measurement of deep neural networks using persistent homology. *Annals of Mathematics and Artificial Intelligence*, 90(1):75–92, January 2022. doi:10.1007/s10472-021-09761-3.

39 Matthew Wheeler, Jose Bouza, and Peter Bubenik. Activation landscapes as a topological summary of neural network performance. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3865–3870. IEEE, 2021. doi:10.1109/BIGDATA52589.2021.9671368.

40 Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. Link Prediction with Persistent Homology: An Interactive View. In *International conference on machine learning*, pages 11659–11669. PMLR, 2021. URL: http://proceedings.mlr.press/v139/yan21b.html.

**41** Kai Yu, Tong Zhang, and Yihong Gong. Improved local coordinate coding using local tangents. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1215–1222, 2010. Table 1 reports a 12.0% error rate for a linear SVM trained directly on raw MNIST images.

**42** Songzhu Zheng, Yikai Zhang, Hubert Wagner, Mayank Goswami, and Chao Chen. Topological detection of trojaned neural networks. *Advances in Neural Information Processing Systems*, 34:17258–17272, 2021. URL: `https://proceedings.neurips.cc/paper/2021/hash/8fd7f981e10b41330b618129afcaab2d-Abstract.html`.

**43** Sharon Zhou, Eric Zelikman, Fred Lu, Andrew Y Ng, Gunnar E Carlsson, and Stefano Ermon. Evaluating the disentanglement of deep generative models through manifold topology. In *International Conference on Learning Representations*, 2020.

**44** Ali Zia, Abdelwahed Khamis, James Nichols, Zeeshan Hayder, Vivien Rolland, and Lars Petersson. Topological Deep Learning: A Review of an Emerging Paradigm, February 2023. arXiv:2302.03836 [cs]. `doi:10.48550/arXiv.2302.03836`.