Lines: 500

Cup Product Persistence and Its Efficient Computation

Tamal K. Dey ✉
Department of Computer Science, Purdue University, USA

Abhishek Rathod ✉
Department of Computer Science, Purdue University, USA

Abstract

It is well-known that the cohomology ring has a richer structure than homology groups. However, until recently, the use of cohomology in persistence setting has been limited to speeding up of barcode computations. Some of the recently introduced invariants, namely, persistent cup-length [12], persistent cup modules [13, 25] and persistent Steenrod modules [22], to some extent, fill this gap. When added to the standard persistence barcode, they lead to invariants that are more discriminative than the standard persistence barcode. In this work, we devise an $O(dn^4)$ algorithm for computing the persistent k-cup modules for all $k \in \{2, \dots, d\}$, where d denotes the dimension of the filtered complex, and n denotes its size. Moreover, we note that since the persistent cup length can be obtained as a byproduct of our computations, this leads to a faster algorithm for computing it. Finally, we introduce a new stable invariant called partition modules of cup product that is more discriminative than persistent k-cup modules and devise a fast time algorithm for computing it.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Algebraic topology

Keywords and phrases Persistent cohomology, cup product, image persistence, persistent cup module

1 Introduction

Persistent homology is one of the principal tools in the fast growing field of topological data analysis. A solid algebraic framework [29], a well-established theory of stability [8, 9] along with fast algorithms and software [1–3, 6, 23] to compute complete invariants called barcodes of filtrations have led to the successful adoption of single parameter persistent homology as a data analysis tool [16, 17]. This standard persistence framework operates in each (co)homology degree separately and thus cannot capture the interactions across degrees in an apparent way. To achieve this, one may endow a cohomology vector space with the well-known *cup product* forming a graded algebra. Then, the isomorphism type of such graded algebras can reveal information including interactions across degrees. However, even the best known algorithms for determining isomorphism of graded algebras run in exponential time in the worst case [7]. So it is not immediately clear how one may extract new (persistent) invariants from the product structure efficiently in practice.

Cohomology has already shown to be useful in speeding up persistence computations before [1, 2, 6]. It has also been noted that additional structures on cohomology provide an avenue to extract rich topological information [5, 12, 21, 22, 28]. To this end, in a recent study, the authors of [12] introduced the notion of (the persistent version of) an invariant called the *cup length*, which is the maximum number of cocycles with a nonzero product. In another version [13], the authors of [12] introduced an invariant called *barcodes of persistent k-cup modules* which are stable, and can add more discriminating ability (Figure 1). Computing this invariant allows us to capture interactions among various degrees. In Example 1, we provide simple examples for which persistent cup modules can disambiguate filtered spaces where ordinary persistence and persistent cup-length fail. Notice that for a filtered d-complex, the

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

k-cup modules for $k \in \{2, \ldots, d\}$ may not be a strictly finer invariant on its own compared to ordinary persistence. It can however add more information as Example 1 illustrates.

▶ **Example 1.** See Figure 1. Let K^1 be a cell complex obtained by taking a wedge of four circles and two 2-spheres. Let K^2 be a cell complex obtained by taking a wedge of two circles, a sphere and a 2-torus. Let K^3 be a cell complex obtained by taking a wedge of two tori.

▶ Remark 2. Throughout, for a cell complex C, the filtration for which all the k-dimensional cells of C arrive at the same index is referred to as the *natural cell filtration associated to* C.

Consider the natural cell filtrations $\mathsf{K}^1_\bullet$, $\mathsf{K}^2_\bullet$ and $\mathsf{K}^3_\bullet$. Standard persistence cannot tell apart $\mathsf{K}^1_\bullet$, $\mathsf{K}^2_\bullet$ and $\mathsf{K}^3_\bullet$ as the barcode for the three filtrations are the same. Persistent cup length cannot distinguish $\mathsf{K}^2_\bullet$ from $\mathsf{K}^3_\bullet$, whereas the barcodes for persistent cup modules for $\mathsf{K}^1_\bullet$, $\mathsf{K}^2_\bullet$ and $\mathsf{K}^3_\bullet$ are all different. See Example 19 in Appendix B for another example.

In Section 3 and 4, we show how to compute the persistent k-cup modules for all $k \in \{2, \ldots, d\}$ in $O(dn^4)$ time, where d denotes the dimension of the filtered complex, and n denotes its size. Moreover, since the persistent cup length of a filtration can be obtained as a byproduct of cup modules computation [12], we get an efficient algorithm to compute this invariant as well. Our approach for computing barcodes of persistent k-cup modules involves computing the image persistence of the cup product viewed as a map from the tensor product of the cohomology vector space to the cohomology vector space itself. This approach requires careful bookkeeping of restrictions of cocycles as one processes the simplices in the reverse filtration order. Algorithms for computing image persistence have been studied earlier by Cohen-Steiner et al. [11] and recently by Bauer and Schmahl [4]. However, the algorithms in [4, 11] work only for monomorphisms of filtrations making them inapplicable to our setting.

In Section 5, we introduce a new invariant called the partition modules of the cup product which is more discriminative than the k-cup modules. We observe that this invariant is stable for Rips and Čech filtrations (Appendix D), and we devise an algorithm that computes all the partition modules in $O(c(d)n^4)$ where $c(d)$ is subexponential in d as shown in Appendix C.

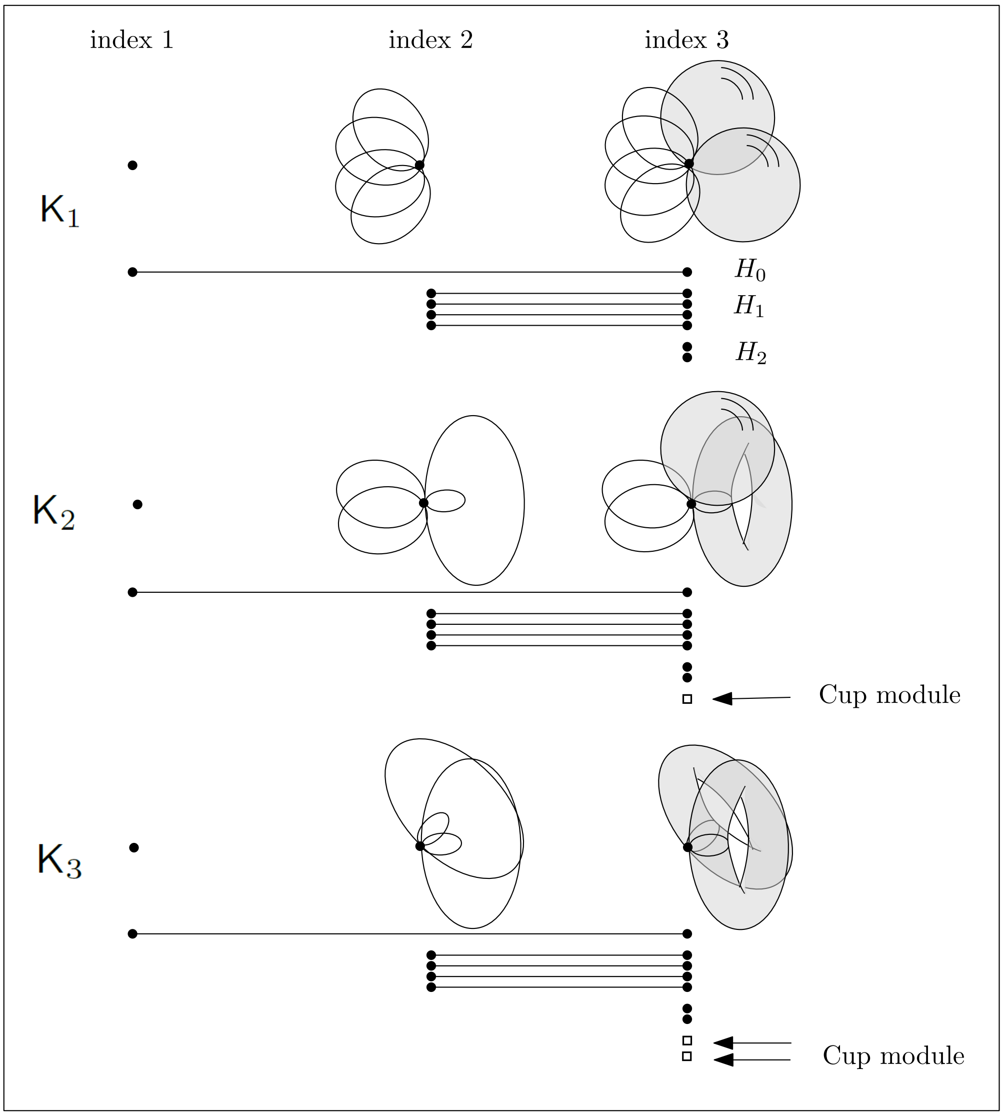
2 Background and preliminaries

Througout, we use n to denote the size of the filtered complex K, $[n]$ to denote the set $\{1, 2, \ldots, n\}$ and I to denote the set $\{0, 1, 2, \ldots, n\}$.

2.1 Persistent cohomology

In this paper, we work with mod-2 cohomology. We briefly recall some of the topological preliminaries in Appendix A. For an in-depth study, we refer the reader to [19, 20]. Let P denote a poset category such as $\mathbb{N}$, $\mathbb{Z}$, or $\mathbb{R}$, and **Simp** denote the category of simplicial complexes. A *P-indexed filtration* is a functor $\mathcal{F} : P \to \textbf{Simp}$ such that $\mathcal{F}_s \subseteq \mathcal{F}_t$ whenever $s \le t$. A P-indexed *persistence module* $V_\bullet$ is a functor from a poset category P to the category of (graded) vector spaces. The morphisms $\psi_{s,t} : V_s \to V_t$ for $s \le t$ are referred to as *structure maps*. We assume it to be of *finite type*, that is, $V_\bullet$ is pointwise finite dimensional and all morphisms $\psi_{s,t}$ for $s \le t$ are isomorphisms outside a finite subset of P. A P-indexed module W is a *submodule* of V if $W_s \subset V_s$ for all $s \in P$ and the structure maps $W_s \to W_t$ are restrictions of $\psi_{s,t}$ to W_s.

A persistence module $V_\bullet$ defined on a totally ordered set such as $\mathbb{N}$, $\mathbb{Z}$, or $\mathbb{R}$ decomposes uniquely up to isomorphism into simple modules called *interval modules* whose structure maps are identity and the vector spaces have dimension one. The support of these interval modules collectively constitute what is called the barcode of $V_\bullet$ and denoted by $B(V_\bullet)$.



66 **Figure 1** Example 1 Persistent cup modules distinguishes all three cellular filtrations.

When we have a filtration $\mathcal{F}$ on P where the complexes change only at a finite set of values $a_1 < a_2 < \ldots < a_n$, we can reindex the filtration with integers, and refine it so that only one simplex is added at every index. Reindexing and refining in this manner one can obtain a simplex-wise filtration of the final simplicial complex K defined on an indexing set with integers. For the remainder of the paper, we assume that the original filtration on P is simplex-wise to begin with. This only simplifies our presentation, and we do not lose generality. With this assumption, we obtain a filtration indexed on I after writing $\mathsf{K}_{a_i} = \mathsf{K}_i$,

$$\mathsf{K}_\bullet : \emptyset = \mathsf{K}_0 \hookrightarrow \mathsf{K}_1 \hookrightarrow \cdots \hookrightarrow \mathsf{K}_n = \mathsf{K}.$$

Applying the functor C^*, we obtain a persistence module $\mathsf{C}^*(\mathsf{K}_\bullet)$ of cochain complexes whose structure maps are cochain maps defined by restrictions induced by inclusions:

$$\mathsf{C}^*(\mathsf{K}_\bullet) : \mathsf{C}^*(\mathsf{K}_n) \to \mathsf{C}^*(\mathsf{K}_{n-1}) \to \cdots \to \mathsf{C}^*(\mathsf{K}_0),$$

and applying the functor H^*, we get a persistence module $\mathsf{H}^*(\mathsf{K}_\bullet)$ of graded cohomology vector spaces whose structure maps are linear maps induced by the above-mentioned restrictions:

$$\mathsf{H}^*(\mathsf{K}_\bullet) : \mathsf{H}^*(\mathsf{K}_n) \to \mathsf{H}^*(\mathsf{K}_{n-1}) \to \cdots \to \mathsf{H}^*(\mathsf{K}_0).$$

For simplifying the description of the algorithm, we work with I^{op}-indexed modules $\mathsf{H}^*(\mathsf{K}_\bullet)$ and $\mathsf{C}^*(\mathsf{K}_\bullet)$. The barcode $B(M)$ (see section 2.4) of a finite-type P^{op}-module M can be obtained from the barcode $B(N)$ of its associated I^{op}-module N by writing the interval $(j, i] \in B(N)$ for $j < i < n$ as $[a_{j+1}, a_{i+1}) \in B(M)$, and the interval $(j, n] \in B(N)$ as $[a_{j+1}, \infty) \in B(M)$. In this convention, we refer to i (or n) as a *birth index*, j as a *death index*, and intervals of the form $(j, n]$ as *essential bars*.

▶ **Definition 3** (Restriction of cocycles). *For a filtration $\mathsf{K}_\bullet$, if ζ is a cocycle in complex K_b, but ceases to be a cocycle at K_{b+1}, then ζ^i is defined as $\zeta^i = \zeta \cap \mathsf{C}^*(\mathsf{K}_i)$ for $i \leq b$, and in this case, we say that ζ^i is the restriction of ζ to index i. For $i > b$, ζ^i is set to the zero cocycle.*

▶ **Definition 4** (Persistent cohomology basis). *Let $\Omega_{\mathsf{K}} = \{\zeta_{\mathbf{i}} \mid \mathbf{i} \in B(\mathsf{H}^*(\mathsf{K}_\bullet))\}$ be a set of cocycles, where for every $\mathbf{i} = (d_i, b_i]$, $\zeta_{\mathbf{i}}$ is a cocycle in K_{b_i} but no more a cocycle in K_{b_i+1}. If for every index $j \in [n]$, the cocycle classes $\left\{[\zeta_{\mathbf{i}}^j] \mid \zeta_{\mathbf{i}} \in \Omega_{\mathsf{K}}\right\}$ form a basis for $\mathsf{H}^*(\mathsf{K}_j)$, then we say that Ω_{K} is a persistent cohomology basis for $\mathsf{K}_\bullet$, and the cocycle $\zeta_{\mathbf{i}}$ is called a representative cocycle for the interval $\mathbf{i}$. If $b_i = n$, $[\zeta_{\mathbf{i}}]$ is called an* essential class.

2.2 Simplicial cup product

Simplicial cup products connect cohomology groups across degrees. Let $\prec$ be an arbitrary but fixed total order on the vertex set of K. Let ξ and ζ be cocycles of degrees p and q respectively. The cup product of ξ and ζ is the $(p+q)$-cocycle $\xi \smile \zeta$ whose evaluation on any $(p+q)$-simplex $\sigma = \{v_0, \ldots, v_{p+q}\}$ is given by

$$(\xi \smile \zeta)(\sigma) = \xi(\{v_0, ..., v_p\}) \cdot \zeta(\{v_p, \ldots, v_{p+q}\}). \tag{1}$$

This defines a map $\smile : \mathsf{C}^p(\mathsf{K}) \times \mathsf{C}^q(\mathsf{K}) \to \mathsf{C}^{p+q}(\mathsf{K})$, which assembles to give a map $\smile : \mathsf{C}^*(\mathsf{K}) \times \mathsf{C}^*(\mathsf{K}) \to \mathsf{C}^*(\mathsf{K})$ for the cochain complex $\mathsf{C}^*(\mathsf{K})$. Using the fact that $\delta(\zeta \smile \xi) = \delta\xi \smile \zeta + \xi \smile \delta\zeta$, it follows that $\smile$ induces a map $\smile : \mathsf{H}^*(\mathsf{K}) \times \mathsf{H}^*(\mathsf{K}) \to \mathsf{H}^*(\mathsf{K})$. It can be shown that the map $\smile$ is independent of the ordering $\prec$.

Using the universal property for tensor products and linearity, the bilinear maps for

$$\smile : \mathsf{C}^p(\mathsf{K}) \times \mathsf{C}^q(\mathsf{K}) \to \mathsf{C}^{p+q}(\mathsf{K}) \quad \text{assemble to give a linear map} \quad \smile : \mathsf{C}^*(\mathsf{K}) \otimes \mathsf{C}^*(\mathsf{K}) \to \mathsf{C}^*(\mathsf{K}).$$

and the bilinear maps for

$$\smile\colon \mathsf{H}^p(\mathsf{K})\times\mathsf{H}^q(\mathsf{K}) \to \mathsf{H}^{p+q}(\mathsf{K}) \quad \text{assemble to give a linear map} \quad \smile\colon \mathsf{H}^*(\mathsf{K})\otimes\mathsf{H}^*(\mathsf{K}) \to \mathsf{H}^*(\mathsf{K}).$$

Finally, we state two well-known facts about cup products that are used throughout.

▶ **Theorem 5** (Commutativity [20])**.** *$[\xi] \smile [\zeta] = [\zeta] \smile [\xi]$ for all $[\xi], [\zeta] \in \mathsf{H}^*(\mathsf{K})$.*

▶ **Theorem 6** (Functoriality of the cup product [20])**.** *Let $f : \mathsf{K} \to \mathsf{L}$ be a simplicial map and let $f^* : \mathsf{H}^*(\mathsf{L}) \to \mathsf{H}^*(\mathsf{K})$ be the induced map on cohomology. Then, $f^*([\xi] \smile [\zeta]) = f^*([\xi]) \smile f^*([\zeta])$ for all $[\xi], [\zeta] \in \mathsf{H}^*(\mathsf{K})$.*

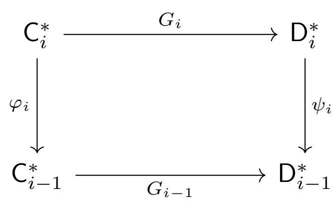
2.3 Image persistence

The category of persistence modules is abelian since the indexing category P is small and the category of vector spaces is abelian. Thus, kernels, cokernels, and direct sums are well-defined. Persistence modules obtained as images, kernels and cokernels of morphisms were first studied in [11]. In this section, we provide a brief overview of image persistence modules.

Let $\mathsf{C}_\bullet$ and $\mathsf{D}_\bullet$ be two persistence modules of cochain complexes:

$$\mathsf{C}^*_n \xrightarrow{\varphi_n} \mathsf{C}^*_{n-1} \xrightarrow{\varphi_{n-1}} \dots \xrightarrow{\varphi_1} \mathsf{C}^*_0 \qquad \textit{and} \qquad \mathsf{D}^*_n \xrightarrow{\psi_n} \mathsf{D}^*_{n-1} \xrightarrow{\psi_{n-1}} \dots \xrightarrow{\psi_1} \mathsf{D}^*_0,$$

such that for $0 \le i \le n$ the graded vector spaces C^*_i and D^*_i (along with the respective coboundary maps) are cochain complexes, and the structure maps $\{\varphi_i : \mathsf{C}^*_i \to \mathsf{C}^*_{i-1} \mid i \in [n]\}$ and $\{\psi_i : \mathsf{D}^*_i \to \mathsf{D}^*_{i-1} \mid i \in [n]\}$ are cochain maps. Let $G_\bullet : \mathsf{C}_\bullet \to \mathsf{D}_\bullet$ be a *morphism of persistence modules of cochain complexes*, that is, there exists a set of cochain maps $G_i : \mathsf{C}^*_i \to \mathsf{D}^*_i$ $\forall i \in \{0, \dots, n\}$, and the following diagram commutes for every $i \in [n]$.



Applying the cohomology functor H^* to the morphism $G_\bullet\colon \mathsf{C}_\bullet \to \mathsf{D}_\bullet$ induces another morphism of persistence modules, namely, $\mathsf{H}^*(G_\bullet)\colon \mathsf{H}^*(\mathsf{C}_\bullet) \to \mathsf{H}^*(\mathsf{D}_\bullet)$. Moreover, the image $\operatorname{im}\mathsf{H}^*(G_\bullet)$ is a persistence module. Like any other single-parameter persistence module, an image persistence module decomposes uniquely into intervals called its *barcode* [29].

As noted in [4], a natural strategy for computing the image of $\mathsf{H}^*(G_\bullet)$ is to write it as

$$\operatorname{im}\mathsf{H}^*(G_\bullet) \cong \frac{G_\bullet(\mathsf{Z}^*(\mathsf{C}_\bullet))}{G_\bullet(\mathsf{Z}^*(\mathsf{C}_\bullet)) \cap \mathsf{B}^*(\mathsf{D}_\bullet)},$$

where the i-th terms for the numerator and the denominator are given respectively by $(G_\bullet(\mathsf{Z}^*(\mathsf{C}_\bullet)))_i = G_i(\mathsf{Z}^*(\mathsf{C}_i))$ and $(G_\bullet(\mathsf{Z}^*(\mathsf{C}_\bullet)) \cap \mathsf{B}^*(\mathsf{D}_\bullet))_i = G_i(\mathsf{Z}^*(\mathsf{C}_i)) \cap \mathsf{B}^*(\mathsf{D}_i)$.

Tensor product image persistence. Consider the following map given by cup products

$$\smile_\bullet\colon \mathsf{C}^*(\mathsf{K}_\bullet) \otimes \mathsf{C}^*(\mathsf{K}_\bullet) \to \mathsf{C}^*(\mathsf{K}_\bullet). \tag{2}$$

Taking $G_\bullet = \smile_\bullet$ in the definition of image persistence, we get a persistence module, denoted by $\mathrm{im}\,\mathsf{H}^*(\smile \mathsf{K}_\bullet)$, which is the same as the persistent cup module introduced in [13]. Whenever the underlying filtered complex is clear from the context, we use the shorthand notation $\mathrm{im}\,\mathsf{H}^*(\smile_\bullet)$ instead of $\mathrm{im}\,\mathsf{H}^*(\smile \mathsf{K}_\bullet)$. Our aim is to compute its barcode denoted by $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$.

2.4 Barcodes

Let $\mathsf{K}_\bullet$ denote a filtration on the index set $I = \{0, 1, \ldots, n\}$. Assume that $\mathsf{K}_\bullet$ is simplex-wise, that is, $\mathsf{K}_i \setminus \mathsf{K}_{i-1}$ is a single simplex. Consider the persistence module $\mathsf{H}^*_\bullet$ obtained by applying the cohomology functor H^* on the filtration $\mathsf{K}_\bullet$, that is, $\mathsf{H}^*_i = \mathsf{H}^*(\mathsf{K}_i)$. The structure maps $\{\varphi^*_i : \mathsf{H}^*(\mathsf{K}_i) \to \mathsf{H}^*(\mathsf{K}_{i-1}) \mid i \in [n]\}$ for this module are induced by the cochain maps $\{\varphi_i : \mathsf{C}^*(\mathsf{K}_i) \to \mathsf{C}^*(\mathsf{K}_{i-1}) \mid i \in [n]\}$. Since $\mathsf{K}_\bullet$ is simplex-wise, each linear map φ^*_i is either injective with a cokernel of dimension one, or surjective with a kernel of dimension one, but not both. Such a persistence module $\mathsf{H}^*_\bullet$ decomposes into interval modules supported on a unique set of intervals, namely the barcode of $\mathsf{H}^*_\bullet$ written as $B(\mathsf{H}^*_\bullet) = \{(d_i, b_i] \,|\, b_i \geq d_i, b_i, d_i \in I\}$. Notice that since I is the indexing poset of $\mathsf{K}_\bullet$, I^{op} is the indexing poset of $\mathsf{H}^*_\bullet$. For $r > s$, we define $\varphi^*_{r,s} = \varphi^*_{s+1} \circ \cdots \circ \varphi^*_{r-1} \circ \varphi^*_r$ and $\varphi_{r,s} = \varphi_{s+1} \circ \cdots \circ \varphi_{r-1} \circ \varphi_r$.

▶ Remark 7. Since $\mathrm{im}\,\mathsf{H}^*(\smile_\bullet)$ is a submodule of $\mathsf{H}^*(\mathsf{K}_\bullet)$, the structure maps of $\mathrm{im}\,\mathsf{H}^*(\smile_\bullet)$ for every $i \in I$, namely, $\mathrm{im}\,\mathsf{H}^*(\smile_i) \to \mathrm{im}\,\mathsf{H}^*(\smile_i)$ are given by restrictions of φ^*_i to $\mathrm{im}\,\mathsf{H}^*(\smile_i)$.

▶ **Definition 8.** *For any $i \in \{0, \ldots, n\}$, a nontrivial cocycle $\zeta \in \mathsf{Z}^*(\mathsf{K}_i)$ is said to be a* product cocycle of K_i *if $[\zeta] \in \mathrm{im}\,\mathsf{H}^*(\smile_i)$.*

▶ **Proposition 9.** *For a filtration $\mathsf{K}_\bullet$, the birth indices of $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$ are a subset of the birth indices of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, and the death indices of $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$ are a subset of the death indices of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$.*

Proof. Let $(d_i, b_i]$ and $(d_j, b_j]$ be (not necessarily distinct) intervals in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, where $b_j \geq b_i$. Let ξ_i and ξ_j be representatives for $(d_i, b_i]$ and $(d_j, b_j]$ respectively. If $\xi_i \smile \xi_j^{b_i}$ is trivial, then by the functoriality of cup product, $\varphi_{b_i,r}(\xi_i \smile \xi_j^{b_i}) = \varphi_{b_i,r}(\xi_i) \smile \varphi_{b_i,r}(\xi_j^{b^i}) = \xi_i^r \smile \xi_j^r$ is trivial $\forall r < b_i$. Writing contrapositively, if $\exists r < b_i$ for which $\xi_i^r \smile \xi_j^r$ is nontrivial, then $\xi_i \smile \xi_j^{b_i}$ is nontrivial. Noting that $\mathrm{im}\,\mathsf{H}^*(\smile_\ell)$ for any $\ell \in \{0, \ldots, n\}$ is generated by $\{[\xi_i^\ell] \smile [\xi_j^\ell] \mid \xi_i, \xi_j \in \Omega_\mathsf{K}\}$, it follows that an index b is the birth index of a bar in $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$ only if it is the birth index of a bar in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, proving the first claim.

Let $\Omega'_{j+1} = \{[\tau_1], \ldots, [\tau_k]\}$ be a basis for $\mathrm{im}\,\mathsf{H}^*(\smile_{j+1})$. Then, Ω'_{j+1} extends to a basis Ω_{j+1} of $\mathsf{H}^*(\mathsf{K}_{j+1})$. If j is not a death index of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, then $\varphi_{j+1}(\tau_1), \ldots, \varphi_{j+1}(\tau_k)$ are all nontrivial and linearly independent. From Remark 7, it follows that j is not a death index of $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$, proving the second claim. ◀

▶ **Corollary 10.** *For a filtration $\mathsf{K}_\bullet$, if d is a death index of $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$, then at most one bar of $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$ has death index d.*

Proof. Using the fact that if the rank of a linear map $f : V_1 \to V_2$ is $\dim V_1 - 1$, then the rank of $f|_{W_1}$ for a subspace $W_1 \subset V_1$ is at least $\dim W_1 - 1$, from Remark 7 it follows that if $\dim \mathsf{H}^*(\mathsf{K}_d) = \dim \mathsf{H}^*(\mathsf{K}_{d+1}) - 1$, then

$$\dim(\mathrm{im}\,\mathsf{H}^*(\smile_d)) + 1 \geq \dim(\mathrm{im}\,\mathsf{H}^*(\smile_{d+1})) \geq \dim(\mathrm{im}\,\mathsf{H}^*(\smile_d)) \quad \text{proving the claim.}$$ ◀

▶ Remark 11. The persistent cup module is a submodule of the original persistence module. Let $\dim(\mathrm{im}\,\mathsf{H}^p_i)$ denote $\dim(\mathrm{im}\,\mathsf{H}^p(\smile_i))$. In the barcode $B(\mathrm{im}\,\mathsf{H}^*(\smile_\bullet))$, if $\mathsf{K}_i = \mathsf{K}_{i-1} \cup \{\sigma^p\}$, then either (i) $\dim(\mathrm{im}\,\mathsf{H}^p_i) > \dim(\mathrm{im}\,\mathsf{H}^p_{i-1})$, or (ii) $\dim(\mathrm{im}\,\mathsf{H}^{p-1}_i) < \dim(\mathrm{im}\,\mathsf{H}^{p-1}_{i-1})$, or (iii) there is no change: $\dim(\operatorname{im} \mathsf{H}_i^p) = \dim(\operatorname{im} \mathsf{H}_{i-1}^p)$ and $\dim(\operatorname{im} \mathsf{H}_i^{p-1}) = \dim(\operatorname{im} \mathsf{H}_{i-1}^{p-1})$. The decrease (increase) in persistent cup modules happens only if there is a decrease (increase) in ordinary cohomology. Multiple bars of $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet))$ may have the same birth index. But, if i is a death index, then Corollary 10 says that it is so for at most one bar in $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet))$.

3 Algorithm for the barcode of persistent cup module

Our goal is to compute the barcode of $\operatorname{im} \mathsf{H}^*(\smile_\bullet)$, which being an image module is a submodule of $\mathsf{H}^*(\mathsf{K}_\bullet)$. The vector space $\operatorname{im} \mathsf{H}^*(\smile_i)$ is a subspace of the cohomology vector space $\mathsf{H}^*(\mathsf{K}_i)$. Let us call this subspace the *cup space* of $\mathsf{H}^*(\mathsf{K}_i)$. Our algorithm keeps track of a basis of this cup space as it processes the filtration in the reverse order. This backward processing is needed because the structure maps between the cup spaces are induced by restrictions $\varphi_{j,i} \colon \mathsf{C}^*(\mathsf{K}_j) \to \mathsf{C}^*(\mathsf{K}_i)$ that are, in turn, induced by inclusions $\mathsf{K}_j \supseteq \mathsf{K}_i$, $i \leq j$. In particular, a cocycle/coboundary in K_j is taken to its restriction in K_i for $i \leq j$. Our algorithm keeps track of the birth and death of the cocycle classes in the cup spaces as it proceeds through the restrictions in the reverse filtration order. We maintain a basis of nontrivial product cocycles in a matrix $\mathbf{S}$ whose classes S form a basis for the cup spaces. In particular, cocycles in $\mathbf{S}$ are born and die with birth and death of the elements in cup spaces.

A cocycle class from $\mathsf{H}^*(\mathsf{K}_i)$ may enter the cup space $\operatorname{im} \mathsf{H}^*(\smile_i)$ signalling a birth or may leave (become zero) the cohomology vector space and hence the cup space signalling a death. Interestingly, multiple births may happen, meaning that multiple independent cocycle classes may enter the cup space, whereas at most a single class can die because of Corollary 10. To determine which class from the cohomology vector space enters the cup space and which one leaves it, we make use of the barcode of $\mathsf{H}^*(\mathsf{K}_\bullet)$. However, the classes of the bases maintained in $\mathbf{H}$ do not directly provide bases for the cup spaces. Hence, we need to compute and maintain $\mathbf{S}$ separately, of course, with the help of $\mathbf{H}$.

Let us consider the case of birth first. Suppose that a cocycle ξ at degree p is born at index $k = b_i$ for $\mathsf{H}^*(\mathsf{K}_\bullet)$. With ξ, a set of product cocycles are born in some of the degrees $p+q$ for $q \geq 1$. To detect them, we first compute a set of candidate cocycles by taking the cup product of cocycles $\xi \smile \zeta$, for all cocycles $\zeta \in \mathbf{H}$ at b_i which can potentially augment the basis maintained in $\mathbf{S}$. The ones among the candidate cocycles whose classes are independent w.r.t. the current basis maintained in $\mathbf{S}$ are determined to be born at b_i. Next, consider the case of death. A product cocycle ζ in degree r ceases to exist if it becomes linearly dependent of other product cocycles. This can happen only if the dimension of $\mathsf{H}^r(\mathsf{K}_\bullet)$ itself has reduced under the structure map going from $k+1$ to k. It suffices to check if any of the nontrivial cocycles in $\mathbf{S}$ have become linearly dependent or trivial after applying restrictions. In what follows, we use $\deg(\zeta)$ to denote the degree of a cocycle ζ.

Algorithm CupPers $(\mathsf{K}_\bullet)$

- Step 1. Compute barcode $B(\mathcal{F}) = \{(d_i, b_i]\}$ of $\mathsf{H}^*(\mathsf{K}_\bullet)$ with representative cocycles ξ_i; Let $\mathbf{H} = \{\xi_i \mid [\xi_i] \text{ essential and } \deg(\xi_i) > 0\}$; Initialize $\mathbf{S}$ with the coboundary matrix $\partial^\perp$ obtained by taking transpose of the boundary matrix ∂;
- Step 2. For $k := n$ to 1 do
  - Restrict the cocycles in $\mathbf{S}$ and $\mathbf{H}$ to index k;
  - Step 2.1 For every i with $k = b_i$ (k is a birth-index) and $\deg(\xi_i) > 0$
    - Step 2.1.1 If $k \neq n$, update $\mathbf{H} := [\mathbf{H} \mid \xi_i]$
    - Step 2.1.2 For every $\xi_j \in \mathbf{H}$

i. If $(\zeta \leftarrow \xi_i \smile \xi_j) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := k$ and $\zeta \cdot \text{rep@birth} := \zeta$

- Step 2.2 If $k = d_i$ (k is a death-index) for some i and $\deg(\xi_i) > 0$ then
  - Step 2.2.1 Reduce $\mathbf{S}$ with left-to-right column additions
  - Step 2.2.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(k, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
  - Step 2.2.3 Update $\mathbf{H}$ by removing the column ξ_i

Algorithm CupPers describes this algorithm with a pseudocode. First, in Step 1, we compute the barcode of the cohomology persistence module $\mathsf{H}^*(\mathsf{K}_\bullet)$ along with a persistent cohomology basis. This can be achieved in $O(n^3)$ time using either the annotation algorithm [6, 16] or the pCoH algorithm [15]. The basis H is maintained with the matrix $\mathbf{H}$ whose columns are cocycles represented as the support vectors on simplices. The matrix $\mathbf{H}$ is initialized with all cocycles ξ_i that are computed as representatives of the bars $(d_i, b_i]$ for the module $\mathsf{H}^*(\mathsf{K}_\bullet)$ which get born at the first (w.r.t. reverse order) complex $\mathsf{K}_n = \mathsf{K}$. The matrix $\mathbf{S}$ is initialized with the coboundary matrix $\partial^\perp$ with standard cochain basis. Subsequently, nontrivial cocycle vectors are added to $\mathbf{S}$. The classes of the nontrivial cocycles in matrix $\mathbf{S}$ form a basis S for the cup space at any point in the course of the algorithm.

In Step 2, we process cocycles in the reverse filtration order. At each index k, we do the following. If k is a birth index for a bar $(-, b_i]$ (Step 2.1), that is, $k = b_i$ for a bar with representative ξ_i in the barcode of $\mathsf{H}^*(\mathsf{K}_\bullet)$, first we augment $\mathbf{H}$ with ξ_i to keep it current as a basis for the vector space $\mathsf{H}^*(\mathsf{K}_k)$ (Step 2.2.1). Now, a new bar for the persistent cup module can potentially be born at k. To determine this, we take the cup product of ξ_i with all cocycles in $\mathbf{H}$ and check if the cup product cocycle is non-trivial and is independent of the cocycles in $\mathbf{S}$. If so, a product cocycle is born at k that is added to $\mathbf{S}$ (Step 2.1.2). To check this independence, we need $\mathbf{S}$ to have current coboundary basis along with current nontrivial product cocycle basis S that are both updated with restrictions. Note that we need a for loop in Step 2.1 because at $k = n$, there can be multiple births in $\mathsf{H}^*(\mathsf{K}_\bullet)$.

▶ Remark 12. Restrictions in $\mathbf{H}$ and $\mathbf{S}$ are implemented by zeroing out the corresponding row associated to the simplex σ_i when we go from K_i to K_{i-1} and $\mathsf{K}_i \setminus \mathsf{K}_{i-1} = \{\sigma_i\}$.

If k is a death index (Step 2.2), potentially the class of a product cocycle from $\mathbf{S}$ can be a linear combination of the classes of other product cocycles after $\mathbf{S}$ has been updated with restriction. We reduce $\mathbf{S}$ with left-to-right column additions and detect the column that is zeroed out (Step 2.2.1). If the column ζ is zeroed out, the class $[\zeta]$ dies at k and we generate a bar with death index k and birth index equal to the index when ζ was born (Step 2.2.2). Finally, we update $\mathbf{H}$ by removing the column for ξ_i (Step 2.2.3).

3.1 Rank functions and barcodes

Let $P \subseteq \mathbb{Z}$ be a finite set with induced poset structure from $\mathbb{Z}$. Let $\mathbf{Int}(P)$ denote the set of all intervals in P. Recall that P^{op} denotes the opposite poset category. Given a P^{op}-indexed persistence module $V_\bullet$, the rank function $\mathsf{rk}_{V_\bullet} : \mathbf{Int}(P) \to \mathbb{Z}$ assigns to each interval $I = [a, b] \in \mathbf{Int}(P)$ the rank of the linear map $V_b \to V_a$. It is well known that (see [10, 17]) the barcode of $V_\bullet$ viewed as a function $\mathsf{Dgm}_{V_\bullet} : \mathbf{Int}(P) \to \mathbb{Z}$ can be obtained from the rank function by the inclusion-exclusion formula:

$$\mathsf{Dgm}_{V_\bullet}([a, b]) = \mathsf{rk}_{V_\bullet}[a, b] - \mathsf{rk}_{V_\bullet}[a-1, b] + \mathsf{rk}_{V_\bullet}[a, b+1] - \mathsf{rk}_{V_\bullet}[a-1, b+1] \quad (3)$$

To prove the correctness of Algorithm CupPers, we use the following elementary fact.

▶ Fact 1. A class that is born at an index $\geq b$ dies at a iff $\mathsf{rk}_{V_\bullet}([a, b]) < \mathsf{rk}_{V_\bullet}([a+1, b])$.

3.2 Correctness of Algorithm CupPers

▶ **Theorem 13.** *Algorithm* CupPers *computes the barcode of the persistent cup module.*

Proof. In what follows, we abuse notation by denoting the restriction at index k of a cocycle ζ born at b also by the symbol ζ. That is, index-wise restrictions are always performed, but not always explicitly mentioned. We use $\{\xi_i\}$ to denote cocycles in the persistent cohomology basis computed in Step 1. The proof uses induction to show that for an arbitrary birth index b in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, if all bars for the persistent cup module with birth indices $b' > b$ are correctly computed, then the bars beginning with b are also correctly computed.

To begin with we note that in Algorithm CupPers, as a consequence of Proposition 9, we need to check if an index k is a birth (death) index of $B(\operatorname{im}\mathsf{H}^*(\smile_\bullet))$ only when it is a birth (death) index of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$. Also, from Corollary 10, we know that at most one cycle dies at a death index of $B(\operatorname{im}\mathsf{H}^*(\smile_\bullet))$ (justifying Step 2.2.2).

We now introduce some notation. In what follows, we denote the persistent cup module by $V_\bullet$. For a birth index b, let S_b be the cup space at index b. Let C_b be the vector space of the product cocycle classes created at index b. In particular, the classes in C_b are linearly independent of classes in S_{b+1}. For a birth index $b < n$, S_b can be written as a direct sum $S_b = S_{b+1} \oplus C_b$. For index n, we set $S_n = C_n$. Then, for a birth index $b \in \{0, \dots, n\}$, C_b is a subspace of $\mathsf{H}^*(\mathsf{K}_b)$. C_b can be written as:

$$C_b = \begin{cases} \langle [\xi_i] \smile [\xi_j] \mid \xi_i, \xi_j \text{ are essential cocycles of } \mathsf{H}^*(\mathsf{K}_\bullet)\rangle & \text{if } b = n \\ \langle [\xi_i] \smile [\xi_j] \mid \xi_i \text{ is born at } b, \text{ and } \xi_j \text{ is born at an index } \geq b\rangle & \text{if } b < n \end{cases}$$

For a birth index b, let $\mathbf{C}_b$ be the submatrix of $\mathbf{S}$ formed by representatives whose classes generate C_b, which augments $\mathbf{S}$ in Step 2.1.2 (i) when $k = b$ in the **for** loop. The cocycles in $\mathbf{C}_b$ are maintained for $k \in \{b, \dots, 1\}$ via subsequent restrictions to index k. Let $\mathbf{S}_b$ be the submatrix of $\mathbf{S}$ containing representative product cocycles that are born at index $\geq b$. Clearly, $\mathbf{C}_b$ is a submatrix of $\mathbf{S}_b$ for $b < n$, and $\mathbf{C}_n = \mathbf{S}_n$.

Let DP_b be the set of filtration indices for which the cocycles in $\mathbf{C}_b$ become successively linearly dependent to other cocycles in $\mathbf{S}_b$. That is, $d \in \mathrm{DP}_b$ if and only if there exists a cocycle ζ in $\mathbf{C}_b$ such that ζ is independent of all cocycles to its left in matrix $\mathbf{S}$ at index $d+1$, but ζ is either trivial or a linear combination of cocycles to its left at index d.

For the base case, we show that the death indices of the essential bars are correctly computed. First, we observe that for all $d \in \mathrm{DP}_n$, $\mathsf{rk}_{V_\bullet}([d, n]) = \mathsf{rk}_{V_\bullet}([d+1, n]) - 1$. Using Fact 1, it follows that the algorithm computes the correct barcode for $\operatorname{im}\mathsf{H}^*(\smile_\bullet)$ only if the indices in DP_n are the respective death indices for the essential bars. Since the leftmost columns of $\mathbf{S}$ are coboundaries from $\partial^\perp$ followed by cocycles from $\mathbf{C}_n$, and since we perform only left-to-right column additions in Step 2.2.1 to zero out cocycles in $\mathbf{C}_n$, the base case holds true. By (another) simple inductive argument, it follows that the computation of indices in DP_n does not depend on the specific ordering of representatives within $\mathbf{C}_n$.

Let $b < n$ be a birth index in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$. For induction hypothesis, assume that for every birth index $b' > b$ the indices in $\mathrm{DP}_{b'}$ are the respective death indices of the bars of $\operatorname{im}\mathsf{H}^*(\smile_\bullet)$ born at b'. By construction, the cocycles $\{\zeta_1, \zeta_2, \dots\}$ in $\mathbf{S}$ are sequentially arranged by the following rule: If ζ_i and ζ_j are two representative product cocycles in $\mathbf{S}$, then $i < j$ if the birth index b_i of the interval represented by ζ_i is greater than or equal to the birth index b_j of the interval represented by ζ_j. Then, as a consequence of the induction hypothesis, for a cocycle $\zeta \in \mathbf{C}_b \setminus \mathbf{S}_b$, we assign the correct birth index to the interval represented by ζ only if ζ can be written as a linear combination of cocycles to its left in matrix $\mathbf{S}$.

Now, suppose that at some index $d \in \mathrm{DP}_b$ we can write a cocycle ζ in submatrix $\mathbf{C}_b$ as a linear combination of cocycles to its left in $\mathbf{S}$. For such a $d \in \mathrm{DP}_b$, $\mathsf{rk}_{V_\bullet}([d,b]) = \mathsf{rk}_{V_\bullet}([d+1,b]) - 1$. Hence, using Fact 1, a birth index $\geq b$ must be paired with d.

However, since $\mathrm{DP}_b \cap \mathrm{DP}_{b'} = \emptyset$ for $b < b'$, it follows from the inductive hypothesis that the only birth index that can be paired to d is b. Moreover, since we take restrictions of cocycles in $\mathbf{S}$, all cocycles in $\mathbf{C}_b$ eventually become trivial or linearly dependent on cocycles to its left in $\mathbf{S}$. So, DP_b has the same cardinality as the number of cocycles in $\mathbf{C}_b$, and all the bars that are born at b must die at some index in DP_b. As a final remark, it is easy to check that the computation of indices in DP_b is independent of the specific ordering of representatives within $\mathbf{S}_b$ by a simple inductive argument. ◀

Time complexity of CupPers. Let the input simplex-wise filtration have n additions and hence the complex K have n simplices. Step 1 of CupPers can be executed in $O(n^3)$ time using algorithms in [6, 15]. The outer loop in Step 2 runs $O(n)$ times. For each death index in Step 2.2, we perform left-to-right column additions as done in the standard persistence algorithm to bring the matrix in reduced form. Hence, for each death index, Step 2.2 can be performed in $O(n^3)$ time. Since there are at most $O(n)$ death indices, the total cost for Step 2.2 in the course of the algorithm is $O(n^4)$.

Step 2.1 apparently incurs higher cost than Step 2.2. This is because at each birth point, we have to test the product of multiple pairs of cocycles stored in $\mathbf{H}$. However, we observe that there are at most $O(n^2)$ products of pairs of representative cocycles that are each computed and tested for linear independence at most once. In particular, if ξ_i and ξ_j represent $(d_i, b_i]$ and $(d_j, b_j]$ resp. with $b_i \leq b_j$, then $\xi_i \smile \xi_j$ is computed and tested for independence iff $b_i > d_j$ and the test happens at b_i. Using Equation (1), computing $\xi_i \smile \xi_j$ takes linear time. So the cost of computing the $O(n^2)$ products is $O(n^3)$. Moreover, since each independence test takes $O(n^2)$ time with the assumption that $\mathbf{S}$ is kept reduced all the time, Step 2.1 can be implemented to run in $O(n^4)$ time over the entire algorithm.

Finally, since restrictions of cocycles in $\mathbf{S}$ and $\mathbf{H}$ are computed by zeroing out corresponding rows, the total time to compute restrictions over the course of the algorithm is $O(n^2)$. Combining all costs, we get an $O(n^4)$ complexity bound for CupPers.

4 Algorithm for the barcode of persistent k-cup modules

While considering the *persistent* 2*-cup modules* (referred to as *persistent cup modules* in Section 3) is the natural first step, it must be noted that the invariants thus computed can still be enriched by considering *persistent k-cup modules*. As a next step, we consider image persistence of the k-fold tensor products.

Image persistence of k-fold tensor product. Consider image persistence of the map

$$\smile_\bullet^k \colon \mathsf{C}^*(\mathsf{K}_\bullet) \otimes \mathsf{C}^*(\mathsf{K}_\bullet) \otimes \cdots \otimes \mathsf{C}^*(\mathsf{K}_\bullet) \to \mathsf{C}^*(\mathsf{K}_\bullet) \tag{4}$$

where the tensor product is taken k times. Taking $G_\bullet = \smile_\bullet^k$ in the definition of image persistence, we get the module $\operatorname{im} \mathsf{H}^*(\smile_\bullet^k)$ which is same as the persistent k-cup module introduced in [13]. Our aim is to compute $B(\operatorname{im} \mathsf{H}^*(\smile^k \mathsf{K}_\bullet))$ (written as $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet^k))$ when the complex is clear from the context). Likewise, the degree-wise barcodes $B(\operatorname{im} \mathsf{H}^p(\smile_\bullet))$ and $B(\operatorname{im} \mathsf{H}^p(\smile_\bullet^k))$ can also be defined and computed. We omit the details for brevity.

▶ **Definition 14.** *For any $i \in \{0, \ldots, n\}$, a nontrivial cocycle $\zeta \in \mathsf{Z}^*(\mathsf{K}_i)$ is said to be an* order-k product cocycle of K_i *if* $[\zeta] \in \operatorname{im} \mathsf{H}^*(\smile_i^k)$.

4.1 Computing barcode of persistent k-cup modules

The order-k product cocycles can be viewed recursively as cup products of order-$(k-1)$ product cocycles with another cocycle. This suggests a recursive algorithm for computing the barcode of persistent k-cup module: compute the barcode of persistent $(k-1)$-cup module recursively and then use that to compute the barcode of persistent k-cup module just like the way we computed persistent 2-cup module using the bars for ordinary persistence. In the algorithm OrderkCupPers, we assume that the barcode with representatives for $\mathsf{H}^*(\mathsf{K}_\bullet)$ has been precomputed which is denoted by the pair of sets $(\{(d_{i,1}, b_{i,1}], \{\xi_{i,1}\})$. For simplicity, we assume that this pair is accessed by the recursive algorithm as a global variable and is not passed at each recursion level. At each recursion level k, the algorithm computes the barcode-representative pair denoted as $(\{(d_{i,k}, b_{i,k}], \{\xi_{i,k}\})$. Here, the cocycles $\xi_{i,k}$ are the initial cocycle representatives (before restrictions) for the bars $(d_{i,k}, b_{i,k}]$. At the time of their respective births $b_{i,k}$, they are stored in the field $\xi_{i,k} \cdot \text{rep@birth}$.

Algorithm OrderkCupPers $(\mathsf{K}_\bullet, k)$

- Step 1. If $k = 2$, return the barcode with representatives $\{(d_{i,2}, b_{i,2}], \xi_{i,2}\}$ computed by CupPers on $\mathsf{K}_\bullet$
  else $\{(d_{i,k-1}, b_{i,k-1}], \xi_{i,k-1}\} \leftarrow$ OrderkCupPers$(\mathsf{K}_\bullet,\ k-1)$
  Let $\mathbf{H} = \{\xi_{i,1} \mid [\xi_{i,1}]$ essential & $\deg(\xi_{i,1}) > 0\}$; $\mathbf{R} := \{\xi_{i,k-1} \mid b_{i,k-1} = n\}$; $\mathbf{S} := \partial^{\perp}$;
- Step 2. For $\ell := n$ to 1 do
  - Restrict the cocycles in $\mathbf{S}$, $\mathbf{R}$, and $\mathbf{H}$ to index ℓ;
  - Step 2.1 For every r s.t. $b_{r,1} = \ell \neq n$ (i.e., ℓ is a birth-index) and $\deg(\xi_{r,1}) > 0$
    - Step 2.1.1 Update $\mathbf{H} := [\mathbf{H} \mid \xi_{r,1}]$
    - Step 2.1.2 For every $\xi_{j,k-1} \in \mathbf{R}$
      i. If $(\zeta \leftarrow \xi_{r,1} \smile \xi_{j,k-1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.2 For all s such that $\ell = b_{s,k-1}$
    - Step 2.2.1 If $\ell \neq n$, update $\mathbf{R} := [\mathbf{R} \mid \xi_{s,k-1}]$
    - Step 2.2.2 For every $\xi_{i,1} \in \mathbf{H}$
      i. If $(\zeta \leftarrow \xi_{s,k-1} \smile \xi_{i,1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.3 If $\ell = d_{i,1}$ (i.e. ℓ is a death-index) and $\deg(\xi_{i,1}) > 0$ for some i then
    - Step 2.3.1 Reduce $\mathbf{S}$ with left-to-right column additions
    - Step 2.3.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(\ell, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
    - Step 2.3.3 Remove the column $\xi_{i,1}$ from $\mathbf{H}$
    - Step 2.3.4 Remove the column $\xi_{j,k-1}$ from $\mathbf{R}$ if $d_{j,k-1} = \ell$ for some j

A high-level pseudocode for computing the barcode of persistent k-cup module is given by algorithm OrderkCupPers. The algorithm calls itself recursively to generate the sets of bar-representative pairs for the persistent $(k-1)$-cup module. As in the case of persistent 2-cup modules, birth and death indices of order-k product cocycle classes are subsets of birth and death indices resp. of ordinary persistence. Thus, as before, at each birth index of the cohomology module, we check if the cup product of a representative cocycle (maintained in matrix $\mathbf{H}$) with a representative for persistent $(k-1)$-cup module (maintained in matrix $\mathbf{R}$) generates a new cocycle in the barcode for persistent k-cup module (Steps 2.1.2(i), 2.2.2(i)). If so, we note this birth with the resp. cocycle (by annotating the column) and add it to the matrix **S** that maintains a basis for live order-k product cocycles. At each death index, we check if an order-k product cocycle dies by checking if the matrix **S** loses a rank through restriction (Step 2.3.1). If so, the cocycle in **S** that becomes dependent to other cocycles through a matrix reduction is designated to be killed (Step 2.3.2) and we note the death of a bar in the k-cup module barcode. We update **H**, **R** appropriately (Steps 2.3.3, 2.3.4). At a high level, this algorithm is similar to CupPers with the role of **H** played by both **H** and **R** as they host the cocycles whose products are to be checked during the birth and the role of **S** in both algorithms remains the same, that is, check if a product cocycle dies or not.

Correctness and complexity of OrderkCupPers Correctness can be established the same way as for CupPers. See Appendix F for a sketch of the proof. For complexity, observe that we incur a cost from recursive calling in Step 1 and $O(n^4)$ cost from Step 2 with a similar analysis we did for CupPers while noting that there are once again a total of $O(n^2)$ product cocycles to be checked for independence at birth (Steps 2.1 and 2.2). Then, we get a recurrence for time complexity as $T(n, k) = T(n, k-1) + O(n^4)$ and $T(n, 2) = O(n^4)$ which solves to $T(n, k) = O(kn^4)$. Note that $k \leq d$, the dimension of K. This gives an $O(dn^4)$ algorithm for computing the barcodes of persistent k-cup modules for all $k \in \{2, \ldots, d\}$.

▶ Remark 15. In [25, Remark 4.18], a method to compute k-cup modules via the rank invariant is briefly sketched, but no complexity analysis is given. An obvious estimate for computing the d-cup module with the strategy mentioned in [25] would take $O(n^{d+5})$ time (generate $O(n^2)$ pairs (a, b), generate all possible candidate $O(n^d)$ tuples of live cocyles whose product at a is nonzero, and then $O(n^3)$ time to check if a generated tuple contributes to the basis at a). In contrast, our algorithm runs in $O(dn^4)$ time, which is substantially faster.

▶ Remark 16. In Sections 3 and 4, we devised algorithms to compute (absolute) persistent k-cup modules. The algorithms for computing *relative* persistent k-cup modules are minor variations (See Appendix G). Through Examples 35 and 36 in Appendix G, we also observe that unlike in the case of ordinary persistence [15], we do not have any duality that gives bijection of bars between barcodes of absolute and relative cup modules.

4.2 Faster computation of the persistent cup-length

The *cup length* of a ring is defined as the maximum number of multiplicands that together give a nonzero product in the ring. Let $\mathbf{Int}_*$ denote the set of all closed intervals of $\mathbb{R}$. Let $\mathcal{F}$ be an $\mathbb{R}$-indexed filtration of simplicial complexes. The *persistent cup-length function* $\mathbf{cuplength}_\bullet : \mathbf{Int}_* \to \mathbb{N}$ is defined as a function from the set of closed intervals to the set of non-negative integers, which assigns to each interval $[a, b]$, the cup-length of the image ring $\text{im}\left(\mathsf{H}^*(\mathsf{K})[a, b]\right)$, which is the ring $\text{im}\left(\mathsf{H}^*(\mathsf{K}_b) \to \mathsf{H}^*(\mathsf{K}_a)\right)$.

Given a P-indexed filtration $\mathcal{F}$ of a d-complex K of size n, let $V_\bullet^k$ denote its persistent k-cup module. Leveraging the fact that $\mathbf{cuplength}_\bullet([a, b]) = \text{argmax}\{k \mid \mathsf{rk}_{V_\bullet^k}([a, b]) \neq 0\}$ (see Proposition 5.9 in [13]), the algorithm described in Section 4 can be used to compute the persistent cup-length in $O(dn^4)$ time, whereas $O(n^{d+2})$ is a coarse estimate for the runtime of the algorithm described in [12]. Thus, for $d \geq 3$, our complexity bound for computing the persistent cup length is strictly better. We refer the reader to Appendix E for further details.

5 Partition modules of the cup product: a more refined invariant

A partition λ_q of an integer q is a multiset of integers that sum to q, written as $\lambda_q \vdash q$. That is, a multiset $\lambda_q = \{s_1, s_2, \ldots, s_\ell\}$ is a partition of q if $s_1 + s_2 + \cdots + \ldots s_\ell = q$. The integers $s_1, s_2, \dots, s_\ell$ are non-decreasing. For every partition λ_q of q, we define a submodule $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile \mathsf{K}_\bullet))$ (written as $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_\bullet))$ when K is clear from context) of $\operatorname{im} \mathsf{H}^q(\smile_\bullet^\ell))$:

$$\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_i)) = \langle [\alpha_1] \smile [\alpha_2] \smile \cdots \smile [\alpha_\ell] \mid [\alpha_j] \in \mathsf{H}^{s_j}(\mathsf{K}_i) \text{ for } j \in [\ell] \rangle.$$

The structure map $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_i)) \to \operatorname{im} \mathsf{H}^{\lambda_q}(\smile_{i-1}))$ is the restriction of φ_i^* to $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_i))$.

For an integer $q \geq 1$, let $\mathcal{P}(q)$ denote the number of partitions of q. In [14], Pribitkin proved that for $q \geq 1$, $\mathcal{P}(q) < \frac{e^{c\sqrt{q}}}{q^{\frac{3}{4}}}$, where $c = \pi\sqrt{2/3}$. For a d-complex K, let $\mathcal{P}^\uparrow(d)$ denote the total number of partition modules. Below, we obtain an upper bound for $\mathcal{P}^\uparrow(d)$.

$$\mathcal{P}^\uparrow(d) \ = \ \sum_{q=2}^{d} \mathcal{P}(q) \ < \ \sum_{q=2}^{d} \frac{e^{c\sqrt{q}}}{q^{\frac{3}{4}}} \ < \ d^{\frac{1}{4}} e^{c\sqrt{d}}.$$

When d is small, as is often the case in practice, $\mathcal{P}^\uparrow(d)$ is also small. For instance, $\mathcal{P}^\uparrow(2) = 1$, $\mathcal{P}^\uparrow(3) = 3$, $\mathcal{P}^\uparrow(4) = 7$.

Partition modules are more discriminative than persistent cup modules. From Remark 17 and Example 18, it follows that barcodes of partition modules are a strictly finer invariant compared to barcodes of cup modules.

▶ Remark 17. Given two filtrations $\mathsf{K}_\bullet$ and $\mathsf{L}_\bullet$, suppose that for some ℓ and q, $\operatorname{im} \mathsf{H}^q(\smile^\ell \mathsf{K}_\bullet))$ and $\operatorname{im} \mathsf{H}^q(\smile^\ell \mathsf{L}_\bullet))$ are distinct. Without loss of generality, there exists a bar $(d, b]$ in $B(\operatorname{im} \mathsf{H}^q(\smile \mathsf{K}_\bullet)))$ with no matching bar in $B(\operatorname{im} \mathsf{H}^q(\smile \mathsf{L}_\bullet)))$. Let ζ be a representative for the bar $(d, b]$. Then, $[\zeta]$ can be written as $[\zeta_1] \smile [\zeta_2] \smile \cdots \smile [\zeta_\ell]$ in K_b. Let s_i for each $i \in [\ell]$ denote the degree of cocycle class $[\zeta_i]$. Then, $\lambda_q = \{s_1, s_2, \dots, s_\ell\}$ is a partition of q. It follows that the bar $(d, b]$ will be present in $B(\operatorname{im} \mathsf{H}^{\lambda_q}(\smile \mathsf{K}_\bullet)))$ but not in $B(\operatorname{im} \mathsf{H}^{\lambda_q}(\smile \mathsf{L}_\bullet)))$.

▶ **Example 18.** Let $\mathsf{L}^1 = (S^3 \times S^1) \vee S^2 \vee S^2$ and $\mathsf{L}^2 = (S^2 \times S^2) \vee S^1 \vee S^3$. The natural cell filtrations $\mathsf{L}_\bullet^1$ and $\mathsf{L}_\bullet^2$ have isomorphic persistence modules and persistent cup modules. While $\mathsf{L}_\bullet^1$ has a nontrivial barcode for $\operatorname{im} \mathsf{H}^{(3,1)}$ and a trivial barcode for $\operatorname{im} \mathsf{H}^{(2,2)}$, the opposite is true for $\mathsf{L}_\bullet^2$. See Example 20 in Appendix B for details.

Partition modules are not a complete invariant. Let C^1 be the 3-torus, and $\mathsf{C}^2 = \mathbb{RP}^2 \vee \mathbb{RP}^2 \vee \mathbb{RP}^3$. The natural cell filtrations $\mathsf{C}_\bullet^1$ and $\mathsf{C}_\bullet^2$ have isomorphic persistence modules, isomorphic persistent cup modules as well as isomorphic partition modules. Yet, C^1 and C^2 have non-isomorphic cohomology algebras. See Example 21 in Appendix B for details.

The barcodes of all the partition modules of the cup product can be computed in $O(d^{\frac{1}{4}} e^{c\sqrt{d}} n^4)$ time, where $c = \pi\sqrt{2/3}$ time. The algorithm for computing them is described in Appendix C. In Appendix D, using functoriality of the cup product, we observe that partition modules are stable for Čech and Rips filtrations w.r.t. the interleaving distance.

6 Conclusion.

The cup product is a cohomology operation that gives the cohomology vector spaces the structure of a graded ring [19]. One could also use other operations such as Massey products and Steenrod squares [24, 26, 27]. Recently, Lupo et al. [22] introduced invariants called Steenrod barcodes and devised algorithms for their computation, which were implemented in the software `steenroder`. Our work complements the results in Lupo et al. [22], Contessoto et al. [12, 13] and Mémoli et al. [25]. While Contessoto et al. [13] introduced persistent k-cup modules invariant and established its stability, in this work, we devise an algorithm to compute it efficiently. We also introduce a more discriminative stable invariant called partition modules and provide an efficient algorithm to compute it. We believe that the combined advantages of a fast algorithm and favorable stability properties make cup modules and partition modules valuable additions to the topological data analysis pipeline.

References

1. Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.
2. Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Topological methods in data analysis and visualization III*, pages 103–117. Springer, 2014.
3. Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat – persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017.
4. Ulrich Bauer and Maximilian Schmahl. Efficient Computation of Image Persistence. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:14, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2023.14`.
5. Francisco Belchí and Anastasios Stefanou. A-infinity persistent homology estimates detailed topology from point cloud datasets. *Discrete & Computational Geometry*, pages 1–24, 2021.
6. Jean-Daniel Boissonnat, Tamal K Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In *European Symposium on Algorithms*, pages 695–706. Springer, 2013.
7. Peter Brooksbank, E O'Brien, and James Wilson. Testing isomorphism of graded algebras. *Transactions of the American Mathematical Society*, 372(11):8067–8090, 2019.
8. Frédéric Chazal, Vin De Silva, and Steve Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, 2014.
9. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005.
10. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, Jan 2007. `doi:10.1007/s00454-006-1276-5`.
11. 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.
12. Marco Contessoto, Facundo Mémoli, Anastasios Stefanou, and Ling Zhou. Persistent cup-length. In *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 31:1–31:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
13. Marco Contessoto, Facundo Mémoli, Anastasios Stefanou, and Ling Zhou. Persistent cup-length, 2021. URL: `https://arxiv.org/abs/2107.01553v3`, `doi:10.48550/ARXIV.2107.01553`.
14. Wladimir de Azevedo Pribitkin. Simple upper bounds for partition functions. *The Ramanujan Journal*, 18(1):113–119, 2009.
15. Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co) homology. *Inverse Problems*, 27(12):124003, 2011.
16. Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis.* Cambridge University Press, 2022.
17. Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction.* Applied Mathematics. American Mathematical Society, 2010.

18 David Eppstein. Python code for generating partitions. `https://code.activestate.com/recipes/218332/`. Accessed: 2023-11-30.

19 Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.

20 Jean-Claude Hausmann. *Mod two homology and cohomology*, volume 10. Springer, 2014.

21 Estanislao Herscovich. A higher homotopic extension of persistent (co)homology. *Journal of Homotopy and Related Structures*, 13(3):599–633, 2018.

22 Umberto Lupo, Anibal M. Medina-Mardones, and Guillaume Tauzin. Persistence Steenrod modules. *Journal of Applied and Computational Topology*, 6(4):475–502, 2022.

23 Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The Gudhi library: Simplicial complexes and persistent homology. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, pages 167–174, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

24 William S. Massey. Higher order linking numbers. *Journal of Knot Theory and Its Ramifications*, 7:393–414, 1998.

25 Facundo Mémoli, Anastasios Stefanou, and Ling Zhou. Persistent cup product structures and related invariants. *Journal of Applied and Computational Topology*, 2023.

26 Robert E. Mosher and Martin C. Tangora. *Cohomology operations and applications in homotopy theory*. Courier Corporation, 2008.

27 Norman E Steenrod. Products of cocycles and extensions of mappings. *Annals of Mathematics*, pages 290–320, 1947.

28 Andrew Yarmola. *Persistence and computation of the cup product*. Undergraduate honors thesis, Stanford University, 2010.

29 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356, 2004.

A Mod-2 (co)homology

Given a simplicial complex K, let $\mathsf{K}^{(p)}$ denote the set of p-simplices of K. A p-cochain of K is a function $\zeta : \mathsf{K}^{(p)} \to \mathbb{Z}_2$ with finite support. Equivalently, a p-cochain is a subset of $\mathsf{K}^{(p)}$. For any non-negative integer p, since the p-cochains can be added to each other with $\mathbb{Z}_2$ additions, they form a $\mathbb{Z}_2$-vector space called the *p-th cochain group*, denoted by $\mathsf{C}^p(\mathsf{K})$.

The *coboundary of a p-simplex* is a $(p+1)$-cochain that corresponds to the set of its $(p+1)$-cofaces. The coboundary map is linearly extended from p-simplices to p-cochains, where the *coboundary of a cochain* is the $\mathbb{Z}_2$-sum of the coboundaries of its elements. This extension is known as the *coboundary homomorphism*, and is denoted by $\delta_p : \mathsf{C}^p(\mathsf{K}) \to \mathsf{C}^{p+1}(\mathsf{K})$. A cochain $\zeta \in \mathsf{C}^p(\mathsf{K})$ is called a *p-cocycle* if $\delta_p \zeta = 0$, that is, $\zeta \in \ker \delta_p$. The collection of p-cocycles forms the *p-th cocycle group* of K, denoted by $\mathsf{Z}^p(\mathsf{K})$, which is also a vector space under $\mathbb{Z}_2$ addition. A cochain $\eta \in \mathsf{C}^p(\mathsf{K})$ is said to be a *p-coboundary* if $\eta = \delta_{p-1}\xi$ for some cochain $\xi \in \mathsf{C}^{p-1}(\mathsf{K})$, that is, $\eta \in \operatorname{im} \delta_{p-1}$. The collection of p-coboundaries forms the p-th coboundary group of K, denoted by $\mathsf{B}^p(\mathsf{K})$ which is also a vector space under $\mathbb{Z}_2$ addition. The three vector spaces are related as follows: $\mathsf{B}^p(\mathsf{K}) \subset \mathsf{Z}^p(\mathsf{K}) \subset \mathsf{C}^p(\mathsf{K})$. Therefore, we can define the quotient space $\mathsf{H}^p(\mathsf{K}) = \mathsf{Z}^p(\mathsf{K})/\mathsf{B}^p(\mathsf{K})$, which is called the p-th cohomology group of K. The elements of the vector space $\mathsf{H}^p(\mathsf{K})$, known as the p-th *cohomology group* of K, are equivalence classes of p-cocycles, where p-cocycles are equivalent if their $\mathbb{Z}_2$-difference is a p-coboundary. Equivalent cocycles are said to be *cohomologous*. For a p-cocycle ζ, its corresponding cohomology class is denoted by $[\zeta]$. The p-th *Betti number* of K, denoted by $\beta^p(\mathsf{K})$ is defined as $\beta^p(\mathsf{K}) = \dim \mathsf{H}^p(\mathsf{K})$. For a cocycle η and a simplex σ, the evaluation map $\langle \eta, \sigma \rangle$ is defined as follows: $\langle \eta, \sigma \rangle = 1$ if σ is in the support of η, and 0 otherwise.

A vector space V is said to be graded with an index set I if $V = \oplus_{i \in I} V_i$. Cochain and cohomology groups form graded vector spaces, where the grading is achieved with degree. Specifically, we work with graded cochain and cohomology vector spaces $\mathsf{C}^*(\mathsf{K}) = \bigoplus_{p \in \mathbb{N}} \mathsf{C}^p(\mathsf{K})$, and $\mathsf{H}^*(\mathsf{K}) = \bigoplus_{p \in \mathbb{N}} \mathsf{H}^p(\mathsf{K})$, respectively.

A *cochain complex* is a pair (C^*, δ) where C^* is a graded vector space and δ is a linear map satisfying $\delta(\mathsf{C}^p) \subset \mathsf{C}^{p+1}$ and $\delta \circ \delta = 0$. Observe that (C^*, δ) is graded in the increasing order of degrees. For instance, for a simplicial complex, the simplicial cochain groups along with the respective coboundary maps assemble to give a cochain complex.

Given two cochain complexes (C^*, δ_C) and (D^*, δ_D), a linear map $\psi : \mathsf{D}^* \to \mathsf{C}^*$ satisfying $\psi(\mathsf{D}^p) \subset \mathsf{C}^p$ for all p is a *cochain map* if $\psi \circ \delta_D = \delta_C \circ \psi$. For every $p \in \{0, 1, 2, \dots\}$, applying the cohomology functor H^p to a cochain complex (C^*, δ), gives its p-th cohomology group, which is the quotient space $\mathsf{H}^p(\mathsf{C}^*) = \frac{\ker(\delta : \mathsf{C}^p \to \mathsf{C}^{p+1})}{\operatorname{im}(\delta : \mathsf{C}^{p-1} \to \mathsf{C}^p)}$, and applying it to a cochain map $\psi : \mathsf{D}^* \to \mathsf{C}^*$ induces a linear map $\mathsf{H}^p(\psi) : \mathsf{H}^p(\mathsf{D}^*) \to \mathsf{H}^p(\mathsf{C}^*)$.

Let L be a subcomplex of a simplicial complex K. The couple (K, L) is called a *simplicial pair*. The *p-th relative cochain group* is given by $\mathsf{C}^p(\mathsf{K}, \mathsf{L}) = \operatorname{Hom}(\mathsf{C}_p(\mathsf{K}, \mathsf{L}), \mathbf{Z}_2)$. For every p, $\mathsf{C}^p(\mathsf{K}, \mathsf{L})$ can be viewed as a subgroup of $\mathsf{C}^p(\mathsf{K})$. The relative couboundary maps $\delta_p : \mathsf{C}^p(\mathsf{K}, \mathsf{L}) \to \mathsf{C}^{p+1}(\mathsf{K}, \mathsf{L})$ are obtained as restrictions of the absolute coboundary maps. Then, the *p-th relative cocycle group* $\mathsf{Z}^p(\mathsf{K}, \mathsf{L})$ and the *(p+1)-th relative coboundary group* $\mathsf{B}^p(\mathsf{K}, \mathsf{L})$ are respectively given by the kernel and the image of δ_p. Finally, the p-th cohomology group $\mathsf{H}^p(\mathsf{K}, \mathsf{L})$ is given by $\mathsf{H}^p(\mathsf{K}, \mathsf{L}) = \mathsf{Z}^p(\mathsf{K}, \mathsf{L})/\mathsf{B}^p(\mathsf{K}, \mathsf{L})$.

A.1 Tensor products of cochain complexes

Given two vector spaces U and V with basis B_U and B_V respectively, the tensor product $U \otimes V$ is the vector space with the set of all formal products $u \otimes v$, $u \in B_U$, $v \in B_V$, as a basis. One may view $u \otimes v$ as the function sending $(u, v) \in B_U \times B_V$ to 1 and all other elements to 0, and $U \otimes V$ as the space of all bilinear functions defined on $U \times V$. One may extend the definition of the tensor product to cochain complexes viewed as graded vector spaces. Given two cochain complexes A and B (whose respective coboundary maps are both denoted by δ), the *tensor product* $A \otimes B$ is the cochain complex whose degree-p group is

$$(A \otimes B)^p = \bigoplus_{i+j=p} A^i \otimes B^j,$$

where $A^i \otimes B^j$ is the tensor product of $\mathbb{Z}_2$-vector spaces, and whose coboundary map is given by the Leibniz rule (specialized to $\mathbb{Z}_2$-vector spaces).

$$\delta(a \otimes b) = \delta a \otimes b + a \otimes \delta b, \quad \text{where } a \text{ and } b \text{ are vectors in } A^i \text{ and } B^j\text{, respectively.}$$

B Additional examples

▶ **Example 19.** In this section, we provide an additional example that highlights the discriminating power of persistent cup modules.

Filtered real projective space. The real projective space $\mathbb{RP}^n$ is the space of lines through the origin in $\mathbb{R}^{n+1}$. It is homeomorphic to the quotient space $S^n/(u \simeq -u)$ obtained by identifying the antipodal points of a sphere, which in turn is homeomorphic to $D^n/(v \simeq -v)$ for $v \in \partial D^n$. Since $S^{n-1}/(u \simeq -u) \cong \mathbb{RP}^{n-1}$, $\mathbb{RP}^n$ can be obtained from $\mathbb{RP}^{n-1}$ by attaching a cell D^n using the projection $\wp_n : S^{n-1} \to \mathbb{RP}^{n-1}$. Thus, $\mathbb{RP}^n$ is a CW complex with one cell in every dimension from 0 to n. This gives rise to the natural cell filtration $\mathbb{RP}^n_\bullet$ for $\mathbb{RP}^n$, where cells of successively higher dimension are introduced with attaching maps $\wp_i$ for $i \in [n]$ described above. Finally, the cohomology algebra of $\mathbb{RP}^n$ is given by $\mathbb{Z}_2[x]/(x^{n+1})$, where $x \in \mathsf{H}^1(\mathbb{RP}^n)$ [20, pg. 146].

Filtered complex projective space. The complex projective space $\mathbb{CP}^n$ is the space of complex lines through the origin in $\mathbb{C}^{n+1}$. It is homeomorphic to the quotient space $S^{2n+1}/S^1 \cong S^{2n+1}/(u \simeq \lambda_q u)$, which in turn can be shown to be homeomorphic to $D^{2n}/(v \simeq \lambda_q v)$ for $v \in \partial D^{2n}$ for all $\lambda_q \in \mathbb{C}$, $|\lambda_q| = 1$. Therefore, $\mathbb{CP}^n$ is obtained from $\mathbb{CP}^{n-1}$ by attaching a $2n$-dimensional cell D^{2n} using the projection $\wp'_{2n} : S^{2n-1} \to \mathbb{CP}^{n-1}$. Thus, $\mathbb{CP}^n$ is a CW complex with one cell in every even dimension from 0 to $2n$. This yields the natural cell filtration $\mathbb{CP}^n_\bullet$ for $\mathbb{CP}^n$ where a cell of dimension $2i$ is added to the CW complex for $i \in [n]$ with the attaching maps $\wp'_{2i}$ for $i \in [n]$ described above. The cohomology algebra of $\mathbb{CP}^n$ is given by $\mathbb{Z}_2[y]/(y^{n+1})$, where $y \in \mathsf{H}^2(\mathbb{CP}^n)$ [20, pg. 241].

Filtered wedge of spheres. Let $\mathsf{L}^n = S^1 \vee \cdots \vee S^n$ be a wedge of spheres of increasing dimensions. Let p be the basepoint of L^n. The filtration $\mathsf{L}^n_\bullet$ can be described as follows: $\mathsf{L}^n_0 = p$, and for $i \in \{1, \dots, n\}$, $\mathsf{L}^n_i = S^1 \vee \cdots \vee S^i$, where for each index i, a cell of dimension i is added with the attaching map that takes the boundary of the i-cell to the basepoint p. The cohomology algebra of L^n is trivial in the sense that $x \smile y = 0$ for all $x, y \in \mathsf{H}^*(\mathsf{L})$.

Standard persistence cannot distinguish $\mathsf{L}^n_\bullet$ from $\mathbb{RP}^n_\bullet$ since they have the same standard persistence barcode. Persistent cup length for $\mathbb{RP}^n_\bullet$ and $\mathbb{CP}^n_\bullet$ for all intervals $[i, j]$ with $n \geq i \geq 1$ is equal to i, and hence persistent cup length cannot disambiguate these filtrations.

Finally, persistent cup modules can tell apart $\mathsf{L}^n_\bullet$, $\mathbb{RP}^n_\bullet$ and $\mathbb{CP}^n_\bullet$ as their cup module barcodes are different. This follows from the fact that the degrees of the generator of the cohomology algebras of $\mathbb{RP}^n_\bullet$ and $\mathbb{CP}^n_\bullet$ are different.

▶ **Example 20.** Let $\mathsf{L}^1 = (S^3 \times S^1) \vee S^2 \vee S^2$ and $\mathsf{L}^2 = (S^2 \times S^2) \vee S^1 \vee S^3$. The natural cell filtrations $\mathsf{L}^1_\bullet$ and $\mathsf{L}^2_\bullet$ have isomorphic persistence modules and persistent cup modules. While $\mathsf{L}^1_\bullet$ has a nontrivial barcode for $\operatorname{im}\mathsf{H}^{(3,1)}$ and a trivial barcode for $\operatorname{im}\mathsf{H}^{(2,2)}$, the opposite is true for $\mathsf{L}^2_\bullet$.

The barcodes for the persistence modules (using the convention from Section 2.1) are

$$\begin{aligned}
&B(\mathsf{H}^0(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^0(\mathsf{L}^2_\bullet)) = \{(-1,4]\},\\
&B(\mathsf{H}^1(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^1(\mathsf{L}^2_\bullet)) = \{(0,4]\},\\
&B(\mathsf{H}^2(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^2(\mathsf{L}^2_\bullet)) = \{(1,4],(1,4]\}\\
&B(\mathsf{H}^3(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^3(\mathsf{L}^2_\bullet)) = \{(2,4]\} \text{ and}\\
&B(\mathsf{H}^4(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^4(\mathsf{L}^2_\bullet)) = \{(3,4]\}.
\end{aligned}$$

For the persistent cup modules, $B(\operatorname{im}\mathsf{H}^4(\smile \mathsf{L}^1_\bullet)) = B(\operatorname{im}\mathsf{H}^4(\smile \mathsf{L}^2_\bullet)) = \{(3,4]\}$. For other degrees, the persistent cup modules are trivial.

Finally, for partition modules $B(\operatorname{im}\mathsf{H}^{(2,2)}(\smile \mathsf{L}^2_\bullet)) = \{(3,4]\}$ and $B(\operatorname{im}\mathsf{H}^{(2,2)}(\smile \mathsf{L}^1_\bullet))$ is empty, while $B(\operatorname{im}\mathsf{H}^{(3,1)}(\smile \mathsf{L}^2_\bullet))$ is empty and $B(\operatorname{im}\mathsf{H}^{(3,1)}(\smile \mathsf{L}^1_\bullet)) = \{(3,4]\}$.

▶ **Example 21.** Let C^1 be the 3-torus, and $\mathsf{C}^2 = \mathbb{RP}^2 \vee \mathbb{RP}^2 \vee \mathbb{RP}^3$. The natural cell filtrations $\mathsf{C}^1_\bullet$ and $\mathsf{C}^2_\bullet$ have isomorphic persistence modules, isomorphic persistent cup modules as well as isomorphic partition modules. Yet, C^1 and C^2 have non-isomorphic cohomology algebras.

The barcodes for the persistence modules are

$$\begin{aligned}
&B(\mathsf{H}^0(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^0(\mathsf{L}^2_\bullet)) = \{(-1,3]\},\\
&B(\mathsf{H}^1(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^1(\mathsf{L}^2_\bullet)) = \{(0,3],(0,3],(0,3]\},\\
&B(\mathsf{H}^2(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^2(\mathsf{L}^2_\bullet)) = \{(1,3],(1,3],(1,3]\} \text{ and}\\
&B(\mathsf{H}^3(\mathsf{L}^1_\bullet)) = B(\mathsf{H}^3(\mathsf{L}^2_\bullet)) = \{(2,3]\}.
\end{aligned}$$

The barcodes for the persistence cup modules are

$$\begin{aligned}
&B(\operatorname{im}\mathsf{H}^2(\smile \mathsf{L}^1_\bullet)) = B(\operatorname{im}\mathsf{H}^2(\smile \mathsf{L}^2_\bullet)) = \{(1,3],(1,3],(1,3]\} \text{ and}\\
&B(\operatorname{im}\mathsf{H}^3(\smile \mathsf{L}^1_\bullet)) = B(\operatorname{im}\mathsf{H}^3(\smile \mathsf{L}^2_\bullet)) = \{(2,3]\}.
\end{aligned}$$

The barcodes for the partition modules are

$$\begin{aligned}
&B(\operatorname{im}\mathsf{H}^{(1,1)}(\smile \mathsf{L}^1_\bullet)) \;\; = B(\operatorname{im}\mathsf{H}^{(1,1)}(\smile \mathsf{L}^2_\bullet)) \;\; = \{(1,3],(1,3],(1,3]\},\\
&B(\operatorname{im}\mathsf{H}^{(2,1)}(\smile \mathsf{L}^1_\bullet)) \;\; = B(\operatorname{im}\mathsf{H}^{(2,1)}(\smile \mathsf{L}^2_\bullet)) \;\; = \{(2,3]\} \text{ and}\\
&B(\operatorname{im}\mathsf{H}^{(1,1,1)}(\smile \mathsf{L}^1_\bullet)) = B(\operatorname{im}\mathsf{H}^{(1,1,1)}(\smile \mathsf{L}^2_\bullet)) = \{(2,3]\}.
\end{aligned}$$

The cohomology algebra $\mathsf{H}^*(\mathsf{C}^1) \approx \mathbb{Z}_2[a,b,c]/(a^2,b^2,c^2)$. Note that $\mathsf{H}^*(\mathbb{RP}^2) \approx \mathbb{Z}_2[a]/(a^3)$ and $\mathsf{H}^*(\mathbb{RP}^3) \approx \mathbb{Z}_2[a]/(a^4)$. Let $\mathsf{H}^>$ denote the positive parts of H^*. Then, the cohomology algebra of C^2 is $\mathsf{H}^*(\mathsf{C}^2) \approx \mathbb{Z}_2\mathbf{1} \oplus \mathsf{H}^>(\mathbb{RP}^2) \oplus \mathsf{H}^>(\mathbb{RP}^2) \oplus \mathsf{H}^>(\mathbb{RP}^3)$.

Unlike $\mathsf{H}^*(\mathsf{C}^2)$, there does not exist a cocycle x in the algebra $\mathsf{H}^*(\mathsf{C}^1)$ such that x^3 is nonzero. Hence $\mathsf{H}^*(\mathsf{C}^1)$ and $\mathsf{H}^*(\mathsf{C}^2)$ are non-isomorphic.

C Algorithm for computing partitions modules of the cup product

Algorithm CupPers2Parts describes an algorithm for computing the barcode of the module $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile_\bullet))$ for $\lambda_q \vdash q$ when $|\lambda_q| = 2$. First, in Step 0, we need to check if the barcode for the partition $\lambda_q = \{s_1, s_2\}$ has already been computed because CupPers2Parts is called from ExtendCupPersKParts possibly multiple times with the same argument λ_q. In Step 1, we compute the barcode of the cohomology persistence module $\mathsf{H}^*(\mathsf{K}_\bullet)$ along with a persistent cohomology basis. As in CupPers2Parts, a basis is maintained with the matrix $\mathbf{H}$ whose columns are (restricted) representative cocycles. The matrix $\mathbf{H}$ is initialized with essential cocycles. The matrix $\mathbf{S}$ is initialized with the coboundary matrix $\partial^\perp$ with standard cochain basis. Subsequently, nontrivial cocycle vectors are added to $\mathbf{S}$. For every k, the classes of the nontrivial cocycles in matrix $\mathbf{S}$ form a basis for $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_k))$. In particular, a cocycle $\zeta = \xi_1 \cup \xi_2$ is added to S only if $\deg(\xi_1) = s_1$ and $\deg(\xi_2) = s_2$ or vice versa. Other than the details mentioned here, CupPers2Parts is identical to CupPers.

Algorithm CupPers2Parts $(\mathsf{K}_\bullet, \lambda_q)$

- Step 0. If the barcode for the partition $\lambda_q = \{s_1, s_2\}$ has already been computed, then return the barcode with representatives $\{(d_{i,2}, b_{i,2}], \xi_{i,2}\}$.
- Step 1. Compute barcode $B(\mathcal{F}) = \{(d_i, b_i]\}$ of $\mathsf{H}^*(\mathsf{K}_\bullet)$ with representative cocycles ξ_i; Let $\mathbf{H} = \{\xi_i \mid [\xi_i] \text{ essential}\}$; Initialize $\mathbf{S}$ with the coboundary matrix $\partial^\perp$ obtained by taking transpose of the boundary matrix ∂;
- Step 2. For $k := n$ to 1 do
  - Restrict the cocycles in $\mathbf{S}$ and $\mathbf{H}$ to index k;
  - Step 2.1 For every i s.t. $k = b_i$ (k is a birth-index)
    - Step 2.1.1 If $k \neq n$, update $\mathbf{H} := [\mathbf{H} \mid \xi_i]$
    - Step 2.1.2 If $\deg(\xi_i) = s_1$
      1. Step 2.1.2.1 For every $\xi_j \in \mathbf{H}$ with $\deg(\xi_j) = s_2$
         i. If $(\zeta \leftarrow \xi_i \smile \xi_j) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := k$ and $\zeta \cdot \text{rep@birth} := \zeta$
    - Step 2.2.2 If $\deg(\xi_i) = s_2$ and $s_1 \neq s_2$
      1. Step 2.2.2.1 For every $\xi_j \in \mathbf{H}$ with $\deg(\xi_j) = s_1$
         i. If $(\zeta \leftarrow \xi_i \smile \xi_j) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := k$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.2 If $k = d_i$ for some i then (k is a death-index)
    - Step 2.2.1 Reduce $\mathbf{S}$ with left-to-right column additions
    - Step 2.2.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(k, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
    - Step 2.2.3 Update $\mathbf{H}$ by removing the column ξ_i

▶ **Definition 22** (Refinement of a partition). *Let λ_q and λ'_q be partitions of q. We say λ_q* refines *λ'_q if the parts of λ'_q can be subdivided to produce the parts of λ_q.*

For example, $(1,1,1,1) \vdash 4$ and $(1,2,1) \vdash 4$ and $(1,1,1,1)$ is a refinement of $(1,2,1)$.

▶ Remark 23. If a partition λ_q is a refinement of a partition λ'_q, then $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_\bullet))$ is a submodule of $\operatorname{im} \mathsf{H}^{\lambda'_q}(\smile_\bullet))$.

▶ **Definition 24** (Extension of a partition). *Let p and q be integers, with $q > p$. Let $\lambda_q = (s_1, s_2, \ldots, s_m)$ be a partition of q and $\lambda_p = (s'_1, s'_2, \ldots, s'_\ell)$ be a partition of p for some integers ℓ and m, with $m > \ell$. We say λ_q* extends *λ_p if $s_i = s'_i$ for $i \in [\ell]$. We say that λ_q* extends *λ_p* by one *if $|\lambda_q| = |\lambda_p| + 1$.*

For example, $(2,2) \vdash 4$ and $(2,2,3) \vdash 5$, and $(2,2,3)$ *extends* $(2,2)$ *by one.*

Algorithm ExtendCupPersKParts describes an algorithm for computing the barcode of the module $\operatorname{im} \mathsf{H}^{\lambda_t}(\smile_\bullet)$ for $\lambda_t \vdash t$. In Step 0, we check if the barcode for the partition λ_t has already been computed because ExtendCupPersKParts is called recursively from ExtendCupPersKParts possibly multiple times with the same argument λ_t. In Step 1, we first check if $|\lambda_t| = 2$, in which case, we invoke CupPers2Parts and return. Otherwise, $|\lambda_t| = k > 2$, and the algorithm calls itself recursively to generate the sets of bar-representative pairs for the module $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_\bullet)$), where λ_t is a partition that extends λ_q by one. As in the case of OrderkCupPers, the birth and death indices of order-k product cocycle classes are subsets of birth and death indices resp. of ordinary persistence. Therefore, at each birth index of the cohomology module, we check if the cup product of a representative cocycle with degree $t-q$ (maintained in matrix $\mathbf{H}$) with a representative for $\operatorname{im} \mathsf{H}^{\lambda_q}(\smile_\bullet)$) (which has degree q and is maintained in matrix $\mathbf{R}$) generates a new cocycle in the barcode for $\operatorname{im} \mathsf{H}^{\lambda_t}(\smile_\bullet)$) (Steps 2.1.2(i), 2.2.2(i)). If so, we note this birth with the resp. cocycle (by annotating the column) and add it to the matrix $\mathbf{S}$ that maintains a basis for live order-k product cocycles whose respective degrees form a partition λ_t of t.The case of death (Step 2.3) is identical to OrderkCupPers.

Algorithm ExtendCupPersKParts $(\mathsf{K}_\bullet, \lambda_t)$

- Step 0. If the barcode for the partition λ_t has already been computed, then return the barcode with representatives $\{(d_{i,k}, b_{i,k}], \xi_{i,k}\}$. Else, let λ_q be any partition such that λ_t extends λ_q by one, and let $k = |\lambda_t|$.
- Step 1. If $|\lambda_t| = 2$, return the barcode with representatives $\{(d_{i,2}, b_{i,2}], \xi_{i,2}\}$ computed by CupPers2Parts$(\mathsf{K}_\bullet, \lambda_t)$
  Set $\{(d_{i,k-1}, b_{i,k-1}], \xi_{i,k-1}\} \leftarrow$ ExtendCupPersKParts$(\mathsf{K}_\bullet, \lambda_q)$
  Let $\mathbf{H} = \{\xi_{i,1} \mid [\xi_{i,1}]$ essential and $\deg(\xi_{i,1}) = t - q\}$; $\mathbf{R} := \{\xi_{i,k-1} \mid b_{i,k-1} = n\}$; $\mathbf{S} := \partial^\perp$;
- Step 2. For $\ell := n$ to 1 do
  - Restrict the cocycles in $\mathbf{S}$, $\mathbf{R}$, and $\mathbf{H}$ to index ℓ;
  - Step 2.1 For every r s.t. $b_{r,1} = \ell \neq n$ (i.e., ℓ is a birth-index) and $\deg(\xi_{r,1}) = t - q$
    - Step 2.1.1 Update $\mathbf{H} := [\mathbf{H} \mid \xi_{r,1}]$
    - Step 2.1.2 For every $\xi_{j,k-1} \in \mathbf{R}$
      i. If $(\zeta \leftarrow \xi_{r,1} \smile \xi_{j,k-1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.2 For all s such that $\ell = b_{s,k-1}$
    - Step 2.2.1 If $\ell \neq n$, update $\mathbf{R} := [\mathbf{R} \mid \xi_{s,k-1}]$
    - Step 2.2.2 For every $\xi_{i,1} \in \mathbf{H}$
      i. If $(\zeta \leftarrow \xi_{s,k-1} \smile \xi_{i,1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.3 If $\ell = d_{i,1}$ (i.e. ℓ is a death-index) then
    - Step 2.3.1 Reduce $\mathbf{S}$ with left-to-right column additions
    - Step 2.3.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(\ell, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
    - Step 2.3.3 Remove the column $\xi_{i,1}$ from $\mathbf{H}$
    - Step 2.3.4 Remove the column $\xi_{j,k-1}$ from $\mathbf{R}$ if $d_{j,k-1} = \ell$ for some j

For every $k \in \{2, \ldots, d\}$, **Algorithm** ComputePartitionBarcodes first generates all partitions of integer k, and then for every partition λ_k of k computes the barcode of the partition module $\operatorname{im} \mathsf{H}^{\lambda_k}(\smile_\bullet)$).

Algorithm ComputePartitionBarcodes ($\mathsf{K}_\bullet$)

- Step 1. For $k := 2$ to d do
  - Step 1.1 Compute the set of partitions of k. Denote it by Λ_k.
  - Step 1.2 For every partition $\lambda_k \in \Lambda_k$ do
    - Step 1.2.1 $\{(d_{i,|\lambda_k|}, b_{i,|\lambda_k|}], \xi_{i,|\lambda_k|}\} \leftarrow$ ExtendCupPersKParts($\mathsf{K}_\bullet$,λ_k).

Correctness and complexity. The correctness proofs for CupPers and ExtendCupPersKParts are identical to those of CupPers2Parts and OrderkCupPers, respectively.

All partitions of an integer k can be generated in output-sensitive time using partitions of integer $k-1$. For instance, see [18] for a Python code to do the same. Hence, Step 1.1 of ComputePartitionBarcodes runs in time $O(\mathcal{P}^{\uparrow}(d))$ which is upper bounded by $O(d^{\frac{1}{4}}e^{c\sqrt{d}})$, where $c = \pi\sqrt{2/3}$ (See Section 5). Note that ExtendCupPersKParts (and CupPers2Parts) executes beyond Steps 0 with a parameter λ_k only when it is called for the first time with that parameter. The total number of calls to ExtendCupPersKParts that proceed to Steps 1 is, therefore, bounded by $\mathcal{P}^{\uparrow}(d)$. If there are subsequent recursive calls to ExtendCupPersKParts with λ_k as a parameter it returns at Step 0. Note that ExtendCupPersKParts calls itself recursively only once (in Step 1). So the total number of calls where ExtendCupPersKParts returns at Step 0 is bounded by $\mathcal{P}^{\uparrow}(d)$. If ExtendCupPersKParts returns at Step 0, the cost of execution is $O(1)$, else it is $O(n^4)$. Hence, the total cost of Step 1.2 of ComputePartitionBarcodes is $\mathcal{P}^{\uparrow}(d)O(n^4)$ which is $O(d^{\frac{1}{4}}e^{c\sqrt{d}}n^4)$.

D Stability

We establish stability of partition modules of the cup product for Rips and Čech complexes. In particular, we show that when the Gromov-Haudorff distance (Hausdorff distance) between a point cloud and its perturbation is bounded by a small constant, then the interleaving distance between barcodes of respective Rips (Čech)partition modules is also bounded by a small constant.

D.1 Geometric complexes

▶ **Definition 25** (Rips complexes). *Let X be a finite point set in $\mathbb{R}^d$. The Rips complex of X at scale t consists of all simplices with diameter at most t, where the diameter of a simplex is the maximum distance between any two points in the simplex. In other words,*

$$\mathrm{VR}_t(X) = \{S \subset X \mid \operatorname{diam} S \leq t\}.$$

The Rips filtration of X, denoted by $\mathrm{VR}_\bullet(X)$, is the nested sequence of complexes $\{\mathrm{VR}_t(X)\}_{t\geq 0}$, where $\mathrm{VR}_s(X) \subseteq \mathrm{VR}_t(X)$ for $s \leq t$.

▶ **Definition 26** (Čech complexes). *Let X be a finite point set in $\mathbb{R}^d$. Let $D_{r,x}$ denote a Euclidean ball of radius r centered at x. The Čech complex of X for radius r consists of all simplices satisfying the following condition:*

$$\check{\mathsf{C}}_r(X) = \{S \subset X \mid \bigcap_{x\in S} D_{r,x} \neq \emptyset\}.$$

The Čech filtration of X, denoted by $\check{\mathsf{C}}_\bullet(X)$, is the nested sequence of complexes $\{\check{\mathsf{C}}_r(X)\}_{r\geq 0}$, where $\check{\mathsf{C}}_s(X) \subseteq \check{\mathsf{C}}_t(X)$ for $s \leq t$.

D.2 The Gromov-Hausdorff distance

Let X and Y be compact subspaces of a metric space M with distance d. For a point $p \in X$, $d(p;Y)$ is defined as

$$d(p, Y) = \inf \{d(p, q) \mid q \in Y\}$$

and the distance $d(X, Y)$ between spaces X and Y is defined as

$$d(X, Y) = \sup \{d(p, Y) \mid p \in X\}.$$

The Hausdorff distance d_H between X and Y is defined as

$$d_H(X, Y) = \max \{d(X, Y), d(Y, X)\}.$$

The Gromov-Hausdorff distance d_{GH} between X and Y is defined as

$$d_{GH}(X, Y) = \inf \{d_H(f(X); g(Y)) \mid f : X \hookrightarrow M\ ,\ g : Y \hookrightarrow M\}$$

where the infimum is taken over all isometric embeddings $f : X \hookrightarrow M$, $g : Y \hookrightarrow M$ into some common metric space M.

D.3 Stability of partition modules of the cup product

In this section, as a direct consequence of the functoriality of the cup product, we show that the partition modules are stable for Čech and Rips filtrations.

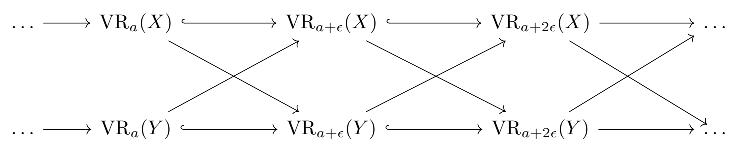
To begin with, let $d_I(M, N)$ denote the interleaving distance between two persistence modules M and N [8]. For finite point sets X and Y in $\mathbb{R}^d$, let $d_H(X, Y)$ denote the Hausdorff distance, and let $d_{GH}(X, Y)$ denote the Gromov-Hausdorff distance between them. Let $\mathrm{VR}_\bullet(X)$ and $\mathrm{VR}_\bullet(Y)$ denote the respective Rips filtrations of X and Y, and let $\check{\mathsf{C}}_\bullet(X)$ and $\check{\mathsf{C}}_\bullet(Y)$ denote the respective Čech filtrations of X and Y.

▶ **Theorem 27.** *Let $\lambda_q = \{s_1, s_2, \ldots, s_\ell\}$ be a partition of an integer q. Then, for finite point sets X and Y in $\mathbb{R}^d$, the following identities hold true:*

$$\frac{1}{2} d_{\mathrm{I}}(\mathrm{im}\, \mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(X)), \mathrm{im}\, \mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(Y))) \leq d_{GH}(X, Y).$$

$$\frac{1}{2} d_{\mathrm{I}}(\mathrm{im}\, \mathsf{H}^{\lambda_q}(\smile \check{\mathsf{C}}_\bullet(X)), \mathrm{im}\, \mathsf{H}^{\lambda_q}(\smile \check{\mathsf{C}}_\bullet(Y))) \leq d_H(X, Y).$$

Proof. Let X and Y be point sets in a common Euclidean space $\mathbb{R}^d$ such that $d_{GH}(X, Y) = \frac{\epsilon}{2}$. Then, in the proof of Lemma 4.3 of [8], Chazal et al. showed that $\mathrm{VR}_\bullet(X)$ and $\mathrm{VR}_\bullet(Y)$ are ϵ-interleaved.



Applying the cohomology functor, we obtain an ϵ-interleaving of the respective cohomology persistence modules. Let $\{\varphi^*_{a',a}\}_{a',a\in\mathbb{R}}$ and $\{\psi^*_{a',a}\}_{a',a\in\mathbb{R}}$ denote the structure maps for the modules $\mathsf{H}^*(\mathrm{VR}_\bullet(X))$ and $\mathsf{H}^*(\mathrm{VR}_\bullet(Y))$, respectively. Also, let $F_{a+\epsilon} : \mathsf{H}^*(\mathrm{VR}_{a+\epsilon}(X)) \to \mathsf{H}^*(\mathrm{VR}_a(Y))$ and $G_{a+\epsilon} : \mathsf{H}^*(\mathrm{VR}_{a+\epsilon}(Y)) \to \mathsf{H}^*(\mathrm{VR}_a(X))$ for all $a \in \mathbb{R}$ be the maps that assemble to give an ϵ-interleaving between $\mathsf{H}^*(\mathrm{VR}_\bullet(X))$ and $\mathsf{H}^*(\mathrm{VR}_\bullet(Y))$.

$$\begin{array}{ccccccc}
\cdots \longleftarrow \mathsf{H}^*(\mathrm{VR}_a(X)) & \xleftarrow{\varphi^*_{a+\epsilon,a}} & \mathsf{H}^*(\mathrm{VR}_{a+\epsilon}(X)) & \xleftarrow{\varphi^*_{a+2\epsilon,a+\epsilon}} & \mathsf{H}^*(\mathrm{VR}_{a+2\epsilon}(X)) & \longleftarrow & \cdots \\
 & F_{a+\epsilon} \times G_{a+\epsilon} & & F_{a+2\epsilon} \times G_{a+2\epsilon} & & F_{a+3\epsilon} \times G_{a+3\epsilon} & \\
\cdots \longleftarrow \mathsf{H}^*(\mathrm{VR}_a(Y)) & \xleftarrow[\psi^*_{a+\epsilon,a}]{} & \mathsf{H}^*(\mathrm{VR}_{a+\epsilon}(Y)) & \xleftarrow[\psi^*_{a+2\epsilon,a+\epsilon}]{} & \mathsf{H}^*(\mathrm{VR}_{a+2\epsilon}(Y)) & \longleftarrow & \cdots
\end{array}$$

For every $j \in [\ell]$, let $[\alpha_j] \in \mathsf{H}^{s_j}(\mathsf{K}_i)$. Then, by the functoriality of the cup product, $\varphi^*_{a+\epsilon,a}([\alpha_1] \smile [\alpha_2] \smile \cdots \smile [\alpha_\ell]) = \varphi^*_{a+\epsilon,a}([\alpha_1]) \smile \varphi^*_{a+\epsilon,a}([\alpha_2]) \smile \cdots \smile \varphi^*_{a+\epsilon,a}([\alpha_\ell])$, and hence for all $a \in \mathbb{R}$, $\varphi^*_{a+\epsilon,a}$ restricts to a map $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(X)) \to \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(X))$.

The functoriality of the cup product also gives us the restrictions $\psi^*_{a+\epsilon,a} : \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(Y)) \to \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(Y))$, $F_{a+\epsilon} : \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(X)) \to \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(Y))$ and $G_{a+\epsilon} : \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(Y)) \to \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(X))$. It is easy to check that the restrictions of the maps $\{F_{a+\epsilon}\}_{a\in\mathbb{R}}$ and $\{G_{a+\epsilon}\}_{a\in\mathbb{R}}$ assemble to give an ϵ-interleaving between the persistence modules $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(X))$ and $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(Y))$ with the restrictions of $\{\varphi^*_{a,a'}\}_{a,a'\in\mathbb{R}}$ and $\{\psi^*_{a,a'}\}_{a,a'\in\mathbb{R}}$ as the structure maps for $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(X))$ and $\operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_\bullet(Y))$, respectively.

$$\begin{array}{ccccccc}
\cdots \longleftarrow \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(X)) & \xleftarrow{\varphi^*_{a+\epsilon,a}} & \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(X)) & \xleftarrow{\varphi^*_{a+2\epsilon,a+\epsilon}} & \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+2\epsilon}(X)) & \longleftarrow & \cdots \\
 & F_{a+\epsilon} \times G_{a+\epsilon} & & F_{a+2\epsilon} \times G_{a+2\epsilon} & & F_{a+3\epsilon} \times G_{a+3\epsilon} & \\
\cdots \longleftarrow \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_a(Y)) & \xleftarrow[\psi^*_{a+\epsilon,a}]{} & \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+\epsilon}(Y)) & \xleftarrow[\psi^*_{a+2\epsilon,a+\epsilon}]{} & \operatorname{im}\mathsf{H}^{\lambda_q}(\smile \mathrm{VR}_{a+2\epsilon}(Y)) & \longleftarrow & \cdots
\end{array}$$

The above diagram, proves the first claim.

Cohen-Steiner et al. [9] showed that if $d_H(X,Y) = \frac{\epsilon}{2}$, then there exists an ϵ-interleaving between $\check{\mathsf{C}}_\bullet(X)$ and $\check{\mathsf{C}}_\bullet(Y)$. Using this fact and repeating the argument above, we obtain the following the second claim. ◀

Thus, if the Gromov-Hausdorff distance between point sets X and Y is small, then the interleaving distance for the respective ordinary persistence modules, cup modules and partition modules of cup product are all small.

E Computing persistent cup-length

This section expands Section 4.2. The *cup length* of a ring is defined as the maximum number of multiplicands that together give a nonzero product in the ring. Let $\mathbf{Int}_*$ denote the set of all closed intervals of $\mathbb{R}$, and let $\mathbf{Int}_\circ$ denote the set of all the open-closed intervals of $\mathbb{R}$ of the form $(a,b]$. Let $\mathcal{F}$ be an $\mathbb{R}$-indexed filtration of simplicial complexes. The *persistent cup-length function* $\mathbf{cuplength}_\bullet : \mathbf{Int}_* \to \mathbb{N}$ (introduced in [12,13]) is defined as the function from the set of closed intervals to the set of non-negative integers.[1] Specifically, it assigns to each interval $[a,b]$, the cup-length of the image ring $\operatorname{im}\big(\mathsf{H}^*(\mathsf{K})[a,b]\big)$, which is the ring $\operatorname{im}\big(\mathsf{H}^*(\mathsf{K}_b) \to \mathsf{H}^*(\mathsf{K}_a)\big)$.

Let the restriction of a cocycle ξ to index k be ξ^k. We say that a cocycle ζ is defined at p if there exists a cocycle ξ in K_q for $q \geq p$ and $\zeta = \xi^p$.

For a persistent cohomology basis Ω, we say that $[d,b)$ is *a supported interval of length k for Ω* if there exists cocycles $\xi_1, \ldots, \xi_k \in \Omega$ such that the product cocycle $\eta^s = \xi_1^s \smile \cdots \smile \xi_k^s$ is nontrivial for every $s \in [d,b)$ and η^s either does not exist or is trivial outside of $[d,b)$.

[1] For simplicity and without loss of generality, we define persistent cup-length only for intervals in $\mathbf{Int}_*$, and persistent cup-length diagram only for intervals in $\mathbf{Int}_\circ$.

In this case, we say that $[d, b)$ *is supported by* $\{\xi_1, \dots, \xi_k\}$. The max-length of a supported interval $[d, b)$, denoted by $\ell_\Omega([d, b))$, is defined as

$$\ell_\Omega([d, b)) = \max\{k \in \mathbb{N} \mid \exists \xi_1, \dots, \xi_k \in \Omega \text{ such that } (d, b] \text{ is supported by } \{\xi_1, \dots, \xi_k\}\}.$$

Let $\mathbf{Int}_\Omega$ be the set of supported intervals of Ω. In order to compute the persistent cup-length function, Contessoto et al. [12] define a notion called the *persistent cup-length diagram*, which is a function $\mathbf{dgm}^{\smile}_\Omega : \mathbf{Int}_\circ \to \mathbb{N}$, that assigns to every interval $[d, b)$ in $\mathbf{Int}_\Omega \subset \mathbf{Int}_\circ$ its max-length $\ell_\Omega([d, b))$, and assigns zero to every interval in $\mathbf{Int}_\circ \setminus \mathbf{Int}_\Omega$.

It is worth noting that unlike the order-k product persistence modules, the persistent cup-length diagram is not a topological invariant as it depends on the choice of representative cocycles. While the persistent cup-length diagram is not useful on its own, in Contessoto et al. [12], it serves as an intermediate in computing the persistent cup-length (a stable topological invariant) due to the following theorem.

▶ **Theorem 28** (Contessoto et al. [12]). *Let $\mathcal{F}$ be a filtered simplicial complex, and let Ω be a persistent cohomology basis for $\mathcal{F}$. The persistent cup-length function* $\mathbf{cuplength}_\bullet$ *can be retrieved from the persistent cup-length diagram* $\mathbf{dgm}^{\smile}_\Omega$ *for any $(a, b] \in \mathbf{Int}_\circ$ as follows.*

$$\mathbf{cuplength}_\bullet([a, b]) = \max_{(c,d] \supset [a,b]} \mathbf{dgm}^{\smile}_\Omega((c, d]). \tag{5}$$

Given a P-indexed filtration $\mathcal{F}$, let $V^k_\bullet$ denote its persistent k-cup module. The following result appears as Proposition 5.9 in [13]. We provide a short proof in our notation for the sake of completeness.

▶ **Proposition 29** (Contessoto et al. [13]). $\mathbf{cuplength}_\bullet([a, b]) = \operatorname{argmax}\{k \mid \mathsf{rk}_{V^k_\bullet}([a, b]) \neq 0\}$.

Proof. $\mathbf{cuplength}_\bullet([a, b]) = k \iff$ **1.** There exists a set of cocycles $\{\xi_1, \dots, \xi_k\}$ that are defined at b and $\xi^s_1 \smile \cdots \smile \xi^s_k$ is nontrivial for all $s \in [a, b]$ **2.** For any set of $k+1$ cocycles $\{\zeta_1, \dots, \zeta_{k+1}\}$ that are defined at b, the product $\zeta^s_1 \smile \cdots \smile \zeta^s_{k+1}$ is zero for some $s \in [a, b]$. $\iff$ $\mathsf{rk}_{V^k_\bullet}([a, b]) \neq 0$ and $\mathsf{rk}_{V^{k+1}_\bullet}([a, b]) = 0$. ◀

Given a filtered complex $\mathsf{K}_\bullet : \mathsf{K}_1 \hookrightarrow \mathsf{K}_2 \hookrightarrow \dots$, Contessoto et al. [12] define its p-truncation as the filtration $\mathsf{K}^p_\bullet : \mathsf{K}^p_1 \hookrightarrow \mathsf{K}^p_2 \hookrightarrow \dots$, where for all i, K^p_i denotes the p-skeleton of K_i. We now compare the complexities of computing the persistent cup-length using the algorithm described in Contessoto et al. [12] against computing it with our approach.

Assume that K is a d-dimensional complex of size n, and let n_p denote the number of simplices in the p-skeleton of K. Let $\mathcal{F}$ be a filtration of K and let $\mathcal{F}_p$ be the p-truncation of $\mathcal{F}$. Then, according to Theorem 20 in Contessoto et al. [12], using the persistent cup-length diagram, **1.** the persistent cup-length of $\mathcal{F}$ can be computed in $O(n^{d+2})$ time, **2.** the persistent cup-length of $\mathcal{F}_p$ can be computed in $O(n_p^{p+2})$ time.

In contrast, as noted in Section 3, the barcodes of all the persistent k-cup modules for $k \in \{2, \dots, p\}$ can be computed in $O(p\, n^4)$ time. Note that $\mathsf{rk}_{V^k_\bullet}([a, b]) \neq 0$ if and only if there exists an interval $(x, y]$ in $B(V^k_\bullet)$ such that $(x, y] \supset [a, b]$. This suggests a simple algorithm to compute $\mathbf{cuplength}_\bullet$ from the barcodes of persistent k-cup modules for $k \in \{2, \dots, n\}$, that is, one finds the largest k for which there exists an interval $(x, y] \in B(V^k_\bullet)$ such that $(x, y] \supset [a, b]$. Since the size of $B(V^k_\bullet)$, for every $k \in [n]$, is $O(n)$, the algorithm for extracting the persistent cup-length from the barcode of persistent k-cup modules for $k \in \{2, \dots, d\}$ runs in $O(n^2)$ time. Thus, using the algorithms described in Section 4, the persistent cup-length of a (p-truncated) filtration can be computed in $O(dn^4)$ ($O(pn^4)$) time, which is strictly better than the coarse bound for the algorithm in [12] for $d \geq 3$.

F Correctness of OrderkCupPers

In this section, we provide a brief sketch of correctness of OrderkCupPers. The statements of lemmas and their proofs are analogous to the case when $k = 2$ treated in the main body of the paper.

▶ **Proposition 30.** *Let $\{\varphi_i^* : \mathsf{H}^*(\mathsf{K}_i^*) \to \mathsf{H}^*(\mathsf{K}_{i-1}^*) \mid i \in [n]\}$ denote the structure map of the module $\mathsf{H}^*(\mathsf{K}_\bullet)$. The structure map for the persistent k-cup module $\operatorname{im} \mathsf{H}^*(\smile_\bullet^k)$ is the restriction of $\varphi_\bullet^*$ to the image of $\smile_\bullet^k$.*

Proof. Recall that φ_i^* denotes the induced map on cohomology $\mathsf{H}^*(\mathsf{K}_i) \to \mathsf{H}^*(\mathsf{K}_{i-1})$. Let $\varphi_i^{k\times\otimes}$ denote the tensor product of the map φ_i^* with itself taken k times.

Applying the cohomology functor to the map

$$\smile_\bullet^k : \mathsf{C}^*(\mathsf{K}_\bullet) \otimes \mathsf{C}^*(\mathsf{K}_\bullet) \otimes \cdots \otimes \mathsf{C}^* \to \mathsf{C}^*(\mathsf{K}_\bullet) \tag{6}$$

and using the Künneth theorem for cohomology over fields, we obtain the following diagram:

$$\begin{array}{ccc}
\mathsf{H}^*(\mathsf{K}_i) \otimes \mathsf{H}^*(\mathsf{K}_i) \otimes \cdots \otimes \mathsf{H}^*(\mathsf{K}_i) & \xrightarrow{\quad\smile\quad} & \mathsf{H}^*(\mathsf{K}_i) \\
\Big\downarrow \varphi_i^{k\times\otimes} & & \Big\downarrow \varphi_i^* \\
\mathsf{H}^*(\mathsf{K}_{i-1}) \otimes \mathsf{H}^*(\mathsf{K}_{i-1}) \otimes \cdots \otimes \mathsf{H}^*(\mathsf{K}_{i-1}) & \xrightarrow{\quad\smile\quad} & \mathsf{H}^*(\mathsf{K}_{i-1})
\end{array}$$

For cocycle classes $[\alpha_1], \ldots, [\alpha_k] \in \mathsf{H}^*(\mathsf{K})$, by the functoriality of the cup product, $\varphi_i^*([\alpha_1]) \smile \cdots \smile \varphi_i^*([\alpha_k]) = \varphi_i^*([\alpha_1] \smile \ldots [\alpha_k])$. Since, $[\alpha_1] \smile \cdots \smile [\alpha_k] \in \operatorname{im} \mathsf{H}^*(\smile_i^k)$ is mapped to an element in $\operatorname{im} \mathsf{H}^*(\smile_{i-1}^k)$, the structure map for the persistent k-cup module $\operatorname{im} \mathsf{H}^*(\smile_\bullet^k)$ is the restriction of $\varphi_\bullet^*$ to the image of $\smile_\bullet^k$. ◀

▶ **Definition 31.** *For any $i \in \{0, \ldots, n\}$, a nontrivial cocycle $\zeta \in \mathsf{Z}^*(\mathsf{K}_i)$ is said to be an* order-k product cocycle of K_i *if $[\zeta] \in \operatorname{im} \mathsf{H}^*(\smile_i^k)$.*

▶ **Proposition 32.** *For a filtration $\mathcal{F}$ of simplicial complex K, the birth points of $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet^k))$ are a subset of the birth points of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, and the death points of $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet^k))$ are a subset of the death points of $B(\mathsf{H}^*(\mathsf{K}_\bullet))$.*

Proof. Let $\{(d_{i_j}, b_{i_j}] \mid j \in [k]\}$ be (not necessarily distinct) intervals in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, where $b_{i_{j+1}} \geq b_{i_j}$ for $j \in [k-1]$. Let ξ_{i_j} be a representative for $(d_{i_j}, b_{i_j}]$ for $j \in [k]$.

If $\xi_{i_1} \smile \xi_{i_2}^{b_{i_1}} \smile \cdots \smile \xi_{i_k}^{b_{i_1}}$ is trivial, then by the functoriality of cup product,

$$\begin{aligned}
\varphi_{b_{i_1},r}(\xi_{i_1} \smile \xi_{i_2}^{b_{i_1}} \smile \cdots \smile \xi_{i_k}^{b_{i_1}}) &= \varphi_{b_{i_1},r}(\xi_{i_1}) \smile \varphi_{b_{i_1},r}(\xi_{i_2}^{b_{i_1}}) \smile \cdots \smile \varphi_{b_{i_1},r}(\xi_{i_k}^{b_{i_1}}) \\
&= \xi_{i_1}^r \smile \xi_{i_2}^r \smile \cdots \smile \xi_{i_k}^r
\end{aligned}$$

is trivial $\forall r < b_{i_1}$. Writing contrapositively, if $\exists r < b_{i_1}$ for which $\xi_{i_1}^r \smile \xi_{i_2}^r \smile \cdots \smile \xi_{i_k}^r$ is nontrivial, then $\xi_{i_1} \smile \xi_{i_2}^{b_{i_1}} \smile \cdots \smile \xi_j^{b_{i_1}}$ is nontrivial. Noting that $\operatorname{im} \mathsf{H}^*(\smile_\ell^k)$ for any $\ell \in \{0, \ldots, n\}$ is generated by $\{[\xi_{i_1}^\ell] \smile \{[\xi_{i_2}^\ell] \smile \cdots \smile [\xi_{i_k}^\ell] \mid \xi_{i_j} \in \Omega_{\mathsf{K}}$ for $j \in [k]\}$, it follows that b is the birth point of an interval in $B(\operatorname{im} \mathsf{H}^*(\smile_\bullet^k))$ only if it is the birth point of an interval in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, proving the first claim.

Let $\Omega'_{j+1} = \{[\tau_1], \ldots, [\tau_\ell]\}$ be a basis for $\operatorname{im} \mathsf{H}^*(\smile^k_{j+1})$. Then, Ω'_{j+1} extends to a basis Ω_{j+1} of $\mathsf{H}^*(\mathsf{K}_{j+1})$. If j is not a death index in $B(\mathsf{H}^*(\mathsf{K}_\bullet))$, then $\varphi_{j+1}(\tau_1), \ldots, \varphi_{j+1}(\tau_\ell)$ are all nontrivial and linearly independent. Using Remark 7, it follows that j is not a death index in $B(\operatorname{im} \mathsf{H}^*(\smile^k_\bullet))$, proving the second claim. ◀

▶ **Corollary 33.** *If d is a death index in $B(\operatorname{im} \mathsf{H}^*(\smile^k_\bullet))$, then at most one bar of $B(\operatorname{im} \mathsf{H}^*(\smile^k_\bullet))$ has death index d.*

Proof. The proof is identical to Corollary 10. ◀

Let C_b be the vector space of order-k product cocycle classes created at index b. We note that for a birth index $b \in \{0, \ldots, n\}$, C_b is a subspace of $\mathsf{H}^*(\mathsf{K}_b)$ which can be written as

$$C_b = \begin{cases} \langle [\xi_{i_1}] \smile \cdots \smile [\xi_{i_k}] \mid \xi_{i_j} \text{ for } j \in [k] \text{ are essential cocycles of } \mathsf{H}^*(\mathsf{K}_\bullet) \rangle & \text{if } b = n \\ \langle [\xi_{i_1}] \smile \cdots \smile [\xi^b_{i_k}] \mid \xi_{i_1} \text{ is born at } b \;\&\; \xi_{i_j} \text{ for } j \neq 1 \text{ is born at index } \geq b \rangle & \text{if } b < n \end{cases} \tag{7}$$

For a birth index b, let $\mathbf{C}_b$ be the submatrix of $\mathbf{S}$ formed by representatives whose classes generate C_b, augmented to $\mathbf{S}$ in Steps 2.1.2 (i) and 2.2.2 (i) when $k = b$ in the outer **for** loop of Step 2.

▶ **Theorem 34.** *Algorithm* OrderkCupPers *correctly computes the barcode of persistent k-cup modules.*

Proof. The proof is nearly identical to Theorem 13. The key difference (from Theorem 13) is in how the submatrix $\mathbf{C}_b$ of $\mathbf{S}$ that stores the linearly independent order-k product cocycles born at $\ell = b$ in Steps 2.1 and 2.2 is built. It is easy to check that the classes of the cocycle vectors in $\mathbf{C}_b$ augmented to $\mathbf{S}$ in Steps 2.1 and 2.2 generate the space C_b described in Equation (7). ◀

G Relative cup modules

Let (K, L) be a simplical pair. As in the case of absolute cohomology, for the relative cup product, we have bilinear maps

$$\smile : \mathsf{C}^p(\mathsf{K}, \mathsf{L}) \times \mathsf{C}^q(\mathsf{K}, \mathsf{L}) \to \mathsf{C}^{p+q}(\mathsf{K}, \mathsf{L}) \quad \text{that assemble to give a linear map}$$

$$\smile : \mathsf{C}^*(\mathsf{K}, \mathsf{L}) \otimes \mathsf{C}^*(\mathsf{K}, \mathsf{L}) \to \mathsf{C}^*(\mathsf{K}, \mathsf{L}).$$

Also, we have bilinear maps

$$\smile : \mathsf{H}^p(\mathsf{K}, \mathsf{L}) \times \mathsf{H}^q(\mathsf{K}, \mathsf{L}) \to \mathsf{H}^{p+q}(\mathsf{K}, \mathsf{L}) \quad \text{that assemble to give a linear map}$$

$$\smile : \mathsf{H}^*(\mathsf{K}, \mathsf{L}) \otimes \mathsf{H}^*(\mathsf{K}, \mathsf{L}) \to \mathsf{H}^*(\mathsf{K}, \mathsf{L}).$$

For a filtered complex K, its persistent relative cohomology is given by $\mathsf{H}^*(\mathsf{K}, \mathsf{K}_\bullet)$ with linear maps given by inclusions [15]. Written in our convention for intervals, every finite bar $(d, b]$ in $B(\mathsf{H}^i(\mathsf{K}_\bullet))$, we have a corresponding finite bar $(d, b]$ in $B(\mathsf{H}^{i+1}(\mathsf{K}, \mathsf{K}_\bullet))$, and for every infinite bar $(d, n]$ in $B(\mathsf{H}^i(\mathsf{K}_\bullet))$, we have an infinite bar $(-1, d]$ in $B(\mathsf{H}^i(\mathsf{K}, \mathsf{K}_\bullet))$.

Defining relative cup modules. Consider the following homomorphism given by cup products:

$$\smile_\bullet\colon \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet)\otimes \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet)\to \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet). \tag{8}$$

Taking $G_\bullet = \smile_\bullet$ in the definition of image persistence, we get a persistence module, denoted by $\operatorname{im}\operatorname{rel}\mathsf{H}^*(\smile \mathsf{K}_\bullet)$, which is called the *persistent relative cup module*. Whenever the underlying filtered complex is clear from the context, we use the shorthand notation $\operatorname{im}\operatorname{rel}\mathsf{H}^*(\smile_\bullet)$ instead of $\operatorname{im}\mathsf{H}^*(\smile \mathsf{K}_\bullet)$.

Defining relative k-cup modules. Consider image persistence of the map

$$\smile_\bullet^k\colon \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet)\otimes \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet)\otimes\cdots\otimes \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet)\to \mathsf{C}^*(\mathsf{K},\mathsf{K}_\bullet) \tag{9}$$

where the tensor product is taken k times. Taking $G_\bullet = \smile_\bullet^k$ in the definition of image persistence, we get the *persistent relative k-cup module module* $\operatorname{im}\operatorname{rel}\mathsf{H}^*(\smile_\bullet^k)$.

Next, we will describe how to compute the barcode of $\operatorname{im}\operatorname{rel}\mathsf{H}^*(\smile_\bullet)$, which being an image module is a submodule of $\mathsf{H}^*(\mathsf{K},\mathsf{K}_\bullet)$. The vector space $\operatorname{im}\operatorname{rel}\mathsf{H}^*(\smile_i)$ is a subspace of the vector space $\mathsf{H}^*(\mathsf{K},\mathsf{K}_i)$. Let us call this subspace the *relative cup space* of $\mathsf{H}^*(\mathsf{K},\mathsf{K}_i)$. RelCupPers describes this algorithm to compute relative cup modules. First, in Step 0, we compute the barcode of the cohomology persistence module $\mathsf{H}^*(\mathsf{K},\mathsf{K}_\bullet)$ along with a relative persistent cohomology basis. This can be achieved in $O(n^3)$ time by applying the standard algorithm on the anti-transpose of the boundary matrix [15, Section 3.4]. The basis H is maintained with the matrix $\mathbf{H}$ whose columns are representative cocycles. The matrix $\mathbf{H}$ is initialized with the empty matrix. $\partial^\perp$ maintains the relative coboundaries as one processes the matrix in the reverse filtration order. At index n, $\partial^\perp$ is empty. Throughout, $\partial^\perp$ is stored in the leftmost n columns of $\mathbf{S}$, and there are no other columns in $\mathbf{S}$ at index n. Subsequently, nontrivial relative cocycle vectors are added to $\mathbf{S}$. The classes of the nontrivial cocycles in matrix $\mathbf{S}$ form a basis S for the relative cup space at any point in the course of the algorithm. In Step 2, at each index k, the k-th column of $\partial^\perp$ is populated with the coboundary of k. The remainder of the birth case and the whole of the death case is handled exactly like RelCupPers. The correctness and complexity proofs for RelCupPers are identical to CupPers.

Algorithm RelCupPers $(\mathsf{K}_\bullet)$

- Step 0. Compute barcode $B(\mathcal{F}) = \{(d_i, b_i]\}$ of $\mathsf{H}^*(\mathsf{K},\mathsf{K}_\bullet)$ with representative cocycles ξ_i
- Step 1. Initialize an $n\times n$ coboundary matrix $\partial^\perp$ as the zero matrix; $\partial^\perp$ is maintained as a submatrix of $\mathbf{S}$; Initially all columns in $\mathbf{S}$ come from columns in $\partial^\perp$. Subsequently, in the course of the algorithm, new columns are added to (and removed from) the right of $\partial^\perp$ in $\mathbf{S}$ and the entries of $\partial^\perp$ are also modified; Initialize $\mathbf{H}$ with the empty matrix
- Step 2. For $k := n$ to 1 do
  - For every simplex σ_j that has σ_k as a face, set $\partial^\perp_{j,k} = 1$
  - Step 2.1 For every i with $k = b_i$ (k is a birth-index) and $\deg(\xi_i) > 0$
    - Step 2.1.1 Update $\mathbf{H} := [\mathbf{H} \mid \xi_i]$
    - Step 2.1.2 For every $\xi_j \in \mathbf{H}$
      i. If $(\zeta \leftarrow \xi_i \smile \xi_j) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta\cdot\text{birth} := k$ and $\zeta\cdot\text{rep@birth} := \zeta$
  - Step 2.2 If $k = d_i$ (k is a death-index) for some i and $\deg(\xi_i) > 0$ then
    - Step 2.2.1 Reduce $\mathbf{S}$ with left-to-right column additions

  * Step 2.2.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(k, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
  * Step 2.2.3 Update $\mathbf{H}$ by removing the column ξ_i

In Algorithm RelOrderkCupPers, The initialization and maintenance of the matrix $\mathbf{S}$ and $\partial^\perp$ is the same as for RelCupPers. The matrices $\mathbf{H}$ and $\mathbf{R}$ are intialized with empty matrices. The remainder of the birth case and the whole of the death case are identical to OrderkCupPers. The correctness and complexity proofs for RelOrderkCupPers are identical to OrderkCupPers.

Algorithm RelOrderkCupPers ($\mathsf{K}_\bullet$,k)

- Step 0. If $k = 2$, return the barcode with representatives $\{(d_{i,2}, b_{i,2}], \xi_{i,2}\}$ computed by CupPers on $\mathsf{K}_\bullet$
  else $\{(d_{i,k-1}, b_{i,k-1}], \xi_{i,k-1}\} \leftarrow$ RelOrderkCupPers($\mathsf{K}_\bullet$, $k-1$)
- Step 1. Initialize an $n \times n$ coboundary matrix $\partial^\perp$ as the zero matrix; $\partial^\perp$ is maintained as a submatrix of $\mathbf{S}$; Initially all columns in $\mathbf{S}$ come from columns in $\partial^\perp$. Subsequently, in the course of the algorithm, new columns are added to (and removed from) the right of $\partial^\perp$ in $\mathbf{S}$ and the entries of $\partial^\perp$ are also modified; Initialize $\mathbf{H}$ and $\mathbf{R}$ with empty matrices
- Step 2. For $\ell := n$ to 1 do
  - For every simplex σ_j that has σ_k as a face, set $\partial^\perp_{j,k} = 1$
  - Step 2.1 For every r s.t. $b_{r,1} = \ell \neq n$ (i.e., ℓ is a birth-index) and $\deg(\xi_{r,1}) > 0$
    * Step 2.1.1 Update $\mathbf{H} := [\mathbf{H} \mid \xi_{r,1}]$
    * Step 2.1.2 For every $\xi_{j,k-1} \in \mathbf{R}$
      i. If $(\zeta \leftarrow \xi_{r,1} \smile \xi_{j,k-1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.2 For all s such that $\ell = b_{s,k-1}$
    * Step 2.2.1 If $\ell \neq n$, update $\mathbf{R} := [\mathbf{R} \mid \xi_{s,k-1}]$
    * Step 2.2.2 For every $\xi_{i,1} \in \mathbf{H}$
      i. If $(\zeta \leftarrow \xi_{s,k-1} \smile \xi_{i,1}) \neq 0$ and ζ is independent in $\mathbf{S}$, then $\mathbf{S} := [\mathbf{S} \mid \zeta]$ with column ζ annotated as $\zeta \cdot \text{birth} := \ell$ and $\zeta \cdot \text{rep@birth} := \zeta$
  - Step 2.3 If $\ell = d_{i,1}$ (i.e. ℓ is a death-index) and $\deg(\xi_{i,1}) > 0$ for some i then
    * Step 2.3.1 Reduce $\mathbf{S}$ with left-to-right column additions
    * Step 2.3.2 If a nontrivial cocycle ζ is zeroed out, remove ζ from $\mathbf{S}$, generate the bar-representative pair $\{(\ell, \zeta \cdot \text{birth}], \zeta \cdot \text{rep@birth}\}$
    * Step 2.3.3 Remove the column $\xi_{i,1}$ from $\mathbf{H}$
    * Step 2.3.4 Remove the column $\xi_{j,k-1}$ from $\mathbf{R}$ if $d_{j,k-1} = \ell$ for some j

Lack of duality. In contrast to ordinary persistence, the following examples highlight the fact that the barcodes of persistent (absolute) cup modules differ from persistent relative cup modules. In fact, in general, there doesn't seem to be any bijection between corresponding intervals.

▶ **Example 35.** Let K be a torus with a disk removed. A torus can be obtained by identifying the opposite sides of a $[-1, 1]^2$ square. The space K can be obtained by removing a circle of radius 1 around the origin. We now give the following CW structure to K: Let x_0 and x_1 be the 0-cells, p, q, r and s be the 1-cells and α be the 2-cell. p and q are loops around x_0, r joins x_0 and x_1, and s is a loop around x_1. The attachment of the 2-cell α is given by the word $pqp^{-1}q^{-1}rsr^{-1}$. See Figure 2 for an illustration.

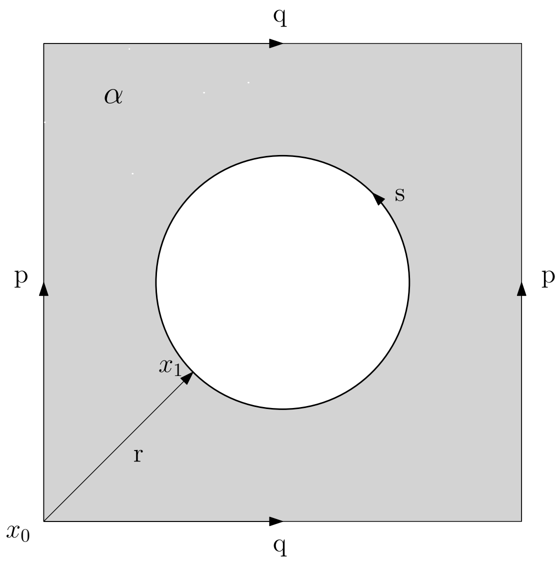


Figure 2 Complex K is a torus with a disk removed.

Consider the cellular filtration $\mathsf{K}_\bullet$ on K:

$$\begin{aligned}
\mathsf{K}_0 &= \{x_0, x_1\}\,, \\
\mathsf{K}_1 &= \mathsf{K}_0 \cup \{s, r\}\,, \\
\mathsf{K}_2 &= \mathsf{K}_1 \cup \{p, q\}\,, \\
\mathsf{K}_3 &= \mathsf{K}_2 \cup \{\alpha\}\,.
\end{aligned}$$

It is easy to check that the persistent (absolute) cup module for $\mathsf{K}_\bullet$ is trivial. However, since $\mathsf{K}_3/\mathsf{K}_1$ is a torus, the persistent relative cup module is nontrivial.

▶ **Example 36.** Let L' be a torus realized as a CW complex with a 0-cell x, two 1-cells a and b and a 2-cell β. We now add a 2-cell α to L' to obtain a CW complex $\mathsf{L} = \mathsf{L}' \cup \{\alpha\}$. See Figure 3 for an illustration.

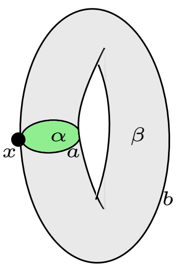


Figure 3 Complex L is a torus with a disk added.

Now consider the following cellular filtration $\mathsf{L}_\bullet$ on L:

$\mathsf{L}_0 = \{x\}$

$\mathsf{L}_1 = \mathsf{L}_0 \bigcup \{a, b\}$

$\mathsf{L}_2 = \mathsf{L}_1 \bigcup \{\beta\}$

$\mathsf{L}_3 = \mathsf{L}_2 \bigcup \{\alpha\}$

For the filtration $\mathsf{L}_\bullet$, the persistent (absolute) cup module is nontrivial since L_2 is a torus. On the other hand, it is easy to check that the persistent relative cup module is trivial.