Efficient Computation of Image Persistence

Ulrich Bauer ✉ 🏠 ⓘD
Department of Mathematics, TUM School of Computation, Information and Technology, and
Munich Data Science Institute, Technical University of Munich, Germany

Maximilian Schmahl ✉
Universität Heidelberg, Germany

—— **Abstract** ——

We present an algorithm for computing the barcode of the image of a morphism in persistent homology induced by an inclusion of filtered finite-dimensional chain complexes. The algorithm makes use of the clearing optimization and can be applied to inclusion-induced maps in persistent absolute homology and persistent relative cohomology for filtrations of pairs of simplicial complexes. The clearing optimization works particularly well in the context of relative cohomology, and using previous duality results we can translate the barcodes of images in relative cohomology to those in absolute homology. This forms the basis for an implementation of image persistence computations for inclusions of filtrations of Vietoris–Rips complexes in the framework of the software Ripser.

1 Introduction

Over the last few decades, *persistent homology* has established its role as an important tool in data science, with numerous applications in a variety of disciplines, including computer vision, neuroscience, materials science, and evolutionary biology [18, 10, 15, 20, 7]. Recently, there has also been renewed interest in *image persistence*, which is a natural extension of persistent homology [21, 22, 17]. Persistent homology starts with a filtration of simplicial complexes $K_\bullet$ and concerns the *barcode*, which encodes the algebraic structure of the *persistence module* $H_*(K_\bullet)$. In contrast, *image persistence* starts with two filtrations $L_\bullet$ and $K_\bullet$ that are related by a map of filtrations $f_\bullet\colon L_\bullet \to K_\bullet$. This map induces a morphism $H_*(f_\bullet)\colon H_*(L_\bullet) \to H_*(K_\bullet)$. The image of this morphism, $\operatorname{im} H_*(f_\bullet)$, is again a persistence module, and image persistence concerns the barcode of this persistence module.

A key part of the appeal of image persistence is that it enables the construction of meaningful *matchings*, i.e., partial bijections, between the barcodes of the domain and codomain of the morphism one starts with. The first such construction that appeared in the literature was the *induced matching* construction, which was introduced by Bauer and Lesnick [2] to give a proof of the famous *stability theorem* of Cohen-Steiner et al. [11]. Such

constructions have now also appeared in work with a more practical focus, with Reani and Bobrowski [21] and García-Redondo et al. [17] proposing and applying general schemes for matching cycles in different filtrations using image persistence, as well as Stucki et al. [22] applying such a method in the context of medical image data analysis.

The first algorithm for computing image persistence in a special case was proposed by Cohen-Steiner et al. [12] for maps $f_\bullet$ of the form $L_\bullet = K_\bullet \cap L \hookrightarrow K_\bullet$ for some fixed subcomplex $L \subseteq K$ (*one-filtration setting*). An implementation for the algorithm from [12] exists in the framework of the software Dionysus by Dmitriy Morozov [19]. The algorithm described by Cohen-Steiner et al. is similar to the standard algorithm for a single filtration and naturally does not make use of many important speed-ups that have been developed for the computation of the barcode of a single filtration since the publication of [12]. Cohen-Steiner et al. also propose an adaption of their method to the general (*two-filtration*) setting using a mapping cylinder construction, which however has never been implemented and might not be computationally feasible. The goal of the present work is to adapt some of the speed-ups for a single filtration to the computation of image persistence, and to show that the resulting algorithm also works for general injective maps $f_\bullet$ without the intersection assumption and without the need for the mapping cylinder construction.

The basic algorithm for computing persistent homology is based on performing *matrix reduction*, a variant of column-wise Gaussian elimination, on a *boundary matrix* associated to the given filtration of simplicial complexes. This algorithm can be made faster using the *clearing* optimization, introduced by Chen and Kerber in [9], and also used implicitly in the cohomology algorithm by de Silva et al. [16]. In short, this optimization makes use of the homological grading of the boundary matrix to disregard certain unnecessary columns in the reduction process. The basic algorithm for image persistence additionally requires the reduction of a permuted boundary matrix, to which clearing cannot be straightforwardly applied. We will remedy this by showing that one can delete the columns in the permuted boundary matrix that were already reduced to 0 in the boundary matrix corresponding to the codomain filtration.

The clearing optimization works particularly well in conjunction with cohomology based algorithms. These were first studied by de Silva et al. in [16] for the single filtration case and justified by certain duality results that provide a translation between barcodes for persistent homology and for *persistent cohomology*, as well as the barcodes for *persistent relative homology* $H_*(K, K_\bullet)$: $H_*(K, K_0) \to \cdots \to H_*(K, K)$ and similarly for *persistent relative cohomology*. These duality results were recently extended by Bauer and Schmahl in [5] in order to also provide translations for images of $H_*(f_\bullet)$ and $H^*(f_\bullet)$, as well as their relative counterparts $H_*(f, f_\bullet)$ and $H^*(f, f_\bullet)$. This allows us to perform cohomology based computations and still obtain the desired barcodes in homology.

To apply clearing in the relative cohomology setting for image persistence, we will reformulate the algorithm for image persistence by Cohen-Steiner et al. [12] in the purely algebraic setting of filtered chain complexes of vector spaces. More precisely, we will consider two filtrations of (co)chain complexes $C_\bullet$ and $C'_\bullet$ and a monomorphism $\varphi_\bullet \colon C_\bullet \to C'_\bullet$. This setup includes both the absolute homology case $C_*(L_\bullet) \hookrightarrow C_*(K_\bullet)$ and the relative cohomology case $C^*(K, K_\bullet) \hookrightarrow C^*(L, L_\bullet)$ from before. The general idea for computing the image of $H_*(\varphi_\bullet)$ is to first write it as a subquotient of $C'_\bullet$:

$$\operatorname{im} H_*(\varphi_\bullet) \cong \frac{\varphi_\bullet(Z_*(C_\bullet))}{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)},$$

where the intersection of persistence modules is to be interpreted indexwise, meaning that $(\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet))_t = \varphi_t(Z_*(C_t)) \cap B_*(C'_t)$.

Performing matrix reductions that make use of the clearing optimization, we will find a pair of inclusion-related *filtration compatible bases* for the filtrations appearing in the equation above. Filtration compatible bases provide a formal framework for many standard arguments for barcode computations via matrix reduction, and they can be interpreted as special cases of *matching diagrams*, which are equivalent to barcodes [3]. Using the general theory of matching diagrams, the data we compute can easily be shown to determine the barcode of $\operatorname{im} H_*(\varphi_\bullet)$.

Applying these general considerations in the relative cohomology setting and combining this with the translation between relative cohomology and absolute homology from [5] yields an algorithm for computing the absolute homology image of $f_\bullet \colon L_\bullet \to K_\bullet$ by reducing two coboundary matrices that can be reduced with clearing as summarized in our main result Theorem 22. An implementation of this method based on Ripser [1] is publicly available [6] and we provide some computational benchmarks. Our software works under the assumption that $L_\bullet = \operatorname{Rips}_\bullet(X, d)$ and $K_\bullet = \operatorname{Rips}_\bullet(X, d')$ are filtrations of Vietoris–Rips complexes corresponding to two metrics d and d' on a finite set X that satisfy $d(x, y) \geq d'(x, y)$ for all $x, y \in X$. This ensures that $L_t = \operatorname{Rips}_t(X, d)$ is a subcomplex of $K_t = \operatorname{Rips}_t(X, d')$ for all t, with the maps $f_t \colon L_t \to K_t$ being given by inclusion. The implementation also makes uses of a version of the emergent and apparent pairs optimizations, which shortcuts the construction of the coboundary matrix and reduces the memory requirements for storing persistence pairings [1].

Contributions

- We propose the first algorithm for the general problem of computing the image of a map in persistent homology induced by an inclusion of filtrations of simplicial complexes, without imposing any restrictions on the subfiltration (called the "two function setting" in [12]) and without the inefficient use of a mapping cylinder (Theorem 22).
- We show that our general method can be augmented by the most important optimizations in persistence computations, including clearing (Corollary 20), cohomology based computations (Proposition 21), and apparent pairs (Section 3.4).
- We provide an implementation in the framework of Ripser [6] and experiments on data sets of varying difficulty (Section 3.5).
- This enables the use of image persistence and consequently induced matchings in computational settings, such as supervised learning [22, 17].

Notation. Throughout the paper, we fix a totally ordered set $(T, \leq)$ to be $\{0, \ldots, n\}$ with the obvious order and a field $\mathbb{F}$ over which all vector spaces are considered.

2 Linear Algebra for Filtrations

In this section, we develop some machinery based on filtration compatible bases, which forms the foundation for our constructions of image persistence barcodes. First, we need to recall some basic theory for persistence modules and barcodes. We write **Vec** for the category of vector spaces over our fixed field $\mathbb{F}$. We fixed $T = \{0, \ldots, n\}$ as a finite totally ordered index set, and we write **T** for T considered as a poset category.

▶ **Definition 1.** *The category of* persistence modules indexed by **T** *is defined as the category* **Vec$^{\mathbf{T}}$** *whose objects are functors* **T** $\to$ **Vec** *and whose morphisms are natural transformations.*

Since $\mathbf{T}$ is a small category and $\mathbf{Vec}$ is an abelian category, the functor category $\mathbf{Vec^T}$ is again abelian, with kernels, cokernels, images, direct sums, and more generally, all limits and colimits given pointwise. The prime example for a persistence module is the persistent homology of a filtration of spaces. Other examples are given by *interval modules.* If $I \subseteq T$ is an interval, the corresponding interval module $C(I)_\bullet$ is defined by

$$C(I)_t = \begin{cases} \mathbb{F} & \text{if } t \in I, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{with structure maps} \qquad C(I)_{t,u} = \begin{cases} \mathrm{id}_\mathbb{F} & \text{if } t, u \in I, \\ 0 & \text{otherwise.} \end{cases}$$

These interval modules are of particular interest because they lead to a structure theory for persistence modules.

▶ **Definition 2.** *If there is a family of intervals $(I_\alpha)_{\alpha \in A}$ such that for a persistence module $M_\bullet$ we have $M_\bullet \cong \bigoplus_{\alpha \in A} C(I_\alpha)_\bullet$, then $M_\bullet$ is said to have a* barcode *given by $(I_\alpha)_{\alpha \in A}$.*

If a persistence module has a barcode, then it is unique, by a version of the Krull–Remak–Schmidt–Azumaya Theorem [8, Theorem 2.7]. In this paper, we will only consider persistence modules consisting of finite dimensional vector spaces, which are guaranteed to have a barcode by Crawley-Boevey's Theorem [14].

Persistent homology is the homology of a chain complex of persistence modules. In practice, the persistence modules forming these chain complexes arise from filtrations of simplicial complexes, so their structure maps are all inclusions. We will now study this kind of persistence module more closely, as our later considerations will mostly happen in terms of chain complexes rather than in terms of homology.

▶ **Definition 3.** *We say that a persistence module $M_\bullet$ is a* filtration *of the vector space $M = M_n$ if for all $t \le u$ the structure map $M_{t,u}$ is a subspace inclusion $M_t \hookrightarrow M_u$. For any $m \in M$, we define its* support *in $M_\bullet$ as $\mathrm{supp}_{M_\bullet}(m) = \{t \in T \mid m \in M_t\}$. A basis $\mathfrak{M}$ of M is said to be* filtration compatible *if $\mathfrak{M}_t = \mathfrak{M} \cap M_t$ is a basis for M_t for all $t \in T$. An ordered basis $(\mathfrak{M}, \le)$ for M is said to be a* filtration compatible ordered basis *if it is filtration compatible and $m \le m' \in \mathfrak{M}$ implies $\mathrm{supp}\, m' \subseteq \mathrm{supp}\, m$.*

If $M_\bullet$ and $M'_\bullet$ are filtrations of vector spaces, we write $M_\bullet \subseteq M'_\bullet$ if $M_t \subseteq M'_t$. We write $M'_\bullet / M_\bullet$ for the persistence module given by $(M'_\bullet / M_\bullet)_t = M'_t / M_t$. Similarly, if $M''_\bullet$ is another filtration with $M''_\bullet \subseteq M'_\bullet$, we write $M_\bullet \cap M''_\bullet$ for the persistence module given by $(M_\bullet \cap M''_\bullet)_t = M_t \cap M''_t$.

Observe that if $M_\bullet$ is a filtration of vector spaces and $\mathfrak{M}$ is a filtration compatible basis, then $(\mathrm{supp}(m))_{m \in \mathfrak{M}}$ is a barcode of $M_\bullet$. By interpreting $\mathfrak{M}$ as a so-called matching diagram, this may be seen as a special case of the general equivalence of matching diagrams and barcodes [3]. This theory also yields the following result that forms the basis for our computational results.

▶ **Proposition 4.** *Let $M_\bullet \subseteq M'_\bullet$ be filtrations of vector spaces with respective filtration compatible bases $\mathfrak{M}$ and $\mathfrak{M}'$ related by an inclusion $\mathfrak{M} \subseteq \mathfrak{M}'$. Then $M'_\bullet / M_\bullet$ has the barcode*

$$\left(\mathrm{supp}_{M'_\bullet}(m) \setminus \mathrm{supp}_{M_\bullet}(m) \right)_{m \in \mathfrak{M}} \cup \left(\mathrm{supp}_{M'_\bullet}(m) \right)_{m \in \mathfrak{M}' \setminus \mathfrak{M}}.$$

We now state some helpful facts about filtration compatible bases. We refer to the full version of this paper [4] for the proofs. We start with a lemma relating supports of basis elements with filtration compatibility.

▶ **Lemma 5.** *Let $M_\bullet$ be a filtration of the vector space M with filtration compatible basis $\mathfrak{M}$. Let $\mathfrak{M}'$ be another basis for M such that there exists a bijection $g \colon \mathfrak{M} \to \mathfrak{M}'$ with $\mathrm{supp}_{M_\bullet}(m) = \mathrm{supp}_{M_\bullet}(g(m))$ for all $m \in \mathfrak{M}$. Then $\mathfrak{M}'$ is a filtration compatible basis for $M_\bullet$.*

Next, we extend a standard fact about intersections of vector spaces to filtrations.

▶ **Lemma 6.** *Let $M'_\bullet, M''_\bullet \subseteq M_\bullet$ be filtrations of vector spaces and let $\mathfrak{M}'$ and $\mathfrak{M}''$ be filtration compatible bases for $M'_\bullet$ and $M''_\bullet$, respectively, such that $\mathfrak{M}' \cup \mathfrak{M}''$ is linearly independent. Then $\mathfrak{M}' \cap \mathfrak{M}''$ is a filtration compatible basis for $M'_\bullet \cap M''_\bullet$. Moreover, for all $m \in \mathfrak{M}' \cap \mathfrak{M}''$*

$$\operatorname{supp}_{M'_\bullet \cap M''_\bullet}(m) = \operatorname{supp}_{M'_\bullet}(m) \cap \operatorname{supp}_{M''_\bullet}(m).$$

We will use the special case where $M'_\bullet$ is included in $M''_\bullet$ at the last filtration step (but not necessarily before):

▶ **Corollary 7.** *Let $M'_\bullet, M''_\bullet \subseteq M_\bullet$ be filtrations of vector spaces $M' \subseteq M'' \subseteq M$, respectively. Moreover, let $\mathfrak{M}' \subseteq \mathfrak{M}''$ be filtration compatible bases for $M'_\bullet$ and $M''_\bullet$, respectively. Then $\mathfrak{M}'$ is a filtration compatible basis for $M'_\bullet \cap M''_\bullet$. Moreover, for all $m \in \mathfrak{M}'$*

$$\operatorname{supp}_{M'_\bullet \cap M''_\bullet}(m) = \operatorname{supp}_{M'_\bullet}(m) \cap \operatorname{supp}_{M''_\bullet}(m).$$

Finally, we state a version of the rank-nullity-theorem for filtrations.

▶ **Lemma 8.** *Let $\phi_\bullet \colon M_\bullet \to P_\bullet$ be a morphism of filtrations of vector spaces and consider the linear map $\phi = \phi_n \colon M \to P$. Let $\mathfrak{M}$ be a filtration compatible basis for $M_\bullet$, let $\mathfrak{M}' = \mathfrak{M} \cap \ker \phi$, and assume that $\mathfrak{M}'' = (\phi(m))_{m \in \mathfrak{M} \setminus \mathfrak{M}'}$ is a linearly independent family of vectors. Then*
- *$\mathfrak{M}'$ is a filtration compatible basis for $\ker \phi_\bullet$,*
- *$\mathfrak{M}''$ is a filtration compatible basis for $\operatorname{im} \phi_\bullet$,*
- *$\operatorname{supp}_{\ker \phi_\bullet}(m') = \operatorname{supp}_{M_\bullet}(m')$ for all $m' \in \mathfrak{M}'$, and*
- *$\operatorname{supp}_{\operatorname{im} \phi_\bullet}(\phi(m)) = \operatorname{supp}_{M_\bullet}(m)$ for all $m \in \mathfrak{M} \setminus \mathfrak{M}'$.*

Note that if one drops the assumption of the above lemma that $P_\bullet$, and hence the image $\operatorname{im} \phi_\bullet$, is a filtration, then it may happen that $\mathfrak{M}' = \mathfrak{M} \cap \ker \phi$ is a basis for the vector space $\ker \phi$ but not a filtration compatible basis for the filtration $\ker \phi_\bullet$.

3 Computing Image Persistence Barcodes

Recall that we fixed a finite totally ordered index set $T = \{0, \ldots, n\}$ and a field $\mathbb{F}$ over which we consider vector spaces. For our purposes, a chain (resp. cochain) complex is a graded finite dimensional vector space with a differential of degree -1 (resp. 1) that squares to 0. A chain complex of persistence modules $C_\bullet$ with differential $\partial_\bullet$ is called a filtration of a chain complex of vector spaces C with differential ∂ if $C_\bullet$ is a filtration of C as a vector space and $\partial_n = \partial$. Recall that a basis for the final vector space in a filtration is called filtration compatible if it yields bases for the constituent vector spaces of the filtration by intersecting. Further, recall that if the basis is ordered, we say that it is a filtration compatible ordered basis if its order refines the order in which the basis elements appear in the filtration.

▶ **Definition 9.** *If $C_\bullet$ is a filtration of the (co)chain complex C with a filtration compatible ordered basis $\mathfrak{C}$, then the matrix D representing the (co)boundary operator on C with respect to $\mathfrak{C}$ is called* filtration (co)boundary matrix.

▶ **Example 10.** *If $K_\bullet : \emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$ is a filtration of finite simplicial complexes, we get a filtration of chain complexes $C_*(K_\bullet)$. A filtration compatible ordered basis is given by the simplices of K, ordered by a linear refinement of the order in which they appear in the filtration. If D^K is a corresponding filtration boundary matrix, then one can check (see [16]) that $(D^K)^\perp$ is a filtration coboundary matrix for the filtration of relative*

cochains $0 = C^*(K, K) = C^*(K, K_n) \subseteq \cdots \subseteq C^*(K, K_0) = C^*(K, \emptyset) = C^*(K)$. The matrix represents the coboundary operator on $C^*(K)$ with respect to the dual basis corresponding to the simplices of K, ordered by the opposite of the filtration order. Here, $(-)^\perp$ denotes taking the transpose of a matrix along its anti-diagonal.

To avoid notational clutter, we will from now on only talk about chain complexes in the general setting, but everything also straightforwardly applies to cochain complexes.

▶ **Definition 11.** *If X is a matrix, we write x_i for the i-th column of the matrix X. For a non-zero column vector x_i, we define $\mathrm{Pivot}\, x_i$ as the largest index where the column has a non-zero entry. We write $\mathrm{Pivots}\, X$ for the set of all indices which occur as pivots of non-zero columns of X. A matrix is called* reduced *if no two non-zero columns have the same pivot.*

Note that any set of non-zero vectors with unique pivots is linearly independent. In particular, the non-zero columns of a reduced matrix are linearly independent.

Computing the barcode for the homology of a filtration of a chain complex is done by *reducing* a filtration boundary matrix D, i.e., performing a variant of Gaussian elimination on the columns of this matrix until one obtains a reduced matrix. This can be expressed as finding a reduced matrix R and a full-rank upper-triangular matrix V such that $R = DV$. The columns of these matrices naturally represent elements of C by interpreting them as coordinate vectors with respect to the ordered basis $\mathfrak{C}$. The barcode for persistent homology may then be obtained from this data as follows.

▶ **Theorem 12** (Cohen-Steiner et al. [13])**.** *Let D be a filtration boundary matrix of a filtration of chain complexes $C_\bullet$ and assume we have a full-rank and upper-triangular matrix V such that $R = DV$ is reduced. Then $H_*(C_\bullet)$ has a barcode given by the multiset*

$$\left\{\mathrm{supp}_{C_\bullet}(r_j) \setminus \mathrm{supp}_{C_\bullet}(v_j) \mid r_j \neq 0\right\} \cup \left\{\mathrm{supp}_{C_\bullet}(v_i) \mid r_i = 0 \text{ and } i \notin \mathrm{Pivots}\, R\right\}.$$

The supports of column vectors that appear in the theorem can easily be determined from the initial data via pivots: If $\mathfrak{M}$ is a filtration compatible ordered basis for a filtration $M_\bullet$ of a vector space M, then we can consider elements of M via their coordinate vectors with respect to $\mathfrak{M}$. Because $\mathfrak{M}$ is a filtration compatible ordered basis, we then have $\mathrm{supp}_{M_\bullet}(v) = \mathrm{supp}_{M_\bullet}(v')$ if and only if $\mathrm{Pivot}\, v = \mathrm{Pivot}\, v'$ for any two such coordinate vectors v and v'. In particular, this means that in the setting of simplicial complexes the support of a column vector is the same as the support of its pivot simplex.

The theorem is formulated in a different language by Cohen-Steiner et al., but the version above also follows as a special case from Theorem 14. Note that the theorem is also compatible with the homological grading: Assume that $C_\bullet = \bigoplus_d C_{\bullet,d}$ is graded with ∂ mapping $C_{\bullet,d}$ to $C_{\bullet,d-1}$. If the filtration compatible ordered basis $\mathfrak{C}$ used to build D is chosen such that its intersection with each grading summand is a filtration compatible ordered basis for that summand, then one gets a barcode for $H_d(C_\bullet)$ by restricting the barcode given in Theorem 12 to those intervals coming from columns that represent d-dimensional cycles.

3.1 Image Barcodes via Matrix Reduction

We now turn to the setting of image persistence. Let $C_\bullet$ and $C'_\bullet$ be filtrations of the chain complexes C and C' with corresponding filtration compatible ordered bases $\mathfrak{C}$ and $\mathfrak{C}'$. Let D and D' be the corresponding filtration boundary matrices. Assume that we are given an injection of filtrations $\varphi_\bullet \colon C_\bullet \to C'_\bullet$ such that the map $\varphi \colon C \to C'$ on the final filtration step is an isomorphism. Note that this is not a restriction, as any injection of filtrations can be

extended to one satisfying this assumption, and subsequently restricting the barcodes to the original indexing set provides the desired result for this more general setting as well. Let F be the matrix representing φ with respect to $\mathfrak{C}$ and $\mathfrak{C}'$ and define the *mixed basis boundary matrix* $D^\varphi = DF^{-1} = F^{-1}D'$. The columns of D^φ thus correspond to $\mathfrak{C}'$, while the rows correspond to $\mathfrak{C}$.

▶ **Example 13.** If $K_\bullet : \emptyset = K_0 \subseteq \cdots \subseteq K_n = K$ and $L_\bullet : \emptyset = L_0 \subseteq \cdots \subseteq L_n = L$ are filtrations of finite simplicial complexes, we get filtrations of chain complexes $C_*(K_\bullet)$ and $C_*(L_\bullet)$. If we are given a monomorphism $f_\bullet \colon L_\bullet \to K_\bullet$ that induces an isomorphism $L \to K$ (i.e., assuming that $L_i \subseteq K_i$ for all i and $L = K$), then we are in the setting above. Filtration compatible ordered bases are given by the simplices of K and L, ordered by a linear refinement of the order in which they appear in the respective filtrations. Let D^L and D^K denote the corresponding filtration boundary matrices, and let D^f denote the mixed basis boundary matrix for the induced map $C_*(L_\bullet) \to C_*(K_\bullet)$. Then, analogously to Example 10, we obtain that $(D^L)^\perp$, $(D^K)^\perp$ and $(D^f)^\perp$ are the filtration and mixed basis coboundary matrices for the relative cohomology counterpart $C^*(K, K_\bullet) \to C^*(L, L_\bullet)$. In the mixed matrix $(D^f)^\perp$, the columns thus correspond to $L_\bullet$, while the rows correspond to $K_\bullet$.

Our goal is to determine a barcode for $\operatorname{im} H_*(\varphi_\bullet)$ by reducing the matrices D and D^φ. Assume that we have $R = DV$ and $R^\varphi = D^\varphi V^\varphi$ reduced with V and V^φ full-rank and upper-triangular. The columns of the matrices R, D, V, R^φ, and D^φ naturally represent elements of C by interpreting them as coordinate vectors with respect to $\mathfrak{C}$. Similarly, the columns of V^φ naturally represent elements of C' by interpreting them as coordinate vectors with respect to $\mathfrak{C}'$. Recall that if X is a matrix, we denote its jth column by x_j. The main result can then be stated as follows.

▶ **Theorem 14.** *The image of $H_*(\varphi_\bullet)$ has a barcode given by the multisets*

$$\left\{\operatorname{supp}_{C_\bullet}(r_j^\varphi) \setminus \operatorname{supp}_{C_\bullet'}(v_j^\varphi) \neq \emptyset \mid r_j^\varphi \neq 0\right\} \cup \left\{\operatorname{supp}_{C_\bullet}(v_i) \mid r_i = 0 \text{ and } i \notin \operatorname{Pivots} R\right\}.$$

Note that the intervals $\operatorname{supp}_{C_\bullet}(v_i)$ in the barcode of $\operatorname{im} H_*(\varphi_\bullet)$ that are not bounded above are precisely the same as those in the barcode of $H_*(C_\bullet)$ as given in Theorem 12.

The proof of Theorem 14 will be based on a sequence of intermediate results. As mentioned in the introduction, the general idea is to write

$$\operatorname{im} H_*(\varphi_\bullet) \cong \frac{\varphi(Z_*(C_\bullet))}{\varphi(Z_*(C_\bullet)) \cap B_*(C_\bullet')},$$

and to find filtration compatible bases $\mathfrak{Z}$ and $\mathfrak{B}$ for $\varphi(Z_*(C_\bullet))$ and $\varphi(Z_*(C_\bullet)) \cap B_*(C_\bullet')$, respectively, such that $\mathfrak{B} \subseteq \mathfrak{Z}$ holds so that we can apply Proposition 4.

If X is a matrix, we will write $\operatorname{cols} X$ for the family of all its non-zero column vectors.

▶ **Lemma 15.** *The family* $\operatorname{cols} V^\varphi$ *is a filtration compatible basis for* $C_\bullet'$, $\mathfrak{B} = \operatorname{cols} FR^\varphi$ *is a filtration compatible basis for* $B_*(C_\bullet')$, *and for all j with $r_j^\varphi \neq 0$ we have*

$$\operatorname{supp}_{B_*(C_\bullet')}(Fr_j^\varphi) = \operatorname{supp}_{C_\bullet'}(v_j^\varphi).$$

Proof. We start by showing that $\operatorname{cols} V^\varphi$ is a filtration compatible basis for $C_\bullet'$: We have $\operatorname{Pivot} v_j^\varphi = j$ since V^φ is full-rank and upper-triangular. It follows that v_j^φ has the same support in C' as the jth element of $\mathfrak{C}'$. Thus, $\operatorname{cols} V^\varphi$ is a filtration compatible basis for $C_\bullet'$ by Lemma 5.

Next, note that $(\partial(v))_{v \in \mathrm{cols}\, V^\varphi \setminus \ker \partial} = \mathrm{cols}\, FR^\varphi$ is linearly independent since R^φ is reduced and F has full rank. Thus, we can apply Lemma 8 to the map of filtrations $\partial_\bullet \colon C'_\bullet \to C'_\bullet$ and the filtration compatible basis $\mathrm{cols}\, V^\varphi$ to obtain that $\mathrm{cols}\, FR^\varphi$ is a filtration compatible basis for $B_*(C'_\bullet) = \mathrm{im}\, \partial_\bullet$. The assertion on the supports follows from the support formula in Lemma 8. ◀

Now that we have a filtration compatible basis for $B_*(C'_\bullet)$, we want to extend it to a filtration compatible basis for $\varphi_\bullet(Z_*(C_\bullet))$.

▶ **Lemma 16.** *Let* $\mathfrak{X} = \mathrm{cols}\, R^\varphi \cup \{v_j \mid j \notin \mathrm{Pivots}\, R^\varphi\}$ *and* $\mathfrak{X}' = \mathfrak{X} \cap \ker \partial$. *Then* $\mathfrak{X}$ *is a filtration compatible basis for* $C_\bullet$, $\mathfrak{Z} = F\mathfrak{X}' = \mathfrak{B} \cup \{Fv_j \mid j \notin \mathrm{Pivots}\, R^\varphi\}$ *is a filtration compatible basis for* $\varphi_\bullet(Z_*(C_\bullet))$, *and for all* $x \in \mathfrak{X}'$ *we have*

$$\mathrm{supp}_{\varphi_\bullet(Z_*(C_\bullet))}(Fx) = \mathrm{supp}_{C_\bullet}(x).$$

Proof. We start by showing that $\mathfrak{X}$ is a filtration compatible basis for $C_\bullet$. The same argument as in the beginning of the proof of Lemma 15 yields that $\mathrm{cols}\, V$ is a filtration compatible basis for $C_\bullet$. Next, note that $\mathfrak{X}$ is linearly independent since all elements have unique pivots: R^φ is reduced and we only consider those v_j with $\mathrm{Pivot}\, v_j = j \notin \mathrm{Pivots}\, R^\varphi$. Moreover, we have a bijection $\mathfrak{X} \to \mathrm{cols}\, V$ given by mapping v_j to itself and mapping r_j^φ to v_i for $i = \mathrm{Pivot}\, r_j^\varphi$. Recall that $\mathrm{Pivot}\, v_i = i = \mathrm{Pivot}\, r_j^\varphi$ implies $\mathrm{supp}_{C_\bullet}(r_j^\varphi) = \mathrm{supp}_{C_\bullet}(v_i)$. Since $\mathrm{cols}\, V$ is a filtration compatible basis for $C_\bullet$, Lemma 5 now implies that $\mathfrak{X}$ is also a filtration compatible basis for $C_\bullet$.

Since R is reduced and thus $(\partial(v))_{v \in \mathfrak{X} \setminus \mathfrak{X}'} \subseteq \mathrm{cols}\, R$ is linearly independent, we can apply Lemma 8 to the boundary operator $\partial_\bullet \colon C_\bullet \to C_\bullet$ and the filtration compatible basis $\mathfrak{X}$. We obtain that $\mathfrak{X}' = F^{-1}\mathfrak{Z}$ is a filtration compatible basis for $\ker \partial_\bullet = Z_*(C_\bullet)$ with $\mathrm{supp}_{Z_*(C_\bullet)}(x) = \mathrm{supp}_{C_\bullet}(x)$ for all $x \in \mathfrak{X}'$. The claim now follows from the fact that $\varphi_\bullet$ is mono, so that its restriction is an isomorphism $Z_*(C_\bullet) \to \varphi_\bullet(Z_*(C_\bullet))$ represented by F. ◀

Since the filtration compatible basis $\mathfrak{B}$ for $B_*(C'_\bullet)$ extends to a basis $\mathfrak{Z}$ for $\varphi_\bullet(Z_*(C_\bullet))$, we can conclude that $\mathfrak{B}$ is also a filtration compatible for $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)$.

▶ **Lemma 17.** *The family* $\mathfrak{B} = \mathrm{cols}\, FR^\varphi$ *is a filtration compatible basis for* $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)$, *and for all* j *with* $r_j^\varphi \neq 0$ *we have*

$$\mathrm{supp}_{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)}(Fr_j^\varphi) = \mathrm{supp}_{C_\bullet}(r_j^\varphi) \cap \mathrm{supp}_{C'_\bullet}(v_j^\varphi).$$

Proof. Recall that $\mathfrak{B}$ is a filtration compatible basis for $B_*(C'_\bullet)$, and $\mathfrak{Z}$ extends $\mathfrak{B}$ to one for $\varphi_\bullet(Z_*(C_\bullet))$. Now Corollary 7 together with the support equalities from Lemmas 15 and 16 yield the claim. ◀

▶ **Lemma 18.** $\mathrm{Pivots}\, R = \mathrm{Pivots}\, R^\varphi$.

Proof. The matrices D and $D^\varphi = DF^{-1}$ have the same column space. Matrix reduction does not change column spaces, so R and R^φ also have the same column space. In particular, every non-zero column of R is a non-trivial linear combination of non-zero columns of R^φ and vice versa. The pivots of a linear combination of a reduced set of column vectors must be the same as the pivot of one of these vectors, so we indeed obtain $\mathrm{Pivots}\, R = \mathrm{Pivots}\, R^\varphi$. ◀

We are now ready to prove the main result of this section.

Proof of Theorem 14. By definition of the induced map in homology, we have

$$\operatorname{im} H_*(\varphi_\bullet) \cong \frac{\varphi_\bullet(Z_*(C_\bullet))}{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)}.$$

The claim follows by applying Proposition 4 to the inclusion $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet) \subseteq B_*(C'_\bullet)$ with the filtration compatible bases $\mathfrak{B} \subseteq \mathfrak{Z}$, with supports as previously determined in Lemmas 16 and 17. Note that in the basis $\mathfrak{Z}$ we choose columns Fv_i with $i \notin \operatorname{Pivots} R^\varphi$, while the formula in Theorem 14 requires $i \notin \operatorname{Pivots} R$. These conditions are, however, equivalent by Lemma 18. ◀

3.2 Clearing

The *clearing* optimization [9] is a key ingredient of efficient persistence computation. We first recall the basic idea of clearing, which applies to the computation of persistent homology of a filtration of chain complexes $C_\bullet$ by reducing the boundary matrix D to $R = DV$. We keep the notation from the beginning of this section, and we assume that our filtration compatible basis $\mathfrak{C}$ is compatible with the homological grading in the sense that the restriction of this basis to each grading summand is again a basis of that summand. Our discussion focuses on chain complexes, but of course the findings naturally apply to *co*chain complexes with the appropriate adjustments to the grading.

If a column r_j of the reduced matrix R is nonzero, then necessarily $r_i = 0$ for $i = \operatorname{Pivot} r_j$. The homological degree of the i-th element of $\mathfrak{C}$ is one less than that of the j-th element. This leads to the clearing procedure: Instead of reducing D by column operations from left to right, we reduce columns in decreasing order of their homological degree (increasing in the case of cohomology). Before reducing the columns in dimension d, we set $r_j = 0$ for all j which appear as pivots of the already reduced columns in dimension $d + 1$.

Turning to the image setting, we also assume that the basis $\mathfrak{C}'$ and the map $\varphi_\bullet \colon C_\bullet \to C'_\bullet$ are compatible with the grading. Here, there is no direct analogue to the procedure outlined above, as the mixed basis boundary matrix D^φ fails to have the property described above; $r_j^\varphi \neq 0$ does not imply $r_i^\varphi = 0$ for $i = \operatorname{Pivot} r_j^\varphi$. In order to obtain a useful condition for columns of R^φ to be zero, we need to additionally consider a reduction $R' = D'V'$ of the boundary matrix $D' = FD^\varphi$.

▶ **Proposition 19.** *Let $R' = D'V'$ and $R^\varphi = D^\varphi V^\varphi$ be reduced. For all indices j we have $r_j^\varphi = 0$ if and only if $r'_j = 0$.*

Proof. First, note that $r_j^\varphi = 0$ if and only if $Fr_j^\varphi = 0$ because F is invertible. Moreover, FR^φ and R' have the same column space, since $FR^\varphi = R'(V')^{-1}V^\varphi$. Thus, the number of zero columns of R^φ is the same as the number of zero columns of R' since their ranks are equal and their non-zero columns are linearly independent. Now, it suffices to show that $r_j^\varphi = 0$ implies $r'_j = 0$, so assume $r_j^\varphi = 0$. Then $Fr_j^\varphi = 0$, but Fr_j^φ is also the same as the j-th column of $R'(V')^{-1}V^\varphi$. This is a linear combination of columns of R' with non-zero coefficient for r'_j since $(V')^{-1}V^\varphi$ is full-rank and upper-triangular. Non-zero columns of R' are linearly independent, so this linear combination can only be zero if $r'_j = 0$. ◀

In order to apply clearing to the reduction of D^φ, one can now reduce D' with clearing as usual, and clear the columns with the same indices in D^φ. Even more than that, one can not only clear the columns of D^φ whose index appears as a pivot in R', but rather *every* column with the same index as a zero column in R', meaning also those that have been reduced to zero via column operations on D'. Thus, with this optimization, the reduction of D^φ only establishes unique pivots among the non-zero columns, but no columns are reduced to zero.

▶ **Corollary 20.** *If D' has already been reduced to R', one can initialize the reduction R^φ of D^φ by setting $r_j^\varphi = 0$ for all j with $r_j' = 0$, and no further columns of R^φ will reduce to 0.*

3.3 Assembling Barcodes from (Co)homology Computations

Recalling our concrete setting of persistent homology for simplicial complexes, assume that we are given filtrations $L_\bullet$ and $K_\bullet$ of two isomorphic simplicial complexes $L \cong K$ and a monomorphism $f_\bullet \colon L_\bullet \to K_\bullet$, inducing an isomorphism $f \colon L \to K$. Following the notation from Example 13 and applying the previous results with $\varphi_\bullet = C_*(f_\bullet)$, we see that the barcode of $\operatorname{im} H_*(f_\bullet)$ can be determined via reductions of D^L and D^f and that the reduction of D^f may be performed with clearing if D^K has already been reduced before.

As known from the single filtration case, clearing requires a full persistence computation in the first homological degree for which persistence is computed. As persistence computations are often only feasible in low dimensions and practitioners are often only interested in barcodes in low degrees, it is much more powerful to apply clearing for *co*homological grading, allowing for the initialization to be performed in degree 0. Thus, our goal is to perform cohomological computations and still recover the image $\operatorname{im} H_*(f_\bullet)$ in homology.

As a first step towards that goal, we recall that $\operatorname{im} H^*(f_\bullet)$ and $\operatorname{im} H_*(f_\bullet)$ have the same barcodes [5]. However, the persistent cochain complex giving rise to persistent cohomology is not a filtration, so the basic matrix reduction algorithm does not directly apply there. Instead, we perform computations in the relative cohomology setting given by the map $H^*(f, f_\bullet)$. Its image no longer has the same barcode as $\operatorname{im} H_*(f_\bullet)$, but there are some correspondence results [5, Section 6.2], which we will summarize next. To state the result, for a barcode B we write $B^\dagger$ for the intervals in B that do not extend to any of the endpoints of our index set T and B^∞ for those intervals that do.

▶ **Proposition 21** (Bauer, Schmahl [5]). *For all degrees d, we have*

$$B(\operatorname{im} H_{d-1}(f_\bullet))^\dagger = B(\operatorname{im} H^d(f, f_\bullet))^\dagger,$$

and the map $I \mapsto T \setminus I$ defines bijections

$$B(\operatorname{im} H_d(f_\bullet))^\infty \leftrightarrow B(H^d(L, L_\bullet))^\infty \qquad and \qquad B(\operatorname{im} H^d(f, f_\bullet))^\infty \leftrightarrow B(H_d(K_\bullet))^\infty.$$

Note that none of the intervals in the barcodes considered here span the whole index set T, since we assume that our filtrations start with $L_0 = K_0 = \emptyset$.

Proposition 21 implies that in order to determine the barcode of $\operatorname{im} H_*(f_\bullet)$, it suffices to compute $B(H^*(L, L_\bullet))^\infty$ and $B(\operatorname{im} H^*(f, f_\bullet))^\dagger$. Following Example 10 and Theorem 12, we observe that $B(H^*(L, L_\bullet))^\infty$ may be determined from a reduction of the coboundary matrix $(D^L)^\perp$, and following Example 13 and Theorem 14, we know that $B(\operatorname{im} H^*(f, f_\bullet))^\dagger$ may be determined from a reduction of the coboundary matrix $(D^f)^\perp$. In the relative cohomology setting, the matrices $(D^L)^\perp$ and $(D^f)^\perp$ play the roles of D' and D^φ in the general setting, so by Corollary 20 we can simultaneously reduce these matrices with clearing.

We summarize the discussion in the following theorem. To simplify notation, we will assume that we are given funtions k and l on $K \cong L$ that induce the filtrations $K_\bullet$ and $L_\bullet$, respectively, via their sublevel set filtrations. For example, if $K_\bullet$ and $L_\bullet$ are Vietoris–Rips filtrations for different metrics on the same set of points, the functions l and k would be given by the corresponding diameter functions. Recall that the column and row indices of the matrices $(D^f)^\perp$ and $(D^L)^\perp$ correspond to the simplices of $K \cong L$ in different orders. We denote the column of a matrix X corresponding to a simplex σ by x_σ. Combining Theorem 14, Corollary 20, and Proposition 21, we can now determine barcodes from reductions of boundary matrices as follows.

▶ **Theorem 22.** *The matrices $(D^f)^\perp$ and $(D^L)^\perp$ can be reduced with clearing, and given reductions $S = (D^f)^\perp W$ and $R = (D^L)^\perp V$, the barcode of $\operatorname{im} H_*(f_\bullet)$ is the multiset*

$$\{[l(\sigma), k(\operatorname{Pivot} s_\sigma)) \neq \emptyset \mid s_\sigma \neq 0\} \cup \{[l(\tau), \infty) \mid r_\tau = 0 \text{ and } \tau \notin \operatorname{Pivots} R\}.$$

Recall that the column and row indices of the coboundary matrices indicated by $(-)^\perp$ correspond to the simplices of $K \cong L$ in reverse filtration order. Hence, the pivot simplex of a column vector appearing in the theorem will be the *first* simplex appearing in the filtration among those that correspond to a non-zero entry of the column, while for the usual boundary matrices D^L, D^K, etc., the pivot simplex of a column would be the one that appears *last* in the filtration.We summarize the algorithm resulting from Theorem 22 in pseudocode in Algorithm 1. To do so, we keep the notation from Section 3.3. In addition, for a column vector c, we write $\operatorname{PivotEntry} c$ to denote the entry of c at its pivot index.

■ **Algorithm 1** Algorithm to compute image persistence via two matrix reductions with clearing in cohomological grading.

Input: Filtration boundary matrix D^L with n columns, mixed basis boundary
        matrix D^f, maximum homological degree p for persistence to be computed
Result: Barcode of $\operatorname{im} H_*(f)$
$R \leftarrow (D^L)^\perp$; $S \leftarrow (D^f)^\perp$; $B \leftarrow \emptyset$
for $m = 0, \ldots, p$ **do**
    while $\exists \sigma <_L \tau$ *with* $r_\sigma \neq 0$, $\operatorname{Pivot} r_\sigma = \operatorname{Pivot} r_\tau$, *and* $\dim \sigma = m$ **do**
        $r_\tau \leftarrow r_\tau - \frac{\operatorname{PivotEntry} r_\tau}{\operatorname{PivotEntry} r_\sigma} r_\sigma$
    for σ *with* $\dim \sigma = m$ **do**
        if $r_\sigma = 0$ **then**
            $s_\sigma \leftarrow 0$
            $B \leftarrow B \sqcup \{[l(\sigma), \infty)\}$
        else if $\sigma \notin \operatorname{Pivots} R$ **then**
            $r_{\operatorname{Pivot} r_\sigma} \leftarrow s_{\operatorname{Pivot} r_\sigma} \leftarrow 0$
    while $\exists \sigma <_L \tau$ *with* $s_\sigma \neq 0$ *and* $\operatorname{Pivot} s_\sigma = \operatorname{Pivot} s_\tau$ **do**
        $s_\tau \leftarrow s_\tau - \frac{\operatorname{PivotEntry} s_\tau}{\operatorname{PivotEntry} s_\sigma} s_\sigma$
    for σ *with* $\dim \sigma = m$, $s_\sigma \neq 0$, *and* $l(\operatorname{Pivot} w_\sigma) < k(\operatorname{Pivot} s_\sigma)$ **do**
        $B \leftarrow B \sqcup \{[l(\sigma), k(\operatorname{Pivot} s_\sigma))\}$
return B

3.4 Apparent and Emergent Pairs in Image Matrix Reduction

An important optimization in persistence computation leading to significant computational improvements is given by utilizing the *apparent pairs* in the filtration, which are pairs (σ, τ) in the filtration such that σ is the latest facet of τ in the filtration and τ is the earliest cofacet of σ. Apparent pairs always form persistence pairs, since the corresponding columns are reduced already in the (co)boundary matrix. More generally, if (σ, τ) is a persistence pair and τ is the earliest cofacet of σ, we say that (σ, τ) is an *emergent cofacet pair*. The special case where such a pair (σ, τ) has persistence 0 can be identified in Ripser [1] during the construction of the columns of the coboundary matrix, terminating this construction early without constructing the entire column.

This strategy turns out to carry over to the image setting as well. The criterion used in Ripser for identifying the pivot index early is that its corresponding simplex appears in the filtration simultaneously with the simplex corresponding to the column. When reducing the

■ **Table 1** Running time and memory usage for image barcode and standard barcode (of the codomain filtration) for different data sets. The filtrations are defined by two different metrics on the point cloud. The maximum homological degree for persistence to be computed is specified by p, the number of points in the data set is specified by $|X|$.

Data Set	p	$|X|$	image barcode	standard barcode
S^2 intrinsic $\rightarrow$ extrinsic	2	128	0.56 s, 45 MB	0.26 s, 47 MB
		256	5.7 s, 315 MB	2.97 s, 316 MB
		512	155 s, 5.7 GB	65.6 s, 5.7 GB
SO(3) intrinsic $\rightarrow$ extrinsic	3	64	0.71 s, 51.7 MB	0.39 s, 52.7 MB
		128	13.7 s, 735 MB	7.3 s, 743 MB
		256	317 s, 13.1 GB	160 s, 13.1 GB
Möbius strip $\rightarrow \mathbb{R}P^2$	1	256	0.34 s, 24.0 MB	0.11 s, 25.0 MB
		512	2.66 s, 159 MB	0.73 s, 159 MB
		1024	25.6 s, 1.06 GB	7.21 s, 1.06 GB
$S^2 \rightarrow \mathbb{R}P^2$	2	32	0.37 s, 11.7 MB	0.00 s, 2.3 MB
		64	10.9 s, 27.1 MB	0.02 s, 7.5 MB
		128	574 s, 608 MB	0.24 s, 31.2 MB

mixed basis coboundary matrix $(D^f)^{\perp}$ for $f_{\bullet} \colon L_{\bullet} \rightarrow K_{\bullet}$, we apply the criterion with respect to the filtration $K_{\bullet}$, which determines the row order and hence the pivot of a column. Note that the apparent or emergent pairs (σ, τ) identified this way thus have the same filtration value for the filtration $K_{\bullet}$.

3.5   Computational Experiments

We provide an implementation [6] of the algorithm resulting from Theorem 22 including the clearing optimization, based on the `simple` branch of Ripser [1], for the special case where $L_{\bullet} = \mathrm{Rips}_{\bullet}(X, d)$ and $K_{\bullet} = \mathrm{Rips}_{\bullet}(X, d')$ are filtrations of Vietoris–Rips complexes corresponding to two metrics d and d' on a finite set X that satisfy $d(x, y) \geq d'(x, y)$ for all $x, y \in X$, with the map between filtrations given by the inclusions of L_t into K_t. Recall that the inequality $d \geq d'$ ensures that L_t is in fact a subcomplex of K_t. We did not include a comparison with Dionysus [19], as the general two-filtration setting considered in this paper is not supported. We further note that computation of image persistence is no longer supported in the current version of Dionysus.

Our computations were done on a notebook computer with an Apple M2 processor and 24 GB memory. The first example is given by X being $\{128, 256, 512\}$ points sampled uniformly from the unit sphere in $\mathbb{R}^3$, with the distance d being given by the geodesic distance on the sphere and the distance d' being given by the Euclidean distance in $\mathbb{R}^3$. The second example consists of $\{64, 128, 256\}$ points sampled uniformly at random from SO(3), with d given by the geodesic distance on SO(3) $\cong \mathbb{R}P^3$ and d' given by the Frobenius norm distance on $\mathbb{R}^{3 \times 3}$ (scaled by a factor of $1/\sqrt{2}$ to ensure that $d \geq d'$ holds). The third example is constructed by sampling $\{256, 512, 1024\}$ points uniformly from a cylinder with height π over a unit circle, equipped with the quotient metric that identifies antipodal points, resulting in a Möbius strip; the canonical map from the cylinder to the unit sphere given by $(\phi, \psi) \mapsto (\sin \phi \cos \psi, \cos \phi \cos \psi, \sin \psi)$ is nonexpanding, and it induces a nonexpanding map from the Möbius strip to the projective plane, both with the intrinsic metric. The fourth example is constructed by sampling $\{32, 64, 128\}$ points from the unit sphere, and considering the canonical quotient map to the projective plane. Running times and memory

usage are summarized in Table 1. Note that the examples differ significantly in terms of difficulty: while the first two examples comparing intrinsic and extrinsic metrics take only roughly twice as long as a standard persistence computation, the other two examples are more demanding, with the last one showing a huge difference in running time and memory usage. We attribute this to the vastly different total orders of simplices for the two filtrations.

References

1 Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *J. Appl. Comput. Topol.*, 5(3):391–423, 2021. `doi:10.1007/s41468-021-00071-5`.

2 Ulrich Bauer and Michael Lesnick. Induced matchings and the algebraic stability of persistence barcodes. *J. Comput. Geom.*, 6(2):162–191, 2015. `doi:10.20382/jocg.v6i2a9`.

3 Ulrich Bauer and Michael Lesnick. Persistence diagrams as diagrams: A categorification of the stability theorem. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule, editors, *Topological Data Analysis*, pages 67–96, Cham, 2020. Springer. `doi:10.1007/978-3-030-43408-3_3`.

4 Ulrich Bauer and Maximilian Schmahl. Efficient computation of image persistence. Preprint, 2022. `arXiv:2201.04170`.

5 Ulrich Bauer and Maximilian Schmahl. Lifespan functors and natural dualities in persistent homology. *Homology Homotopy Appl.*, 2023. To appear. `arXiv:2012.12881`.

6 Ulrich Bauer and Maximilian Schmahl. Ripser for image persistence, 2023. GitHub. URL: `https://github.com/Ripser/ripser/tree/image-persistence-simple`.

7 Michael Bleher, Lukas Hahn, Juan Angel Patino-Galindo, Mathieu Carriere, Ulrich Bauer, Raul Rabadan, and Andreas Ott. Topology identifies emerging adaptive mutations in sars-cov-2. Preprint, 2021. `arXiv:2106.07292`.

8 Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. SpringerBriefs in Mathematics. Springer, 2016. `doi:10.1007/978-3-319-42545-0`.

9 Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings of the 27th European Workshop on Computational Geometry*, 2011. URL: `https://eurocg11.inf.ethz.ch/abstracts/22.pdf`.

10 James Clough, Nicholas Byrne, Ilkay Oksuz, Veronika Zimmer, Julia Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE transactions on pattern analysis and machine intelligence*, PP, September 2020. `doi:10.1109/TPAMI.2020.3013679`.

11 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007. `doi:10.1007/s00454-006-1276-5`.

12 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, Philadelphia, PA, 2009. `doi:10.1137/1.9781611973068.110`.

13 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Computational geometry (SCG'06)*, pages 119–126. ACM, New York, 2006. `doi:10.1145/1137856.1137877`.

14 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *J. Algebra Appl.*, 14(5):1550066, 8, 2015. `doi:10.1142/S0219498815500668`.

15 Y. Dabaghian, F. Mémoli, L. Frank, and G. Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLOS Computational Biology*, 8(8):1–14, August 2012. `doi:10.1371/journal.pcbi.1002581`.

16 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, 17, 2011. `doi:10.1088/0266-5611/27/12/124003`.

17   Inés García-Redondo, Anthea Monod, and Anna Song. Fast topological signal identification and persistent cohomological cycle matching. Preprint, 2022. `arXiv:2209.15446`.

18   Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: `https://proceedings.neurips.cc/paper/2019/file/2d95666e2649fcfc6e3af75e09f5adb9-Paper.pdf`.

19   Dmitriy Morozov. Dionysus. URL: `https://mrzv.org/software/dionysus/`.

20   Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escolar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology*, 26(30):304001, 2015. `doi:10.1088/0957-4484/26/30/304001`.

21   Y. Reani and O. Bobrowski. Cycle registration in persistent homology with applications in topological bootstrap. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022. `doi:10.1109/TPAMI.2022.3217443`.

22   Nico Stucki, Johannes C. Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. To appear in the proceedings of ICML 2023. Preprint, 2022. `arXiv:2211.15272`.