Zigzag Persistence via Reflections and Transpositions

Clément Maria * Steve Y. Oudot †

Abstract

We introduce a new algorithm for computing zigzag persistence, designed in the same spirit as the standard persistence algorithm. Our algorithm reduces a single matrix, maintains an explicit set of chains encoding the persistent homology of the current zigzag, and updates it under simplex insertions and removals. The total worst-case running time matches the usual cubic bound.

A noticeable difference with the standard persistence algorithm is that we do not insert or remove new simplices "at the end" of the zigzag, but rather "in the middle". To do so, we use arrow reflections and transpositions, in the same spirit as reflection functors in quiver theory. Our analysis introduces new kinds of reflections in quiver representation theory: the "injective and surjective diamonds". It also introduces the "transposition diamond" which models arrow transpositions. For each type of diamond we are able to predict the changes in the interval decomposition and associated compatible bases. Arrow transpositions have been studied previously in the context of standard persistent homology, and we extend the study to the context of zigzag persistence. For both types of transformations, we provide simple procedures to update the interval decomposition and associated compatible homology basis.

1 Introduction.

Zigzag persistence, as introduced by Carlsson and de Silva [7], deals with finite sequences of finite-dimensional vector spaces connected by linear maps, where the maps can be oriented forward or backward, as pictured by the bidirectional arrows in the following diagram:

$$(1.1) \qquad V_1 \longleftrightarrow V_2 \longleftrightarrow \cdots \longleftrightarrow V_{n-1} \longleftrightarrow V_n$$

These are special types of *quiver representations* called *zigzag modules*, for which both the Krull-Schmidt principle and Gabriel's theorem hold and take the following form:

Theorem 1.1. (Krull-Schmidt, Gabriel) *Every zigzag module $\mathbb{V}$ over a given field $\mathbb{F}$ is decomposable as a direct sum of indecomposable modules*

$$(1.2) \qquad \mathbb{V} = \mathbb{V}^1 \oplus \mathbb{V}^2 \oplus \cdots \oplus \mathbb{V}^N,$$

where each indecomposable $\mathbb{V}^j$ is isomorphic to some interval module $\mathbb{I}[b_j; d_j]$*, defined as:*

(1.3)

$$\underbrace{0 \overset{0}{\longleftrightarrow} \cdots \overset{0}{\longleftrightarrow} 0}_{[1;b_j-1]} \overset{0}{\longleftrightarrow} \underbrace{\mathbb{F} \overset{\mathbb{1}}{\longleftrightarrow} \cdots \overset{\mathbb{1}}{\longleftrightarrow} \mathbb{F}}_{[b_j;d_j]} \overset{0}{\longleftrightarrow} \underbrace{0 \overset{0}{\longleftrightarrow} \cdots \overset{0}{\longleftrightarrow} 0}_{[d_j+1;m]}$$

Moreover, the decomposition is unique up to isomorphism and reordering of the terms $\mathbb{V}^j$. We call b_j the birth *and d_j the* death *of each interval module $\mathbb{I}[b_j; d_j]$.*

What this result says is that the algebraic structure of a zigzag module $\mathbb{V}$ is completely determined by the finite collection of intervals $[b_j; d_j]$ involved in its decomposition. This collection is called the *persistence barcode* of $\mathbb{V}$, and it is our target object.

In practice, zigzag modules are obtained by computing the homology of finite sequences $\mathbb{K}$ of simplicial complexes connected by inclusion maps, called *zigzag filtrations*:

$$(1.4) \qquad \mathbf{K}_1 \longleftrightarrow \mathbf{K}_2 \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}_{n-1} \longleftrightarrow \mathbf{K}_n$$

Such filtrations appear in a variety of contexts, e.g. topological inference [11, 14], shape classification [5, 10], and clustering [9, 12], to name a few. Most of the time the arrows in the filtrations share the same orientation, however in some notable cases they don't. The families of Vietoris-Rips complexes designed for topological inference by Morozov [19] and analyzed by Oudot and Sheehy [22] are a good illustration. Their efficiency in terms of memory usage depends critically on the fact that some of the inclusions are oriented backwards, so the overall size of the family remains manageable. With such lightweight constructions at hand, the main practical bottleneck in the topological inference pipeline has shifted from building the filtrations to computing their interval decompositions as in theorem 1.1, and there is now a real need for efficient algorithms to perform this task.

Existing Algorithms. The special case where all the arrows in the filtration have the same orientation – commonly known as *standard persistence* – has been extensively studied. In this case, zigzag modules are just modules over the ring of polynomials $\mathbb{F}[t]$, so computing their interval decomposition as in theorem 1.1 comes

*`clement.maria@inria.fr`, INRIA Sophia Antipolis.
†`steve.oudot@inria.fr`, INRIA Saclay-Ile de France.

down to reducing a matrix to column-echelon or row-echelon form over $\mathbb{F}[t]$, with the additional twist that the ordering of the simplices by time of insertion in the filtration must be preserved [15, 23]. Most methods use Gaussian elimination for this reduction and therefore incur a cubic worst-case time complexity in the number n of simplex insertions [20]. Nevertheless, in practice they are observed to behave near linearly in n on typical data[1]. The most optimized ones among them [1, 3] are able to process millions of simplex insertions per second on a recent machine, which is considered fast enough for many practical purposes.

By contrast, the general zigzag case has received much less attention despite its growing interest in applications. Perhaps the main reason is that, unlike standard persistence modules, it is unknown whether general zigzag modules can be viewed as modules over a ring of polynomials. Hence, computing their interval decompositions as in theorem 1.1 requires more elaborate machinery than mere matrix reduction. The so-called *right filtration functor* of Carlsson and de Silva [7] is an example of such machinery. Introduced originally as a tool to prove Gabriel's theorem, it was eventually turned into the first – and so far only – practical algorithm to decompose general zigzag modules [8]. This algorithm works with modules derived at the homology level from zigzag filtrations $\mathbb{K}$ in which each arrow corresponds to a single simplex insertion or deletion. It scans the zigzag filtration $\mathbb{K}$ from left to right, adding or removing one simplex at a time, and maintaining a compatible basis – in fact three, one for the cycles, one for the boundaries, and one for the killing chains – for the *right filtration* of the zigzag prefix $\mathbb{K}[1; i]$ at iteration i (i.e. the zigzag filtration restricted to the i first complexes). Its implementation is available as part of the `C++` library `Dionysus` [19] and performs reasonably well in practice, however nowhere as efficiently as the aforementioned optimized algorithms for standard persistence. Its pseudo-code is also significantly longer and more intricate due to the required extra machinery. As a result, while reproducing it step by step is easy, grasping the higher-level picture of how and why it works is a comparatively challenging task. To what extent the approach (and its theoretical analysis) can be simplified and optimized is becoming an essential question.

Our Contributions. We introduce a new method for computing zigzag persistence in the same context as [8]. Our approach is inspired from another (and more ancient) proof of Gabriel's theorem [2], which performs arrow reflections in sequence in a zigzag module and tracks the corresponding changes in its interval decomposition. Our algorithm scans the input zigzag filtration $\mathbb{K}$ from left to right as before, however it appends to the current prefix $\mathbb{K}[1; i] := \mathbf{K}_1 \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}_i$ a *descending chain* of subcomplexes in which every arrow is backward and corresponds to a single simplex deletion:

$$\mathbf{K}_1 \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}_i \xleftarrow{\tau_m} \mathbf{K}_i \setminus \{\tau_m\} \xleftarrow[\tau_{m-1}]{} \mathbf{K}_i \setminus \{\tau_m, \tau_{m-1}\} \xleftarrow[\tau_{m-2}]{} \cdots \xleftarrow[\tau_1]{} \emptyset$$

Adding such a descending chain is a way for us to anticipate the simplex deletions that will occur in the rest of the input zigzag, even though the actual order in which they will occur may not be the same as ours. Every new simplex insertion or deletion happens at the junction between the zigzag prefix and the descending chain. The corresponding changes in the zigzag filtration can be described as a combination of *arrow reflections* and *arrow transpositions* taking place in the descending chain. An *arrow reflection* is described by the diagram:

$$\begin{tikzcd}[row sep=small] & & \mathbf{K} \cup \{\sigma\} & & \\ \cdots \arrow[r, leftrightarrow] & \mathbf{K} \arrow[ur, "\sigma"] & & \mathbf{K} \arrow[ul, "\sigma"'] \arrow[r, leftrightarrow] & \cdots \\ & & \mathbf{K} \arrow[ul, "\mathbb{1}"] \arrow[ur, "\mathbb{1}"'] & & \end{tikzcd}$$

and an *arrow transposition* is described by the diagram:

$$\begin{tikzcd}[row sep=small] & & \mathbf{K} \cup \{\tau\} \arrow[dl, "\sigma"'] & & \\ \cdots \arrow[r, leftrightarrow] & \mathbf{K} \cup \{\sigma, \tau\} & & \mathbf{K} \arrow[ul, "\tau"'] \arrow[dl, "\sigma"] \arrow[r, leftrightarrow] & \cdots \\ & & \mathbf{K} \cup \{\sigma\} \arrow[ul, "\tau"] & & \end{tikzcd}$$

The changes in the zigzag filtration consist in passing from the bottom representation to the top representation of either one of these diagrams.

These transformations of the zigzag filtration induce *diamonds* at the homology level. Specifically, an arrow reflection induces an *injective or surjective diamond*:

$$\begin{tikzcd}[row sep=small] & & W & & \\ \cdots \arrow[r, leftrightarrow] & V \arrow[ur, "f"] & & V \arrow[ul, "f"'] \arrow[r, leftrightarrow] & \cdots \\ & & V \arrow[ul, "\mathbb{1}"] \arrow[ur, "\mathbb{1}"'] & & \end{tikzcd}$$

depending whether f is injective or surjective. An arrow transposition induces a *transposition diamond*:

$$\begin{tikzcd}[row sep=small] & & W_i \arrow[dl, "d"'] & & \\ \cdots \arrow[r, leftrightarrow] & V_{i+1} & & V_{i-1} \arrow[ul, "b"'] \arrow[dl, "a"] \arrow[r, leftrightarrow] & \cdots \\ & & V_i \arrow[ul, "c"] & & \end{tikzcd}$$

[1]Which explains why they are used in practice instead of asymptotically subcubic methods like the hierarchical matrix reduction algorithm of Milosavljevic et al. [18], which computes persistence in matrix multiplication time.

where a, b, c, d satisfy an exactness hypothesis [7]. We introduce and prove the *Injective Diamond Principle*, the *Surjective Diamond Principle* and the *Transposition Diamond Principle* in order to express the evolution in the interval decomposition of a zigzag module when passing one of the diamonds presented above. These diamond principles add up to the *Exact Diamond Principle* originally introduced by Carlsson and de Silva [7]. The diamonds corresponding to arrow transpositions were studied by Cohen-Steiner et al. [13] in the context of standard persistence. The transposition diamond principle generalizes the study to zigzag persistence.

Our algorithm to compute zigzag persistence is just one big sequence of diamond traversals. We handle each arrow reflection in time $O(m^2)$ and each arrow transposition in time $O(m)$, where m is the size of the largest simplicial complex in the zigzag filtration. Hence our algorithm can decompose zigzag modules into intervals in time $O(nm^2)$ as in [8]. Our preliminary experiments show a good behavior of our algorithm compared to the one in [8] in practice. Moreover, the similarity of our method to the standard persistence algorithm opens the door to all kinds of optimizations. These questions, and others, are discussed at the end of the paper.

Introduction to Quiver Theory. Let $(\mathbb{F}, +, \cdot)$ be an arbitrary field. An *A_n-type quiver* $\mathcal{Q}$ is a directed graph:

$$\bullet_1 \longleftrightarrow \bullet_2 \longleftrightarrow \cdots \longleftrightarrow \bullet_{n-1} \longleftrightarrow \bullet_n$$

where bidirectional arrows are either forward or backward. We define two total order relations on the indices $\{1, \ldots, n\}$ of the vertices of the quiver, depending on the orientation of the arrows. Let $\leq_{\mathbf{b}}$ be the order satisfying for every indices $i, j \in \{1, \ldots, n\}$, $i \leq_{\mathbf{b}} j$ iff:

$$\begin{cases} \text{either} & i = j, \\ \text{or} & i < j \quad \text{and} \quad \bullet_{j-1} \to \bullet_j \quad \text{is forward,} \\ \text{or} & i > j \quad \text{and} \quad \bullet_{i-1} \leftarrow \bullet_i \quad \text{is backward.} \end{cases}$$

Symmetrically, we define the order $\leq_{\mathbf{d}}$ satisfying, for every indices $i, j \in \{1, \ldots, n\}$, $i \leq_{\mathbf{d}} j$ iff:

$$\begin{cases} \text{either} & i = j, \\ \text{or} & i > j \quad \text{and} \quad \bullet_j \leftarrow \bullet_{j+1} \quad \text{is backward,} \\ \text{or} & i < j \quad \text{and} \quad \bullet_i \to \bullet_{i+1} \quad \text{is forward.} \end{cases}$$

We also define $\max_{\leq_{\mathbf{b}}}$ and $\max_{\leq_{\mathbf{d}}}$ which are the maximum function w.r.t. to the orders $\leq_{\mathbf{b}}$ and $\leq_{\mathbf{d}}$ respectively. For example, in the quiver:

$$\bullet_1 \longrightarrow \bullet_2 \longrightarrow \bullet_3 \longleftarrow \bullet_4 \longrightarrow \bullet_5 \longleftarrow \bullet_6$$

the indices satisfy $6 \leq_{\mathbf{b}} 4 \leq_{\mathbf{b}} 1 \leq_{\mathbf{b}} 2 \leq_{\mathbf{b}} 3 \leq_{\mathbf{b}} 5$ and $1 \leq_{\mathbf{d}} 2 \leq_{\mathbf{d}} 4 \leq_{\mathbf{d}} 6 \leq_{\mathbf{d}} 5 \leq_{\mathbf{d}} 3$. Note that the list of indices sorted according to increasing $\leq_{\mathbf{b}}$ is made of, first, the list of indices, in decreasing index order, that are the tail of a backward arrow, then the index 1, then the list of indices, in increasing index order, that are the head of a forward arrow. The list of indices sorted according to increasing $\leq_{\mathbf{d}}$ is made of, first, the list of indices, in increasing index order, that are the tail of a forward arrow, then the index n, then the list of indices, in decreasing index order, that are the head of a backward arrow. Note that these orders are a reformulation of the *birth-time* and *death-time indices* of [7].

An *$\mathbb{F}$-representation* of $\mathcal{Q}$ is an assignement of a finite dimensional $\mathbb{F}$-vector space V_i for every node $\bullet_i$ and an assignement of a linear map $f_i \colon V_i \to V_{i+1}$ for every forward arrow $\bullet_i \to \bullet_{i+1}$ and of a linear map $g_i \colon V_{i+1} \to V_i$ for every backward arrow $\bullet_i \leftarrow \bullet_{i+1}$. We denote such a representation by $\mathbb{V} = (V_i, f_i/g_i)$. In computational topology, an $\mathbb{F}$-representation is called a *zigzag module*.

For an $\mathbb{F}$-representation $\mathbb{V} = (V_i, f_i/g_i)_{i=1\ldots n}$ of an A_n-type quiver $\mathcal{Q}$, we define $\mathbb{V}[b; d]$ to be the representation $\mathbb{V} = (V_i, f_i/g_i)_{i=b\ldots d}$ of the quiver $\mathcal{Q}[b; d]$ restricted to the vertices (and arrows between them) of indices $b \leq i \leq d$. We call $\mathbb{V}[b; d]$ a *restriction* of the module $\mathbb{V}$ to the range $[b; d]$. If $b = 1$, we call $\mathbb{V}[1; d]$ a *prefix* of $\mathbb{V}$ and if $d = n$ we call $\mathbb{V}[b; n]$ a *suffix*. The following theorem states that the interval decomposition of the restriction $\mathbb{V}[b; d]$ is the direct sum of the intervals of the decomposition of $\mathbb{V}$ restricted to $[b; d]$.

Theorem 1.2. (Restriction Principle [7]) *Let $\mathbb{V}$ be a zigzag module that decomposes into:*

$$\mathbb{V} \cong \bigoplus_{j \in J} \mathbb{I}[b_i; d_i]$$

then the restriction $\mathbb{V}[b; d]$ decomposes into:

$$\mathbb{V}[b; d] \cong \bigoplus_{j \in J} \mathbb{I}\left([b_i; d_i] \cap [b; d]\right)$$

where $\mathbb{I}([b_i; d_i] \cap [b; d])$ is the interval module over the interval $[b_i; d_i] \cap [b; d]$ if $[b_i; d_i] \cap [b; d] \neq \emptyset$ and is the 0 module otherwise.

A *subrepresentation* $\mathbb{U}$ of an $\mathbb{F}$-representation $\mathbb{V} = (V_i, f_i/g_i)$ of $\mathcal{Q}$ is a representation $(U_i, f_i'/g_i')$ of $\mathcal{Q}$ such that U_i is a subspace of V_i for every index i and f_i' (respectively g_i') is the restriction of the linear map f_i (resp. g_i) to the subspace U_i (resp. U_{i+1}). In computational topology, a (sub-)representation of an

A_n-type quiver is also called a *zigzag (sub-)module*; in the following, we use both terms indifferently.

Let $\mathbb{V} = (V_i, f_i/g_i)$ and $\mathbb{W} = (W_i, r_i/s_i)$ be two $\mathbb{F}$-representations of a quiver $\mathcal{Q}$. A *morphism of representations* $\phi\colon \mathbb{V} \to \mathbb{W}$ is a set of linear maps $\{\phi_i : V_i \to W_i\}_{i=1\ldots n}$ such that, for every arrow either one of the following diagrams commutes:

$$\begin{tikzcd} V_i \arrow[r, "f_i"] \arrow[d, "\phi_i"'] & V_{i+1} \arrow[d, "\phi_{i+1}"] \\ W_i \arrow[r, "r_i"] & W_{i+1} \end{tikzcd} \qquad \begin{tikzcd} V_i \arrow[d, "\phi_i"'] & V_{i+1} \arrow[l, "g_i"'] \arrow[d, "\phi_{i+1}"] \\ W_i & W_{i+1} \arrow[l, "s_i"] \end{tikzcd}$$

The morphism is called an *isomorphism*, denoted by $\cong$, if every ϕ_i is bijective. Finally, the $\mathbb{F}$-representations of $\mathcal{Q}$ admit a *direct sum* denoted by $\oplus$. For any $\mathbb{V} = (V_i, f_i/g_i), \mathbb{W} = (W_i, r_i/s_i)$, define the $\mathbb{F}$-representation $\mathbb{V}\oplus\mathbb{W}$ to be the representation of $\mathcal{Q}$ with spaces $V_i \oplus W_i$, for all i, and maps $f_i \oplus r_i = \left(\begin{smallmatrix} f_i & 0 \\ 0 & r_i \end{smallmatrix}\right)$ or $g_i \oplus s_i = \left(\begin{smallmatrix} g_i & 0 \\ 0 & s_i \end{smallmatrix}\right)$ for every arrow. An $\mathbb{F}$-representation $\mathbb{V}$ is *decomposable* if it admits two non-zero subrepresentations $\mathbb{U}_1$ and $\mathbb{U}_2$ such that $\mathbb{V} = \mathbb{U}_1 \oplus \mathbb{U}_2$. It is otherwise *indecomposable*. Theorem 1.1 states that the indecomposable representations of an A_n-type quiver are the interval modules.

Representative Sequences. We now introduce the main low-level object used to prove theorems and the validity of our algorithm.

Definition 1.1. *For an $\mathbb{F}$-representation $\mathbb{V} = (V_i, f_i/g_i)$ of an A_n-type quiver $\mathcal{Q}$, a* representative sequence, *denoted by $u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}$, is an n-tuple $(u^{(1)}, \ldots, u^{(n)}) \in V_1 \times \cdots \times V_n$ such that:*

(a) The index b, called the birth index, *satisfies either $b = 1$, or $\bullet_{b-1} \to \bullet_b$ is forward, or $g_{b-1}(u^{(b)}) = 0$. In addition, $u^{(i)} = 0$ for every $i < b$.*

(b) The index d, called the death index, *satisfies either $d = n$, or $\bullet_d \leftarrow \bullet_{d+1}$ is backward, or $f_d(u^{(d)}) = 0$. In addition, $u^{(i)} = 0$ for every $i > d$.*

(c) For all i, $b \le i < d$, either $f_i(u^{(i)}) = u^{(i+1)}$ or $u^{(i)} = g_i(u^{(i+1)})$, depending on the direction of the arrow $\bullet_i \leftrightarrow \bullet_{i+1}$. In addition, $u^{(i)} \neq 0$ for every i, $b \le i \le d$.

The following easy propositions relates the representative sequences to the so-called *interval submodules* of $\mathbb{V}$, i.e. the submodules of $\mathbb{V}$ that are isomorphic to interval representations as in (1.3):

Proposition 1.1. *$\mathbb{U}$ is an interval submodule of $\mathbb{V}$ if and only if there exists a representative sequence $u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}$ such that $\mathbb{U}$ is equal to:*

$$0 \leftrightarrow \cdots 0 \leftrightarrow \langle u^{(b)} \rangle \leftrightarrow \cdots \leftrightarrow \langle u^{(d)} \rangle \leftrightarrow 0 \cdots \leftrightarrow 0$$

If an interval submodule of $\mathbb{V}$ is in direct sum, we call it an *interval summand.* As a direct consequence of the definitions, we deduce:

Proposition 1.2. *Let $\mathbb{V} = (V_i, f_i/g_i)$ be a representation of an A_n-type quiver and $\{ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)} \}_{j \in J}$ be a family of representative sequences for $\mathbb{V}$. If, for every index $i \in \{1, \ldots, n\}$, the set $\{u_j^{(i)} : u_j^{(i)} \neq 0\}_{j\in J}$ is a basis for V_i, then:*

$$\mathbb{V} \cong \bigoplus_{j \in J} \mathbb{I}[b_j; d_j]$$

In this case, we say that the family of representative sequences represents an interval decomposition *for* $\mathbb{V}$.

Arithmetic of Representative Sequences. Let $\mathbb{V} = (V_i, f_i/g_i)$ be a representation of an A_n-type quiver.

Definition 1.2. *Let $\mathbf{u} = u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}$ and $\mathbf{v} = v^{(b')} \leftrightarrow \cdots \leftrightarrow v^{(d')}$ be two representative sequences. Define $b_m = \max_{\le_{\mathbf{b}}}\{b, b'\}$ and $d_m = \max_{\le_{\mathbf{d}}}\{d, d'\}$. If the two sequences satisfy:*

(a) $[b; d] \cap [b'; d'] \neq \emptyset$,

(b) for all $j \in [b; d] \cap [b'; d']$, the vectors $u^{(j)}$ and $v^{(j)}$ are linearly independent in V_j,

(c) $b_m \le d_m$,

then we define the binary operator $*$:

$$(u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}) * (v^{(b')} \leftrightarrow \cdots \leftrightarrow v^{(d')}) \quad := \quad u^{(b_m)} + v^{(b_m)} \leftrightarrow \cdots \leftrightarrow u^{(d_m)} + v^{(d_m)}$$

of birth b_m and death d_m. We also define, for any representative sequence $u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}$ and scalar $\gamma \in \mathbb{F}$, not equal to 0, the scalar multiplication:

$$\gamma \cdot (u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}) := \gamma u^{(b)} \leftrightarrow \cdots \leftrightarrow \gamma u^{(d)}$$

Lemma 1.1. *For two representative sequences $\mathbf{u}$ and $\mathbf{v}$ satisfying conditions (a), (b) and (c) above, and a non-zero scalar γ, $\mathbf{u} * \mathbf{v}$ and $\gamma \cdot \mathbf{u}$ are representative sequences.*

Proof. The case of $\gamma \cdot \mathbf{u}$ is direct. Denote $\mathbf{u} = u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(d)}$ and $\mathbf{v} = v^{(b')} \leftrightarrow \cdots \leftrightarrow v^{(d')}$. We prove that $\mathbf{u} * \mathbf{v}$ is a representative sequence. First, we prove that definition 1.1 *(a)* is satisfied. If $b_{\mathrm{m}} = 1$ or $\bullet_{b_{\mathrm{m}}-1} \to \bullet_{b_{\mathrm{m}}}$ is forward, *(a)* is satisfied. Suppose now that $b_{\mathrm{m}} > 1$ and that the arrow $\bullet_{b_{\mathrm{m}}-1} \leftarrow \bullet_{b_{\mathrm{m}}}$ is backward. We prove that $g_{b_{\mathrm{m}}-1}(u^{(b_{\mathrm{m}})} + v^{(b_{\mathrm{m}})}) = 0$. Suppose, w.l.o.g., that $b_{\mathrm{m}} = b$. Hence $b' \le_{\mathbf{b}} b$ which implies, together with $\bullet_{b_{\mathrm{m}}-1} \leftarrow \bullet_{b_{\mathrm{m}}}$ being backward:

- either $b = b'$, in which case $g_{b_{\mathrm{m}}-1}(u^{(b_{\mathrm{m}})}) = g_{b_{\mathrm{m}}-1}(v^{(b_{\mathrm{m}})}) = 0$,

- or $b > b'$, in which case $g_{b_{\mathrm{m}}-1}(u^{(b_{\mathrm{m}})}) = 0$ and $v^{(b_{\mathrm{m}})} = 0$.

Definition 1.1 (b) is satisfied with a similar proof.

We prove now that definition 1.1 *(c)* is satisfied. Definition 1.2*(c)* ensures that the interval $[b_{\mathrm{m}}; d_{\mathrm{m}}]$ is not empty and definition 1.2*(a)* implies that $[b_{\mathrm{m}}; d_{\mathrm{m}}] \subseteq [b; d] \cup [b'; d']$. As a consequence, for any index $j \in [b_{\mathrm{m}}; d_{\mathrm{m}}]$, $u^{(j)}$ and $v^{(j)}$ are not both equal to 0 and, by virtue of definition 1.2*(b)*, $u^{(j)} + v^{(j)} \neq 0$. Finally, we verify that for all $j \in [b_{\mathrm{m}}; d_{\mathrm{m}}]$, $f_j(u^{(j)} + v^{(j)}) = u^{(j+1)} + v^{(j+1)}$ or $g_j(u^{(j+1)} + v^{(j+1)}) = u^{(j)} + v^{(j)}$. For every $j \in [b_{\mathrm{m}}; d_{\mathrm{m}}] \backslash \{b-1, b'-1, d, d'\}$, definition 1.1 *(c)* is satisfied by linearity of f_j and g_j. Suppose, w.l.o.g., that the index b is contained within $(b_{\mathrm{m}}; d_{\mathrm{m}}]$. Hence, $b' < b$ and $b \leq_{\mathbf{b}} b'$, which implies that $\bullet_{b-1} \leftarrow \bullet_b$ is backward. Consequently, $g_{b-1}(u^{(b)}) = 0$ and $g_{b-1}(u^{(b)} + v^{(b)}) = u^{(b-1)} + v^{(b-1)}$. The case of d contained within $[b_{\mathrm{m}}; d_{\mathrm{m}})$ is similar.

□

The following lemma gives some properties of " $\cdot$ " and " $*$ ":

Lemma 1.2. *Let $\mathbf{u}, \mathbf{v}$ and $\mathbf{w}$ be three representative sequences pairwise satisfying conditions (a), (b) and (c) of definition 1.2. We have:*

1. $\mathbf{u} * \mathbf{v} = \mathbf{v} * \mathbf{u}$ *(commutativity),*

2. $(\mathbf{u} * \mathbf{v}) * \mathbf{w} = \mathbf{u} * (\mathbf{v} * \mathbf{w})$ *(associativity),*

3. $\gamma \cdot (\delta \cdot \mathbf{u}) = (\gamma\delta) \cdot \mathbf{u}$,

4. $1 \cdot \mathbf{u} = \mathbf{u}$.

Proof. The commutativity (1.) is direct. Properties (3.) and (4.) follow directly from the definition of the multiplication " $\cdot$ ". We prove the associativity (2.). First, we prove that the sequences $(\mathbf{u} * \mathbf{v})$ and $\mathbf{w}$ satisfy conditions *(a)*, *(b)* and *(c)* of definition 1.2. Denote by $b_{\mathbf{u}}$, $b_{\mathbf{v}}$ and $b_{\mathbf{w}}$ the births of $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ respectively, and $d_{\mathbf{u}}$, $d_{\mathbf{v}}$ and $d_{\mathbf{w}}$ their deaths. The fact that the pairwise intersections of the intervals $[b_{\mathbf{u}}; d_{\mathbf{u}}]$, $[b_{\mathbf{v}}; d_{\mathbf{v}}]$ and $[b_{\mathbf{w}}; d_{\mathbf{w}}]$ is non-empty implies that their common intersection is non-empty[2]. Hence definition 1.2 *(a)* is satisfied. Definition 1.2 *(b)* and *(c)* are directly inherited from the fact they are pairwise satisfied by $b_{\mathbf{u}}$, $b_{\mathbf{v}}$ and $b_{\mathbf{w}}$. By symmetry, the result applies to the sequences $\mathbf{u}$ and $\mathbf{v} * \mathbf{w}$.

Finally, associativity follows from the associativity of $+$ in each vector space and the associativity of the functions $\max_{\leq_{\mathbf{b}}}$ and $\max_{\leq_{\mathbf{d}}}$.

□

For example, consider the following representation $\mathbb{V}$ of the quiver of A_6-type presented above:

$$\begin{tikzcd} \mathbb{F} \arrow[r, "\mathbb{1}"] & \mathbb{F} \arrow[r, "\mathbb{1}"] & \mathbb{F} & \mathbb{F}^2 \arrow[l, "\pi_1"'] \arrow[r, "\iota_z"] & \mathbb{F}^3 & \mathbb{F}^2 \arrow[l, "\iota_y"'] \\ x \arrow[r, mapsto] & x \arrow[r, mapsto] & x & (x,y) \arrow[l, mapsto] \arrow[r, mapsto] & (x,y,z) & (x,z) \arrow[l, mapsto] \end{tikzcd}$$

Naturally, $\mathbb{V} \cong \mathbb{I}[1;6] \oplus \mathbb{I}[4;5] \oplus \mathbb{I}[5;6]$. Let $x^{(1)} \leftrightarrow \cdots \leftrightarrow x^{(6)}$ and $y^{(4)} \leftrightarrow \cdots \leftrightarrow y^{(5)}$ be representative sequences for submodules respectively isomorphic to $\mathbb{I}[1;6]$ and $\mathbb{I}[4;5]$. We have:

$$\begin{aligned} &(\ x^{(1)} \leftrightarrow \cdots \leftrightarrow x^{(6)}\ ) * (\ y^{(4)} \leftrightarrow \cdots \leftrightarrow y^{(5)}\ ) = \\ &(x+y)^{(1)} \leftrightarrow \cdots \leftrightarrow (x+y)^{(5)} \end{aligned}$$

In the following, the operators " $*$ " and " $\cdot$ " offer tools to manipulate interval submodules of an $\mathbb{F}$-representation.

2 Diamond Principles.

In this section we relate the interval decompositions of two representations $\mathbb{V}$ and $\mathbb{W}$ related by a local change, called a *diamond*. We recall the *Exact Diamond Principle* of [7] and introduce the main theoretical results of the article, specifically the *Injective and Surjective Diamond Principles* and the *Transposition Diamond Principle*.

2.1 Exact Diamonds.

Consider the diagram:

(2.5)

$$\begin{tikzcd}[row sep=small] \mathbb{W} := & & & W_i & & & \\ & V_1 \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_{i-1} \arrow[u, "b"] & & V_{i+1} \arrow[ul, "d"'] \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_n \\ \mathbb{V} := & & & & V_i \arrow[ul, "a"] \arrow[ur, "c"'] & & \end{tikzcd}$$

We say that the following diagram:

(2.6)
$$\begin{tikzcd} V_{i+1} \arrow[r, "d"] & W_i \\ V_i \arrow[u, "c"] \arrow[r, "a"] & V_{i-1} \arrow[u, "b"'] \end{tikzcd}$$

is *exact* [7] if $\operatorname{im} D_1 = \ker D_2$ in the sequence $V_i \xrightarrow{D_1} V_{i-1} \oplus V_{i+1} \xrightarrow{D_2} W_i$, where $D_1(v) = (a(v), c(v))$ and $D_2(x,y) = b(x) - d(y)$. Note in particular that an exact diamond commutes, i.e. $b \circ a = d \circ c$.

If diagram 2.6 is exact we say that the representations $\mathbb{V}$ and $\mathbb{W}$ are related by an *exact diamond at index i*. We recall the *Exact Diamond Principle*:

[2] This is a particular case of Helly's theorem.

Theorem 2.1. (Exact Diamond Principle [7]) *Given $\mathbb{V}$ and $\mathbb{W}$ related by an exact diamond at index i, there is a partial bijection between the intervals of the decompositions of $\mathbb{V}$ and $\mathbb{W}$:*

- *intervals $\mathbb{I}[i;i]$ are unmatched,*
- *for $b < i$, intervals $\mathbb{I}[b;i]$ are matched with intervals $\mathbb{I}[b;i-1]$ and vice versa,*
- *for $d > i$, intervals $\mathbb{I}[i;d]$ are matched with intervals $\mathbb{I}[i+1;d]$ and vice versa,*
- *intervals $\mathbb{I}[b;d]$ are matched with intervals $\mathbb{I}[b;d]$ in all other cases.*

2.2 Injective and Surjective Diamonds. For simplicity of exposition, we assume in the following that every representation $\mathbb{U}$ has an interval decomposition that satisfies the following:

- for every index $i > 1$, there is at most one interval with birth i,
- for every index $j < n$, there is at most one interval with death j.

These conditions are in particular satisfied in zigzag persistent homology (section 3).

We relate the interval decompositions of the bottom representation $\mathbb{V}$ and top representation $\mathbb{W}$ of the following diagram:

(2.7)

$$\begin{tikzcd} \mathbb{W} := & & & & W & & & \\ & V_1 \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V \arrow[ur, "f"] & & V \arrow[ul, "f"'] \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_n \\ \mathbb{V} := & & & & V \arrow[ul, "\mathbb{1}"] \arrow[ur, "\mathbb{1}"'] & & & \end{tikzcd}$$

where the diamond is located at index i in the module. We distinguish the case where f is injective of corank 1 and the case where f is surjective of nullity 1.

Theorem 2.2. (Injective Diamond Principle) *Suppose f is injective of corank 1. Then,*

$$\mathbb{W} \cong \mathbb{V} \oplus \mathbb{I}[i;i]$$

Proof. $\mathbb{V}$ and $\mathbb{W}$ are related by an exact diamond. Indeed, by injectivity of f, $\text{im } D_1 = \ker D_2$ in the sequence $V \xrightarrow{D_1} V \oplus V \xrightarrow{D_2} W$ with $D_1 : v \mapsto (v, v)$ and $D_2 : (x, y) \mapsto f(x) - f(y)$. The nature of the maps and the partial bijection of the exact diamond principle 2.1 imply that there is only an extra summand $\mathbb{I}[i;i]$ in the decomposition of $\mathbb{W}$. $\square$

When f is surjective, the diamond is not exact as $\ker f \neq \{0\}$. For example, in the sequence $V \xrightarrow{D_1} V \oplus V \xrightarrow{D_2} W$, the couple $(u, 0)$ belongs to $\ker D_2 \setminus \text{im } D_1$ for any $u \in \ker f$, $u \neq 0$. Formulating the *Surjective Diamond Principle* requires an explicit expression of the interval submodules of the decomposition, using representative sequences. Recall that $\leq_{\mathbf{b}}$ and $\leq_{\mathbf{d}}$ are some prescribed orders on the indices $\{1, \ldots, n\}$ that depend only on the sequence of arrow orientations in the zigzags. In the following, $\leq_{\mathbf{b}}$ and $\leq_{\mathbf{d}}$ are defined on the quiver of the bottom diagram.

Theorem 2.3. (Surjective Diamond Principle) *Suppose f is surjective of nullity 1, and let ξ be a vector generating its kernel. Let $\{ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)} \}_{j \in J}$ be a family of representative sequences representing an interval decomposition of $\mathbb{V}$. Up to a reordering of the indices in J, write ξ as:*

$$\xi = \alpha_1 u_1^{(i)} + \ldots + \alpha_p u_p^{(i)},$$

with $\alpha_j \neq 0$ for every $1 \leq j \leq p$ and $d_1 \leq_{\mathbf{d}} \ldots \leq_{\mathbf{d}} d_p$. Letting $b_{\ell_p} = \max_{\leq_{\mathbf{b}}} \{b_j\}_{j=1,\ldots,p}$, the modules $\mathbb{V}$ and $\mathbb{W}$ admit the following interval decompositions:

$$\begin{array}{rcl} \mathbb{V} & \cong & \mathbb{U} \oplus \bigoplus_{1 \leq j \leq p} \mathbb{I}[b_j; d_j] \qquad \text{and} \\ \mathbb{W} & \cong & \mathbb{U} \oplus \mathbb{I}[b_{\ell_p}; i-1] \oplus \mathbb{I}[i+1; d_p] \oplus \\ & & \bigoplus_{j=1}^{p-1} \mathbb{I}[b_{\ell_j}; d_j] \end{array}$$

where the pairing $(b_{\ell_j}, d_j)_{1 \leq j \leq p-1}$ is computed as follows (assuming b_{ℓ_p} and d_p are considered as already "paired"):

Algorithm 1: Pairing for Surjective Diamond

for *j from 1 to $p-1$* **do**
  if *b_j not yet paired* **then**
    $b_{\ell_j} \leftarrow b_j$; pair b_{ℓ_j} with d_j;
  end
  else
    $b_{\ell_j} \leftarrow \max_{\leq_{\mathbf{b}},\, k=1,\ldots,p} \{b_k : b_k \text{ not yet paired}\}$;
    pair b_{ℓ_j} with d_j;
  end
end

Sketch of Proof. The proof of the surjective diamond principle consists in working at the level of representative sequences in order to explicitly construct an internal direct sum decomposition of $\mathbb{W}$, from the one of $\mathbb{V}$. Intuitively, the underlying idea of the proof is to split the modules $\mathbb{V}$ and $\mathbb{W}$ at the index i of the

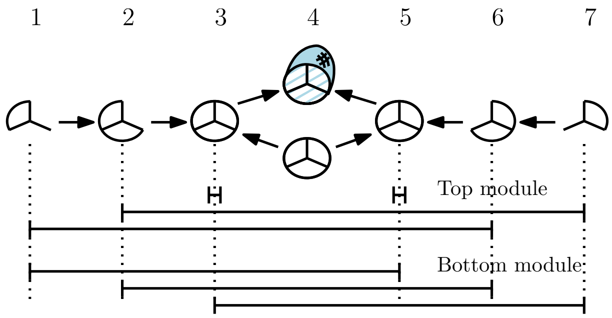

Figure 1: Two zigzags in computational topology, related by a surjective diamond at index 4 in 1-homology, and their interval decompositions. The decompositions are pictured as barcodes, where a bar from index i to index j represents an interval summand $\mathbb{I}[i;j]$.

surjective diamond, in order to consider their prefix $\mathbb{V}[1;i-1] = \mathbb{W}[1;i-1]$ and suffix $\mathbb{V}[i+1;n] = \mathbb{W}[i+1;n]$ separately. We manipulate the restricted representative sequences using $*$ and the scalar multiplication " $\cdot$ " in order to "align" them with the kernel of f, and we join them back to represent submodules of $\mathbb{W}$. We prove the surjective diamond principle in section 7.

$\square$

Consider the example, in computational topology, depicted in figure 1. The bottom and top zigzags of the diagram are related by a surjective diamond reflection and their interval decompositions at the homology level (dimension 1) are presented. Following the intuition, the decomposition for the bottom diagram corresponds to the addition and removal of each circular arc: for example, the birth of the closed interval $\mathbb{I}[2;6]$ agrees with the insertion of the bottom arc (creating a "hole"), and its death agrees with the removal of the bottom arc when following the backward arrow $\bullet_6 \leftarrow \bullet_7$. By contrast, inserting the cap on top of the outer circle in the top zigzag links the three holes together and leads to a new pairing of the births and deaths of the corresponding intervals, according to theorem 2.3. In this example, the order relation $\leq_{\mathbf{b}}$ satisfies $7 \leq_{\mathbf{b}} 6 \leq_{\mathbf{b}} 4 \leq_{\mathbf{b}} 1 \leq_{\mathbf{b}} 2 \leq_{\mathbf{b}} 3 \leq_{\mathbf{b}} 5$ and $\leq_{\mathbf{d}}$ satisfies $1 \leq_{\mathbf{d}} 2 \leq_{\mathbf{d}} 4 \leq_{\mathbf{d}} 7 \leq_{\mathbf{d}} 6 \leq_{\mathbf{d}} 5 \leq_{\mathbf{d}} 3$. The arrow reflection first results in the apparition of intervals $\mathbb{I}[3;3]$ and $\mathbb{I}[5;5]$ (intervals $\mathbb{I}[b_{\ell_p};i-1]$ and $\mathbb{I}[i+1;d_p]$ of the theorem). This induces a redistribution of birth and death indices. The interval dying at index 7 was born initially at index 3, which is already used in $\mathbb{I}[3;3]$. It therefore gets assigned the largest available birth index w.r.t. $\leq_{\mathbf{b}}$, which is 2. Similarly, the interval dying at index 6 gets assigned 1 as new birth index.

Transposition Diamonds. Consider the diagram:

(2.8)

$$\begin{tikzcd}[row sep=small] \mathbb{W} := & & & W_i \arrow[dr, "d"] & & & \\ & V_1 \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_{i-1} \arrow[u, "b"] \arrow[d, "a"'] & V_{i+1} \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_n \\ \mathbb{V} := & & & V_i \arrow[ur, "c"'] & & & \end{tikzcd}$$

We say that the representations $\mathbb{V}$ and $\mathbb{W}$ are related by a *transposition diamond* if the following diagram is *exact*:

$$\begin{tikzcd} W_i \arrow[r, "d"] & V_{i+1} \\ V_{i-1} \arrow[u, "b"] \arrow[r, "a"] & V_i \arrow[u, "c"'] \end{tikzcd}$$

$V_{i-1} \xrightarrow{D_1} V_i \oplus W_i \xrightarrow{D_2} V_{i+1}$ with $D_1(v) = (a(v), b(v))$ and $D_2(x,y) = c(x) - d(y)$ such that $\text{im } D_1 = \ker D_2$

Note that the transposition diamond diagram (2.8) is similar to the exact diamond diagram (2.5) except that the diamond is "rotated by 90^o".

Theorem 2.4. (Transposition Diamond Principle) *Given $\mathbb{V}$ and $\mathbb{W}$ related by a transposition diamond as above, we assume that the maps a,b,c,d are of two different types: injective of corank 1 and surjective of nullity 1. We have:*

1. *if a and c surjective of nullity 1 then $\mathbb{V} \cong \mathbb{U} \oplus \mathbb{I}[b;i-1] \oplus \mathbb{I}[b';i]$ for some indices $b,b' \leq i-1$. Let $(\ldots,u,0,0\ldots)$ and $(\ldots v, a(v), 0\ldots)$, $u,v \in V_{i-1}$, be representative sequences for the interval summands $\mathbb{I}[b;i-1]$ and $\mathbb{I}[b';i]$ respectively. There exists $\alpha \in \mathbb{F}$ such that $v + \alpha u \in \ker b$ and:*

(i) if $\alpha = 0$ then $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[b;i] \oplus \mathbb{I}[b';i-1]$,

(ii) if $\alpha \neq 0$ then $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[\max_{\leq_{\mathbf{b}}}\{b,b'\};i-1] \oplus \mathbb{I}[\min_{\leq_{\mathbf{b}}}\{b,b'\};i]$.

2. *if a and c injective of corank 1 then $\mathbb{V} \cong \mathbb{U} \oplus \mathbb{I}[i;d] \oplus \mathbb{I}[i+1;d']$ for some indices $d,d' \geq i+1$. Let $(0\ldots 0, v, c(v), \ldots)$ and $(0\ldots 0, 0, u, \ldots)$, $v \in V_i$ and $u \in V_{i+1}$, be representative sequences for the interval summands $\mathbb{I}[i;d]$ and $\mathbb{I}[i+1;d']$ respectively. There exists $\alpha \in \mathbb{F}$ such that $u + \alpha c(v) \in \text{im } d$ and:*

(i) if $\alpha = 0$ then $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[i+1;d] \oplus \mathbb{I}[i;d']$,

(ii) if $\alpha \neq 0$ then $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[i;\max_{\leq_{\mathbf{d}}}\{d,d'\}] \oplus \mathbb{I}[i+1;\min_{\leq_{\mathbf{d}}}\{d,d'\}]$.

3. *if a injective of corank 1 and c surjective of nullity 1 then:*
$\mathbb{V} \cong \mathbb{U} \oplus \mathbb{I}[i;d] \oplus \mathbb{I}[b;i]$ *and* $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[i+1;d] \oplus \mathbb{I}[b;i-1]$.

4. *if a surjective of nullity 1 and c injective of corank 1 then:*
$\mathbb{V} \cong \mathbb{U} \oplus \mathbb{I}[i+1;d] \oplus \mathbb{I}[b;i-1]$ *and* $\mathbb{W} \cong \mathbb{U} \oplus \mathbb{I}[i;d] \oplus \mathbb{I}[b;i]$.

Sketch of Proof. The proof of the transposition diamond principle consists in turning a family of representative sequences that represents an interval decomposition of $\mathbb{V}$ into a family of representative sequences that represents an interval decomposition of $\mathbb{W}$. Specifically, every representative sequence for an interval summand of $\mathbb{V}$, with birth b and death d such that $b \notin \{i; i+1\}$ and $d \notin \{i-1, i\}$, is matched via a natural map with a representative sequence for an interval summand of $\mathbb{W}$ with same birth and death. Cases 1.*(i)*, 2.*(i)*, 3. and 4. consider the cases where two intervals with birth in $\{i; i+1\}$ and death in $\{i-1; i\}$ exchange their endpoints. Finally, the cases 1.*(ii)* and 2.*(ii)* are more intricate: given two representative sequences $\mathbf{u}$ and $\mathbf{v}$ with births in $\{i; i+1\}$ or deaths in $\{i-1; i\}$, we form the representative sequence $\mathbf{u} * (\alpha \cdot \mathbf{v})$. This explains the use of $\max_{\leq_{\mathbf{b}}}$ and $\max_{\leq_{\mathbf{d}}}$ to describe the birth and death of the new interval. We prove the transposition diamond principle in section 6. □

3 Zigzag Persistent Homology Algorithm.

We assume familiarity with homology theory, referring the reader to [21] for an introduction. In the following we use simplicial homology with coefficients in a field $\mathbb{F}$.

A *simplicial complex* $\mathbf{K}$ on a finite set of *vertices* V is a collection of simplices $\{\sigma\}$, $\sigma \subseteq V$, such that $\tau \subseteq \sigma \in \mathbf{K} \Rightarrow \tau \in \mathbf{K}$. The dimension $d = |\sigma| - 1$ of σ is its number of elements minus 1. The group of d-chains, denoted $\mathbf{C}_d(\mathbf{K})$, of $\mathbf{K}$ is the group of formal sums of d-simplices with $\mathbb{F}$ coefficients. The *boundary operator* is a linear operator $\partial_d : \mathbf{C}_d(\mathbf{K}) \to \mathbf{C}_{d-1}(\mathbf{K})$ such that $\partial_d \sigma = \partial_d [v_0, \cdots, v_d] = \sum_{i=0}^{d} (-1)^i [v_0, \cdots, \widehat{v_i}, \cdots, v_d]$, where $\widehat{v_i}$ means v_i is deleted from the list. The kernel of ∂_d, denoted by $\mathbf{Z}_d(\mathbf{K})$, is the group of d-*cycles* and the image of ∂_d, denoted by $\mathbf{B}_{d-1}(\mathbf{K})$, is the group of $(d-1)$-*boundaries*. Observing $\partial_d \circ \partial_{d+1} = 0$, the d^{th} homology group $\mathbf{H}_d(\mathbf{K})$ of $\mathbf{K}$ is defined to be the quotient $\mathbf{H}_d(\mathbf{K}) = \mathbf{Z}_d(\mathbf{K}) / \mathbf{B}_d(\mathbf{K})$. We drop the dimension d by considering the external direct sum $\mathbf{C}(\mathbf{K}) = \bigoplus_d \mathbf{C}_d(\mathbf{K})$ and the boundary operator $\partial : \mathbf{C}(\mathbf{K}) \to \mathbf{C}(\mathbf{K})$ extended by linearity. We define $\mathbf{Z}(\mathbf{K})$, $\mathbf{B}(\mathbf{K})$ and $\mathbf{H}(\mathbf{K})$ similarly. Because the coefficients belongs to a field $\mathbb{F}$, these are vector spaces, and since $\mathbf{K}$ is a finite simplicial complex, they have finite dimensions. In this context, the insertion of a simplex σ of dimension d in a simplicial complex $\mathbf{K}$, $\mathbf{K} \xrightarrow{(\sigma)} \mathbf{K} \cup \{\sigma\}$ may either create a homology class in dimension d, or destroy a homology class in dimension $d-1$. In the first case, the map induced at homology level is injective of corank 1. In the second case, it is surjective of nullity 1, and its kernel is spanned by $[\partial\sigma]$.

For ease of exposition, throughout the main body of the paper we assume the field of coefficients to be $\mathbb{Z}_2$. Nevertheless, our approach is not tied to $\mathbb{Z}_2$, and the proofs of the diamonds principles are written for an arbitrary field of coefficients $\mathbb{F}$.

Zigzag Filtrations. A *zigzag filtration* on an A_n-type quiver $\mathcal{Q}$ is an assignment of a simplicial complex $\mathbf{K}_i$ for each vertex $\bullet_i$, and of an elementary inclusion corresponding either to a simplex insertion $\mathbf{K}_i \xrightarrow{(\sigma)} \mathbf{K}_{i+1}$ (i.e. $\mathbf{K}_{i+1} = \mathbf{K}_i \cup \{\sigma\}$) or to a simplex deletion $\mathbf{K}_i \xleftarrow{(\sigma)} \mathbf{K}_{i+1}$ (i.e. $\mathbf{K}_{i+1} = \mathbf{K}_i \setminus \{\sigma\}$) for each arrow $\bullet_i \longleftrightarrow \bullet_{i+1}$. A zigzag filtration $\mathbb{K}$ induces a $\mathbb{Z}_2$-representation $\mathbf{H}(\mathbb{K})$ of $\mathcal{Q}$ at the homology level, whose homology groups and linear maps are induced by the simplicial complexes and elementary inclusions. Computing zigzag persistence consists in computing the interval decomposition of this zigzag module $\mathbf{H}(\mathbb{K})$, given the sequence of simplex insertions and deletions. Note that for an index $i \in \{2, \cdots, n\}$ there is at most one interval with birth i in the interval decomposition of $\mathbf{H}(\mathbb{K})$, and for an index $i \in \{1, \cdots, n-1\}$ there is at most one interval with death i. There may be as many births equal to 1 and deaths equal to n. This is true generally when the maps between the simplicial complexes are elementary inclusions. We call *standard persistence* the case where all arrows are oriented in the same direction. In this case, $\mathbb{K}$ is called a *standard filtration*.

Standard Persistence and Matrix Reduction. We review the presentation of standard persistence as in [15], where explicit chains representing a compatible homology basis are maintained. For a standard filtration $\mathbb{K} = \mathbf{K}_m \xleftarrow{\tau_m} \cdots \xleftarrow{\tau_2} \mathbf{K}_1 \xleftarrow{\tau_1} \emptyset$, there exist chains $\widehat{\tau}_m, \cdots, \widehat{\tau}_1 \in \mathbf{C}(\mathbf{K}_m)$, a partition of the indices $\{1, \cdots, m\} = F \sqcup G \sqcup H$, and a bijective pairing $G \leftrightarrow H$, denoted by $\mathcal{P} \subseteq G \times H$, satisfying the following conditions:

1. for all i, $\mathbf{C}(\mathbf{K}_i) = \langle \widehat{\tau}_i, \cdots, \widehat{\tau}_1 \rangle$,
2. for all $f \in F$, $\partial \widehat{\tau}_f = 0$, and
3. for all pairs $(g, h) \in \mathcal{P}$, $\partial \widehat{\tau}_h = \widehat{\tau}_g$ and hence $\partial \widehat{\tau}_g = 0$.

We call such a partitioned set of chains an *encoding of the persistence module*. Condition 1. is equivalent to the fact that every $\widehat{\tau}_i$ admits τ_i as leading term (i.e. $\widehat{\tau}_i = \varepsilon_1 \tau_1 + \cdots + \varepsilon_i \tau_i$, with $\varepsilon_i \neq 0$). According to Theorem 2.6 of [15], an encoding of a standard persistence module encodes completely its interval decomposition. Indeed, for $f \in F$, $\widehat{\tau}_f$ is a cycle created at index f and whose homology class is non-zero in $\mathbf{K}_m$. For $g \in G$, paired with $h \in H$, $\widehat{\tau}_g$ is a cycle created at index g and whose homology class is non-zero from index g up to index $h-1$, after which it becomes the boundary of the chain $\widehat{\tau}_h$. One can read directly the persistent interval from this encoding. Indeed, let $\mathbf{H}(\mathbb{K}) = \mathbf{H}(\mathbf{K}_m) \longleftarrow \cdots \longleftarrow \mathbf{H}(\mathbf{K}_1) \longleftarrow 0$ be the corresponding persistence module. The representative sequences induced by the cycles $\widehat{\tau}_f$ for $f \in F$ and $\widehat{\tau}_g$ for $g \in G$ are respectively:

$$[\widehat{\tau}_f]^{(m)} \leftrightarrow \cdots \leftrightarrow [\widehat{\tau}_f]^{(f)} \text{ and } [\widehat{\tau}_g]^{(h-1)} \leftrightarrow \cdots \leftrightarrow [\widehat{\tau}_g]^{(g)}$$

where $[\widehat{\tau}_i]^{(j)}$ refers to the homology class of $\widehat{\tau}_i$ in the simplicial complex $\mathbf{K}_j$. This is well-defined because $\widehat{\tau}_i$ has τ_i as leading term and $j \geq i$ in the previous sequences. Moreover, these homology classes are pointwise independent. By virtue of proposition 1.2, these sequences represent the interval decomposition[3] of the zigzag module:

$$\mathbf{H}(\mathbf{K}) \cong \bigoplus_{f \in F} \mathbb{I}[m; f] \bigoplus_{(g,h) \in \mathcal{P}} \mathbb{I}[h-1; g]$$

In the following, we represent an encoding by an $(m \times m)$-matrix[4] $\mathbf{M}$ with $\mathbb{Z}_2$ coefficients, where each column j, denoted by col_j, represents the chain $\widehat{\tau}_j$ in the basis $\{\tau_1, \cdots, \tau_m\}$ of $\mathbf{C}(\mathbf{K}_m)$. Due to Condition 1., $\mathbf{M}$ is upper-triangular, with non-zero elements on the diagonal. For a non-zero column col, we denote by $\mathrm{low}(\mathsf{col})$ the row index of its lowest non-zero element. If the column is null, low is undefined.

For any chain $c \in \mathbf{C}(\mathbf{K}_m)$, represented as a column col in $\mathbf{M}$, and for any set of indices $I \subseteq \{1, \cdots, m\}$, we can express c as a linear combination of the chains $\{\widehat{\tau}_i\}_{i \in I}$ (whenever possible) using the following reduction:

Algorithm 2: `Reduction(col,I)`

while $\exists i_0 \in I$ *with* $\mathrm{low}(\mathsf{col}_{i_0}) = \mathrm{low}(\mathsf{col})$ **do**
  $\mathsf{col} \leftarrow \mathsf{col} + \mathsf{col}_{i_0}$;
end

If the output value of col is 0, then we have computed an expression $c + \sum_{i \in I'} \widehat{\tau}_i = 0$, where $I' \subseteq I$ is the set of indices i_0 picked in the **while** loop. Otherwise, if $\mathsf{col} \neq 0$, then $c \notin \langle \widehat{\tau}_i \rangle_{i \in I}$. The algorithm is valid because $\mathrm{low} : \{1, \cdots, m\} \to \{1, \cdots, m\}$ is injective in $\mathbf{M}$ (actually, the identity) and every column addition strictly reduces $\mathrm{low}(\mathsf{col})$. This reduction is at the heart of the standard persistence algorithm.

For any index $i \in \{1, \cdots m\}$, the boundary group $\mathbf{B}(\mathbf{K}_i)$ is generated by the cycles $\widehat{\tau}_g$, for $g \in G$ paired with an index $h \in H$ such that $g < h \leq i$. The cycle group $\mathbf{Z}(\mathbf{K}_i)$ is generated by all the cycles $\widehat{\tau}_j$ for $j \in F \sqcup G$ and $j \leq i$. In particular, for any chain $c \in \mathbf{C}(\mathbf{K}_m)$, we can express c in the basis $\{\widehat{\tau}_g\}_{g \in G}$ of $\mathbf{B}(\mathbf{K}_m)$ (or prove that it does not belong to $\mathbf{B}(\mathbf{K}_m)$) by representing c as a column col and running `Reduction(col,`G`)`. Similarly for expressing c in $\mathbf{Z}(\mathbf{K}_m)$ by running `Reduction(col,`$F \sqcup G$`)`.

Overview of the Algorithm. Given an input zigzag filtration $\mathbb{K}$, we want to compute the intervals in a direct sum decomposition of the induced zigzag module $\mathbf{H}(\mathbb{K})$ at the homology level. By convention, we denote the complexes in $\mathbb{K}$ by $\mathbf{K}'_j$ for $1 \leq j \leq n$, so as $\mathbb{K}$ is written:

(3.9)

$$\mathbf{K}'_1 \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}'_i \overset{\sigma}{\longleftrightarrow} \mathbf{K}'_{i+1} \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}'_n$$

For two indices b and d, $b \leq d$, we denote by $\mathbb{K}[b, d]$ the restriction of $\mathbb{K}$ to the simplicial complexes contained between complexes $\mathbf{K}'_b$ and $\mathbf{K}'_d$ (included). For a fixed index $i \in \{1, \ldots, n\}$, denote by m the number of simplices of $\mathbf{K}'_i$ and define $\mathbb{K}_i$ to be the following zigzag filtration:

(3.10)

$$\begin{array}{l} \mathbf{K}'_1 \longleftrightarrow \cdots \longleftrightarrow \mathbf{K}'_i = \mathbf{K}_m \xleftarrow{\tau_m} \\ \mathbf{K}_{m-1} \xleftarrow{\tau_{m-1}} \mathbf{K}_{m-2} \xleftarrow{\tau_{m-2}} \cdots \xleftarrow{\tau_1} \emptyset \end{array}$$

where the prefix made of the restriction of $\mathbb{K}_i$ to its i leftmost vector spaces is equal to $\mathbb{K}[1; i]$, and the suffix of $\mathbb{K}_i$, denoted by $\mathbb{K}_i[m; 0]$, made of the $m+1$ rightmost vector spaces contains only backward arrows. The simplices $\tau_1, \cdots, \tau_m$ of $\mathbf{K}'_i$ are removed in an arbitrary order[5] in the suffix $\mathbb{K}_i[m; 0]$. Note that we refer to the set of i leftmost indices of the zigzag filtration $\mathbb{K}_i$ using the interval notation $[1; i]$, and we refer to the set of its $m+1$ rightmost indices using the interval notation $[m; 0]$. The algorithm is iterative and maintains, at the end of iteration i, a set of chains of $\mathbf{K}'_i$ that encodes an interval decomposition of the standard persistence module $\mathbb{K}_i[m; 0]$ and is compatible with the whole zigzag module $\mathbb{K}_i$. We call this set a *compatible basis of* $\mathbb{K}_i$ and detail its definition later. During iteration $i+1$, there are two cases:

[3] Since the persistence module is represented backwards, birth and death times are reversed compared to the standard persistence setting. For instance, in this example, m is the birth time and f is the death time.

[4] This matrix can be viewed as a compact encoding of the matrices R and V in the $R = DV$ decomposition of the boundary matrix D (see [15]).

[5] In the algorithm described below, the simplices are removed in the reverse order of their insertion.

Algorithm 3: Zigzag Persistence Algorithm

$\mathbf{M} \leftarrow \emptyset$;

foreach *arrow* $\bullet_i \xleftrightarrow{\sigma} \bullet_{i+1}$ **do**

- **if** *the arrow is forward* ($\xrightarrow{\sigma}$) **then**
  - compute $\partial\sigma = \sum_{g \in G'} \widehat{\tau}_g + \sum_{f \in F'} \widehat{\tau}_f$ via `Reduction`($\mathsf{col}_{\partial\sigma}$,$F \sqcup G$);
  - **if** $F' = \emptyset$ **then** `injective_diamond`($\mathbf{M}$, G');
  - **else** `surjective_diamond`($\mathbf{M}$, F', G');
- **end**
- **if** *the arrow is backward* ($\xleftarrow{\sigma}$) **then**
  - let $\xleftarrow{(\tau_{i_0})}$ be the backward arrow of $\mathbb{K}_i[m;0]$ such that $\tau_{i_0} = \sigma$;
  - **for** $j = i_0 + 1 \cdots m$ **do** `transposition_diamond`($\mathbf{M}$, $j-1$, j);
  - restrict the encoding to $\mathbb{K}_{i+1}[m-1;0]$;
- **end**

end

---

1. $\mathbf{K}'_i \xrightarrow{\sigma} \mathbf{K}'_{i+1}$ is forward: in this case, we turn the set of chains of $\mathbf{K}_i$ forming a compatible basis of $\mathbb{K}_i$, in the following bottom module, into a set of chains of $\mathbf{K}'_{i+1}$ forming a compatible basis of $\mathbb{K}_{i+1}$, in the following top module:

(3.11)
$$\begin{tikzcd} & & \mathbf{K}'_{i+1} & & & \\ \mathbf{K}'_1 \cdots \arrow[r, leftrightarrow] & \mathbf{K}'_i \arrow[ur, "\sigma"] & & \mathbf{K}'_i \arrow[ul, "\sigma"'] & \mathbf{K}_{m-1} \arrow[l, "\tau_m"'] & \cdots \emptyset \arrow[l, "\tau_{m-1}"'] \\ & & \mathbf{K}'_i \arrow[ul, "\mathbb{1}"] \arrow[ur, "\mathbb{1}"'] & & & \end{tikzcd}$$

Note that the bottom module is simply the module $\mathbb{K}_i$ with two extra identity arrows added for convenience. In particular, it does not change the interval decomposition, up to a shift in the indices. The top module is a valid form for $\mathbb{K}_{i+1}$. The transformation between the two modules is an *arrow reflection*. At the homology level, the zigzag modules corresponding to the bottom and top zigzag filtration are related by an injective diamond if the map $\mathbf{H}(\mathbf{K}'_i) \to \mathbf{H}(\mathbf{K}'_i \cup \{\sigma\})$ induced by the insertion of σ is injective, and are related by a surjective diamond if the map is surjective.

2. $\mathbf{K}'_i \xleftarrow{\sigma} \mathbf{K}'_{i+1}$ is backward: in this case, there exists an index $i_0 \in [m;1]$ such that $\sigma = \tau_{i_0}$. For every j from i_0+1 to m, we transpose the consecutive arrows $\xleftarrow{(\tau_j)}$ and $\xleftarrow{(\tau_{i_0}=\sigma)}$, so as to get in the end:

$$\begin{tikzcd} \mathbf{K}'_1 \cdots \mathbf{K}'_i & \mathbf{K}'_{i+1} \arrow[l, "\tau_{i_0}=\sigma"'] & \mathbf{K}'_{i+1} \setminus \{\tau_m\} \arrow[l, "\tau_m"'] & \cdots \arrow[l, "\tau_{m-1}"'] \\ \cdots & \mathbf{K}'_{i+1} \setminus \{\tau_m, \ldots, \tau_{i_0+1}\} \arrow[l, "\tau_{i_0+1}"] & \cdots \emptyset \arrow[l, "\tau_{i_0-1}"] & \end{tikzcd}$$

Under each arrow transposition, we update the compatible basis of $\mathbb{K}_i$. We finally restrict the basis to the simplices of $\mathbf{K}'_{i+1}$, that is contained in $\mathbf{K}'_i$. Note that an arrow transposition consists in going from the bottom to the top module in the following diagram:

$$\begin{tikzcd} \mathbf{K}'_1 \cdots \mathbf{K}'_i & \cdots \arrow[l, "\tau_m"'] & & & \\ & & \mathbf{K} \cup \{\tau_j\} \arrow[dl, "\sigma"'] \arrow[ull, "\tau_{j+1}"'] & & \\ & \mathbf{K} \cup \{\tau_j, \sigma\} & & \mathbf{K} \arrow[ul, "\tau_j"'] \arrow[dl, "\sigma"] & \cdots \emptyset \arrow[l, "\tau_{j-1}"] \\ & & \mathbf{K} \cup \{\sigma\} \arrow[ul, "\tau_j"] & & \end{tikzcd}$$

At the homology level, the zigzag modules corresponding to the bottom and top zigzag filtrations are related by a transposition diamond by virtue of the Mayer-Vietoris theorem [21] (see also [7] for the case of the exact diamond).

Compatible Basis for a Zigzag Modules. Given $\mathbb{K}_i$ as in (3.10), we suppose that we have $\widehat{\tau}_m, \ldots, \widehat{\tau}_1 \in \mathbf{C}(\mathbf{K}_m)$, a partition $\{1, \ldots, m\} = F \sqcup G \sqcup H$, and a bijective pairing $\mathcal{P} \subseteq G \times H$ that give an encoding of the module $\mathbb{K}_i[m;0]$, which is a standard persistence module. We say that this encoding is *compatible* with the whole zigzag module $\mathbb{K}_i$ iff there exists a direct sum decomposition $\mathbb{K}_i = \bigoplus_j \mathbb{U}_j$ into interval summands $\mathbb{U}_j \cong \mathbb{I}[b_j; d_j]$, together with bijective matching between:

- F and the interval summands $\mathbb{U}_j$ with $b_j \in [1;i]$ and $d_j \in [m;0]$,
- G and the interval summands $\mathbb{U}_j \cong \mathbb{I}[b_j; d_j]$ with $b_j, d_j \in [m-1;0]$,

such that the submodules $\mathbb{U}_j$ are represented by sequences $\mathbb{U}_j = (\#, \ldots, \#, [\widehat{\tau}_f]_m, \ldots, [\widehat{\tau}_f]_f, 0, \ldots)$ for f in F, and $\mathbb{U}_j = (\ldots, [0]_m, \ldots, [0]_h, [\widehat{\tau}_g]_{h-1}, \ldots, [\widehat{\tau}_g]_g, 0, \ldots)$ for $g \in G$. The symbol $\#$ indicates that the vector

Algorithm 4: `injective_diamond`($\mathbf{M}$, G')

let H' be the set of indices $h \in H$ paired with some $g \in G'$ in $\mathcal{P}$;
$\widehat{\tau}_{m+1} \leftarrow \sigma + \sum_{h \in H'} \widehat{\tau}_h$ and let col_{m+1} represent $\widehat{\tau}_{m+1}$; set $F \leftarrow F \cup \{m+1\}$;
add column col_{m+1} to $\mathbf{M}$ and row_{m+1} that is 0 everywhere except at index $m+1$;

Algorithm 5: `surjective_diamond`($\mathbf{M}$, F', G')

set $\mathsf{col}_{f_p} \leftarrow \mathsf{col}_{f_1} + \cdots + \mathsf{col}_{f_p}$; set $b_{\ell_p} \leftarrow \max_{\leq_{\mathbf{b}}}\{b_1, \cdots, b_p\}$;
for *j from* 1 *to* $p-1$ **do**
  set $b \leftarrow b_{f_j}$;
  while *there exists* $j_0 < j$ *or* $j_0 = p$ *such that* $b_{\ell_{j_0}} = b$ **do**
    $\mathsf{col}_{f_j} \leftarrow \mathsf{col}_{f_j} + \mathsf{col}_{f_{j_0}}$; $b \leftarrow \mathrm{pred}_{\mathbf{b}}(b)$;
  end
  set $b_{\ell_j} \leftarrow b$;
end
$F \leftarrow F \setminus \{f_p\}$; $G \leftarrow G \cup \{f_p\}$;
let H' be the set of indices $h \in H$ paired with some $g \in G'$ in $\mathcal{P}$;
$\widehat{\tau}_{m+1} \leftarrow \sigma + \sum_{h \in H'} \widehat{\tau}_h$ and let col_{m+1} represent $\widehat{\tau}_{m+1}$;
add column col_{m+1} to $\mathbf{M}$ and row_{m+1} that is 0 everywhere except at index $m+1$;
$H \leftarrow H \cup \{m+1\}$; and pair f_p and $m+1$ in $\mathcal{P}$

space element at this position in the sequence can be arbitrary. Additionally, we maintain the *birth* $b[\mathsf{col}_j]$ of each column col_j which is equal to the birth of the interval submodule associated to $\widehat{\tau}_j$ in the bijection mentioned above.

Algorithm. The algorithm is purely online: we assume the input to be a stream of couples $(\sigma_i, \partial\sigma_i)_{i \geq 1}$ of a simplex σ_i and its boundary $\partial\sigma_i$, together with a flag specifying the direction of the arrow. At the beginning of iteration $i+1$ of the algorithm, we suppose we have a compatible basis of $\mathbb{K}_i$. The algorithm is iterative and maintains, at the beginning of step i, a matrix $\mathbf{M}$ that represents an encoding of $\mathbb{K}_i[m;0]$ that is compatible with $\mathbb{K}_i$ as defined previously. The procedure is described in algorithm 3.

Note that the intervals are computed when *restricting* the set of chains of the compatible basis at the end of the processing of a backward arrow. Indeed, after the sequence of calls to `transposition_diamond`, the matrix $\mathbf{M}$ maintains a compatible basis for $\mathbb{K}_i$ after transposing the arrows. The chains of the compatible basis are defined on $\mathbf{K}'_i$, and we restrict them to $\mathbf{K}'_{i+1} = \mathbf{K}'_i \setminus \{\sigma\}$. If there exists a chain $\widehat{\tau}_m$ with $m \in F$ in the encoding, and let b_m be its birth, we record an interval $\mathbb{I}[b_m; i]$ in the decomposition of the zigzag persistence module $\mathbf{H}(\mathbb{K})$. Algorithmically, the restriction consists in removing the rightmost column and the bottom row of matrix $\mathbf{M}$.

Complexity. Denote by n the total number of arrows in the quiver and by m the maximal number of simplices of a simplicial complex in the zigzag filtration. The matrix $\mathbf{M}$ contains at most m columns and m rows. The subroutine `Reduction` proceeds to at most $O(m)$ column additions and hence $O(m^2)$ operations. We prove in section 4 that the cost of `surjective_diamond` is $O(m^2)$ operations and the cost of `injective_diamond` is $O(1)$. We also prove that the cost of `transposition_diamond` is $O(m)$, and this subroutine is called $O(m)$ times during an iteration of the algorithm. Finally, the time complexity of the algorithm to compute zigzag persistent homology is $O(nm^2)$, and its memory complexity is $O(m^2)$.

4 Arrow Reflections and Transpositions.

Recall that, for ease of exposition, we assume the field of coefficients to be $\mathbb{Z}_2$. Our proofs are however written for an arbitrary field $\mathbb{F}$.

4.1 Arrow Reflection. As said before, computing an arrow reflection consists in traversing a diamond from bottom to top. We distinguish between the case where the linear map, induced at homology level by the simplex insertion $\xrightarrow{\sigma}$, is injective (of corank 1), and the case where the map is surjective (of nullity 1).

Injective Diamond. If $\partial\sigma = \sum_{g \in G'} \widehat{\tau}_g$ is a sum of boundaries, its insertion creates a new cycle class at the homology level, represented by $\widehat{\tau}_{m+1}$ and constructed in algorithm 4. The induced map at the homology level is injective.

It is easy to verify that, after the update, $\mathbf{M}$

Algorithm 6: $\texttt{transposition\_diamond}(\mathbf{M},\, i,\, i+1)$

$\mathbf{z} \leftarrow \mathbf{M}[i+1][i]$; transpose row_i and row_{i+1};
if $\mathbf{z} = 0$ **then** transpose col_i and col_{i+1}; exchange i and $i+1$ in F, G, H and $\mathcal{P}$;
else
  $\mathbf{M}[i] \leftarrow \mathsf{col}_i + \mathsf{col}_{i+1}$;
  switch $\widehat{\tau}_i, \widehat{\tau}_{i+1}$ **do**
    case *(both cycles)* $i_0 \leftarrow \mathrm{argmin}_{\leq_{\mathbf{d}}}\{b_i, b_{i+1}\}$; $\mathbf{M}[i+1] \leftarrow \mathsf{col}_{i_0}$;
    case *(exactly one cycle)* let $\widehat{\tau}_{i_0}$, $i_0 \in \{i, i+1\}$, be the cycle; $\mathbf{M}[i+1] \leftarrow \mathsf{col}_{i_0}$;
    case *(no cycle)* $i, i+1 \in H$ are paired with $g_i, g_{i+1} \in G$;
      let $i_0, i_1 = i, i+1$ such that $g_{i_0} < g_{i_1}$;
      $\mathbf{M}[g_{i_1}] \leftarrow \mathsf{col}_{g_i} + \mathsf{col}_{g_{i+1}}$; $\mathbf{M}[i+1] \leftarrow \mathsf{col}_{i_0}$;
      if $i_0 \neq i+1$ **then** exchange i and $i+1$ in F, G, H and $\mathcal{P}$;
  endsw
end

represents an encoding of $\mathbb{K}_{i+1}[m+1;0]$. The sequence with one non zero element equal to $[\widehat{\tau}_{m+1}]$ at index $i+1$ is a representative sequence for an interval submodule of $\mathbf{H}(\mathbb{K}_{i+1})$ isomorphic to $\mathbb{I}[i+1;i+1]$. It is a summand because the element $[\widehat{\tau}_{m+1}]$ is linearly independent of the elements $\{[\widehat{\tau}_f]\}_{f\in F}$ in $\mathbf{H}(\mathbf{K}_{i+1})$. Finally, the classes $[\widehat{\tau}_f]_{f\in F}$ remain independent in $\mathbf{H}(\mathbf{K}_{i+1})$ because the map induced at the homology level by the insertion of σ is injective. Consequently, we conclude, using the injective diamond principle 2.2, that the encoding represented by $\mathbf{M}$ is compatible with $\mathbb{K}_{i+1}$.

Surjective Diamond. If $\partial\sigma = \sum_{f\in F'} \widehat{\tau}_f + \sum_{g\in G'} \widehat{\tau}_g$, with $F' \subseteq F$ and $G' \subseteq G$, the kernel of the morphism induced at the homology level by the insertion of σ is spanned by $[\partial\sigma] = \sum_{f\in F'}[\widehat{\tau}_f]$, and this morphism is surjective. Write $F' = \{f_1, \ldots, f_p\}$, where the f_j are ordered such that $d_{f_1} \leq_{\mathbf{d}} \ldots \leq_{\mathbf{d}} d_{f_p}$. We define, for the set of births $\{b_{f_1}, \ldots, b_{f_p}\}$, the function $\mathrm{pred}_{\mathbf{b}}(b)$ that returns the predecessor of the index b w.r.t the order $\leq_{\mathbf{b}}$ among the set $\{b_{f_1}, \ldots, b_{f_p}\}$. The procedure described in algorithm 5 gives the change of compatible encoding underlying the new pairing presented in the surjective diamond principle 2.3. Note that the births b_{ℓ_j} are the ones described in the surjective diamond principle 2.3. For details on the correction of the algorithm, we refer to section 7. In particular, the column operations done in algorithm 5 are exactly the column operations done when reducing matrix $\mathbf{X}$ in algorithm 7 (when matching col_{f_p} with $\mathbf{x}_0$ and, for $j \geq 0$, col_{f_j} with $\mathbf{x}_j$).

In conclusion, we observe that algorithm 5 performs only left to right column additions. Hence, $\mathbf{M}$ is upper-triangular and its is easy to verify that the output matrix $\mathbf{M}$ stores an encoding of $\mathbf{K}_{i+1}[m+1;0]$. It is compatible with the zigzag $\mathbb{K}_{i+1}$ by virtue of the surjective diamond principle. We prove in lemma 7.3 that this algorithm proceeds to a linear number of column operations and hence has complexity $O(m^2)$.

4.2 Arrow Transposition.

Algorithmically, transposing the consecutive arrows $\xleftarrow{(\tau_{i+1})} \bullet \xleftarrow{(\tau_i)}$ consists in transposing rows $i+1$ and i in the matrix $\mathbf{M}$, corresponding to simplices τ_{i+1} and τ_i (see figure 2).

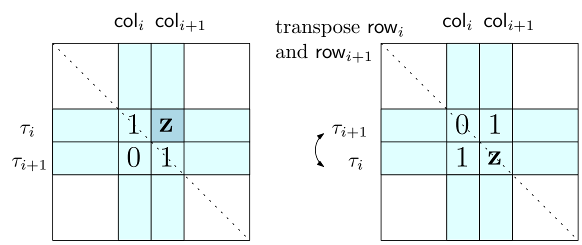

Figure 2: Row transposition

We must maintain the property that $\mathbf{M}$ represents an encoding of the suffix $\mathbb{K}_i[m;0]$ of the zigzag filtration after arrow transposition. Moreover, the new encoding must agree with the interval decomposition described by the transposition diamond principle 2.4. After transposition, $\mathbf{M}$ is not upper triangular anymore. If $\mathbf{z} = 0$, we fall into one of cases $1.(i)$, $2.(i)$, (3) or (4) of the transposition diamond principle. Hence, for $\mathbf{z} = 0$, we transpose the columns col_i and col_{i+1} and update the pairing $\mathcal{P}$ in consequence (the chains represented by col_i and col_{i+1} are paired with the same chains, but are now represented by the $(i+1)^{\text{st}}$ and i^{th} columns of $\mathbf{M}$ respectively). To avoid confusion with the notation col_j that may not correspond to the j^{th} column of $\mathbf{M}$ after transposition, we denote the j^{th} column of $\mathbf{M}$ by $\mathbf{M}[j]$.

If $\mathbf{z} \neq 0$, we fall into one of the cases $1.(ii)$, $2.(ii)$, (3) or (4) of the transposition diamond principle 2.4. Hence,

Data	$\lvert V\rvert$	d	η	ρ	$\dim \mathbf{K}$	$\max \lvert\mathbf{K}\rvert$	nb. arrows	T_{RT}	$T_{\mathtt{Dio}}$
Cli	2000	4	3	3.2	3	66172	7408112	99 sec.	272 sec.
Bro	595	25	3	3.2	3	1219926	277610862	2498 sec.	65331 sec.

Figure 3: Timings for the zigzag persistence algorithms.

we set $\mathbf{M}[i]$ to be the sum $\mathsf{col}_i + \mathsf{col}_{i+1}$, and $\mathbf{M}[i+1]$ to be either col_i or col_{i+1} depending on birth and death indices. We distinguish between the cases where both col_i and col_{i+1} represent cycles (i.e. $i, i+1 \in F \sqcup G$), where only one represents a cycle (i.e. $i \in F \sqcup G$ and $i+1 \in H$ or the other way around), and where none represents a cycle ($i, i+1 \in H$). We present the procedure in algorithm 6.

It is easy to verify that these updates turn $\mathbf{M}$ into an encoding of $\mathbb{K}_i[m;0]$ after arrow transposition. Using the transposition diamond principle 2.4, this encoding is compatible with the whole zigzag. In particular, the sum of cycles corresponds to the "sum" of representative sequence computed in the proof of the transposition diamond principle (section 6). The algorithm `transposition_diamond` proceeds to a constant number of column additions in $\mathbf{M}$, and hence has complexity $O(m)$.

4.3 Experiments. As a proof of concept, we have implemented our zigzag persistence algorithm in `C++`. The implementation relies on a sparse matrix representation, as in the standard persistence algorithm [17]. In figure 3, we compare the performance of our implementation with the one of the `Dionysus` library [19]. The **Cli** data set is a set of points from the *Clifford dataset* described in [22], which admits as underlying spaces a topological circle at small scales, a torus at larger scales and a 3-sphere at even larger scales. The **Bro** data set contains 5×5 *high-contrast patches* derived from natural images, interpreted as vectors in $\mathbb{R}^{25}$, from the Brown database [6].

We construct *oscillating Rips zigzags* on these sets of points. Details on the constructions are listed in figure 3, like the number of points $|V|$, the ambient dimension d, the parameter η and ρ for the oscillating Rips, the maximal dimension of the complex $\dim \mathbf{K}$, the maximal size of a complex in the filtration $\max |\mathbf{K}|$, the total number of arrows of the zigzag filtration "nb. arrows". The timings for the zigzag persistence algorithm with reflections and transpositions is denoted by T_{RT} and the timings for the algorithm, based on the right-filtration [7, 8], implemented in `Dionysus` is denoted by $T_{\mathtt{Dio}}$.

The speed-up is encouraging. In particular, the performance of the two algorithms is comparable on the **Cli** data set, but the algorithm introduced in this article scales much better to the data set **Bro** compared to `Dionysus`. Note that in `Dionysus`, the processing of backward arrows is much slower than the processing of forward arrows in the case of **Bro**. This is partly due to implementation issues. However, counting the timings of `Dionysus` as twice the running time for forward arrows, our algorithm remains faster.

5 Discussion.

Optimizations. Since our algorithm works similarly to the standard persistence algorithm, it is potentially amenable to the same kind optimizations. We are currently working on an adaptation for cohomology, which down the road would permit to use the optimized data structures developed in the recent years [3, 4, 16].

Generalized Diamonds. Our diamond principles hold in a fairly specific setting, but the techniques developed in our proofs may be used in a larger context, the goal being to be able to handle diamonds with arbitrary maps. We suspect that the complexity of the greedy rule to increase very rapidly with the corank and nullity of the maps though.

6 Proof of the Transposition Diamond Principle.

Let $\mathbb{I}[b;d]$ be an interval summand of $\mathbb{V}$. If $b > i+1$ or $d < i-1$, the restriction principle 1.2 implies that the interval is matched with an interval summand $\mathbb{I}[b;d]$ of $\mathbb{W}$.

If $b < i$ and $d > i$, we prove that the interval is matched with an interval summand $\mathbb{I}[b;d]$ of $\mathbb{W}$. Note that if $(\ldots u, a(u), c \circ a(u) \ldots)$ is a representative sequence for the submodule of $\mathbb{V}$ isomorphic to the summand $\mathbb{I}[b;d]$, we use $(\ldots u, b(u), d \circ b(u) \ldots)$ as representative sequence for the submodule of $\mathbb{W}$, where the "$\ldots$" are the same prefixes and suffixes in the two sequences. The new sequence is a representative sequence because, by commutativity, $d \circ b(u) = c \circ a(u)$ and $b(u) \neq 0$. We prove that the values at index i of the representative sequences for $\mathbb{W}$ remain independent. Let $\{\ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)}\ \}_{j \in J}$ be representative sequences for the interval summands of $\mathbb{V}$ satisfying $i \in (b_j; d_j)$. In particular, for every $j \in J$, $u_j^{(i)} = a(u_j^{(i-1)})$ and $u_j^{(i+1)} = c \circ a(u_j^{(i-1)})$. Suppose there exists a family of scalars $\{\alpha_j\}_{j \in J}$, not all zero, such that $\sum_{j\in J}\alpha_j b(u_j^{(i-1)})=0$. By commutativity, this implies that $\sum_{j\in J}\alpha_j d\circ b(u_j^{(i-1)})=\sum_{j\in J}\alpha_j c\circ a(u_j^{(i-1)})=\sum_{j\in J}\alpha_j u_j^{(i+1)}=0$, which is in contradiction with the fact that the family $\{u_j^{(i+1)}\}_{j\in J}$ is free.

We study now the evolution of interval summands $\mathbb{I}[b;d]$, with $b\in\{i,i+1\}$ and $d\in\{i-1,i\}$, when passing a transposition diamond. For all cases of the theorem, we exhibit representative sequences that, together with the ones defined above, represent an interval decomposition for $\mathbb{W}$.

Case 1. Recall that, by commutativity of the diamond, b and d must be both surjective of nullity 1. We first describe $\ker b$. As $c\circ a(v)=0$, the element $(a(v),0)$ belongs to $\ker D_2$ and hence belongs to im D_1 by exactness. The preimage $a^{-1}(\{a(v)\})$ is $v+\langle u\rangle$, thus there exists $\alpha\in\mathbb{F}$ such that $\ker b=\langle v+\alpha u\rangle$ (b has nullity 1). Note that this implies that $b(u)\neq 0$.

(i) If $\alpha=0$, we consider the representative sequences $(\ldots u,b(u),0\ldots)$ and $(\ldots v,0,0\ldots)$.

(ii) If $\alpha\neq 0$, we consider the representative sequences $(\ldots v,0,0\ldots)*\alpha\cdot(\ldots u,b(u),0\ldots)=(\ldots v+\alpha u,0,0\ldots)$ (with birth $\max_{\leq_{\mathbf{b}}}\{b,b'\}$) and $(\ldots w,b(w),0\ldots)$ (with birth $\min_{\leq_{\mathbf{b}}}\{b,b'\}$), where $w\in V_{i-1}$ is taken to be u if $b\leq_{\mathbf{b}} b'$ and v otherwise. Note that the $*$ is well-defined.

Let $\{b(u_j^{(i-1)})\}_{j\in J}$ be the family of i^{th} elements of all representative sequences of $\mathbb{W}$ traversing index i (as defined above). In case (i), $\{b(u_j^{(i-1)})\}_{j\in J}\cup\{b(u)\}$ is a basis for W_i. Indeed, the family is free. Suppose otherwise: there exist scalars α,α_j such that $\alpha b(u)+\sum_{j\in J}\alpha_j b(u_j^{(i-1)})=0$, i.e. $\alpha u+\sum_{j\in J}\alpha_j u_j^{(i-1)}=\beta v$, as $\ker b=\langle v\rangle$. This a contradiction with the fact that $\{u,v\}\cup\{u_j^{(i-1)}\}_{j\in J}$ is free. Additionally, $\dim W_i=1+|J|$.

In case (ii), we prove similarly that $\{b(w)\}\cup\{b(u_j^{(i-1)})\}_{j\in J}$ is a basis for W_i. We conclude using proposition 1.2.

Case 2. Recall that, by commutativity of the diamond, b and d must be both injective of nullity 1. Let $(\ldots 0,v,c(v),\ldots)$ and $(\ldots 0,0,u,\ldots)$, $v\in V_i$ and $u\in V_{i+1}$, be representative sequences for the interval submodules of $\mathbb{V}$ isomorphic to $\mathbb{I}[i;d]$ and $\mathbb{I}[i+1;d']$ respectively. First, we prove that $c(v)\notin$ im d. Suppose otherwise: there exists $w\in W_i$ such that $d(w)=c(v)$. But then (v,w) belongs to $\ker D_2$ and hence to im D_1 by exactness. This is in contradiction with the fact $v\notin$ im a.

Because d has corank 1, there exists $\alpha\in\mathbb{F}$ such that $u+\alpha c(v)\in$ im d. Note that for any α, $u+\alpha c(v)\notin$ im $d\circ b$. Suppose otherwise: $u+\alpha c(v)$ belongs to im $c\circ a$ which implies $u+\alpha c(v)\in$ im c and finally $u\in$ im c, a contradiction. We distinguish the two cases:

(i) If $\alpha=0$ then there exists $w\in W_i\setminus$im b such that $d(w)=u$. We consider the representative sequences $(\ldots 0,w,u\ldots)$ and $(\ldots 0,0,c(v)\ldots)$.

(ii) If $\alpha\neq 0$ then $u\notin$ im d and there exists $w\in W_i\setminus$ im b such that $d(w)=u+\alpha c(v)$. We consider the representative sequences $(\ldots 0,w,u+\alpha c(v)\ldots)$ (with death $\max_{\leq_{\mathbf{d}}}\{d,d'\}$) and $(\ldots 0,0,x,\ldots)$ (with death $\min_{\leq_{\mathbf{d}}}\{d,d'\}$), where $x\in V_{i+1}$ is taken to be u if $d'\leq_{\mathbf{d}} d$ and $c(v)$ otherwise.

In both cases (i) and (ii) we can verify that the family containing w and $\{b(u_j^{(i-1)})\}_{j\in J}$ is free for otherwise the family $\{u,v\}\cup\{d\circ b(u_j^{(i-1)})=u_j^{(i+1)}\}_{j\in J}$ would not be free in V_{i+1}. We conclude using proposition 1.2.

Case 3. We prove that d is injective (of corank 1). Suppose otherwise: there exists $w\neq 0$ in W_i such that $d(w)=0$. Hence, $(0,w)$ belongs to $\ker D_2$ and belongs to im D_1 by exactness. This implies the existence of $x\neq 0$ in V_{i-1} such that $a(x)=0$, a contradiction with a injective. To preserve dimensions, b is thus surjective of nullity 1.

We now prove that there is no summand $\mathbb{I}[i;i]$ in $\mathbb{V}$. Suppose otherwise: there exists $x\in V_i\setminus$ im a and $c(x)$. Hence, $(x,0)\in\ker D_2=$ im D_1 while x does not belong to im a, a contradiction.

Consequently, let $(0\ldots 0,v,c(v)\ldots)$ and $(\ldots,u,a(u),0\ldots)$, $u\in V_{i-1}$ and $v\in V_i$, be representative sequences for the interval submodules of $\mathbb{V}$ isomorphic to $\mathbb{I}[i;d]$ and $\mathbb{I}[b;i]$ respectively. We prove that $c(v)\notin$ im d. Suppose otherwise: there exists $w\in W_i$ such that $d(w)=c(v)$. We use, as previously, the exactness to show that this would imply that v has a preimage through a.

Finally, we prove that $b(u)=0$. Indeed, the element $(0,b(u))$ belongs to $\ker D_2$ because $c\circ a(u)=d\circ b(u)=0$ and hence belongs to im D_1 by exactness. Because a is injective, $a^{-1}(0)=0$, which implies that $b(u)$ must be 0.

In conclusion, we consider the representative sequences $(\ldots 0,0,c(v)\ldots)$ and $(\ldots,u,0,0\ldots)$ for interval submodules of $\mathbb{W}$ that are isomorphic to $\mathbb{I}[i+1;d]$ and $\mathbb{I}[b;i-1]$ respectively. We conclude using proposition 1.2.

Case 4. Case 4. is deduced by symmetry from case 3.

□

7 Proof of the Surjective Diamond Principle.

Consider the diagram:

(7.12)

$$\begin{tikzcd}[column sep=small]
\mathbb{W} := & & & & W & & & & \\
& V_1 \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V \arrow[ur, "f"] \arrow[dr, "\mathbb{1}"'] & & V \arrow[ul, "f"'] \arrow[dl, "\mathbb{1}"] \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_n \\
\mathbb{V} := & & & & V & & & &
\end{tikzcd}$$

where the diamond is located at index i in the module and where f is surjective of nullity 1. Note that the arrows at index i in $\mathbb{V}$ are reverted compared to diagram 2.7. Because the morphisms on these arrows are the identity, it does not change the decomposition of $\mathbb{V}$. Suppose $\mathbb{V}$ is isomorphic to:

$$\mathbb{V} \cong \bigoplus_{j \in J} \mathbb{I}[b_j; d_j].$$

All along the proof, we fix a family of representative sequences $\{\ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)}\ \}_{j \in J}$ that represents the interval decomposition of $\mathbb{V}$. From these representative sequences, we construct explicit representative sequences for the interval submodules in the direct sum decomposition of $\mathbb{W}$. The construction is ad hoc as it depends on the chosen representative sequences of $\mathbb{V}$. However, it provides an interval decomposition of $\mathbb{W}$ that is canonical due to theorem 1.1.

Contributing Intervals. Let $\xi \neq 0$ be a vector in $\ker f$. Up to a reordering of the indices in J, write ξ as:

$$\xi = \alpha_1 u_1^{(i)} + \cdots + \alpha_p u_p^{(i)},$$

with $\alpha_j \neq 0$ for every $1 \leq j \leq p$ and $d_1 \leq_{\mathbf{d}} \cdots \leq_{\mathbf{d}} d_p$. We say that such interval $\mathbb{I}[b_j; d_j]$ of the decomposition, with $1 \leq j \leq p$, *contributes* to the kernel of f.

LEMMA 7.1. *Every interval $\mathbb{I}[b; d]$ of the decomposition of $\mathbb{V}$ that does not contribute to $\ker f$ is matched with an interval $\mathbb{I}[b; d]$ in the decomposition of $\mathbb{W}$.*

Proof. Let $\mathbb{V} \cong \bigoplus_{j=1}^{p} \mathbb{I}[b_j; d_j] \oplus \mathbb{U}$, where $\mathbb{U}$ contains all intervals not contributing to $\ker f$. Consider the morphism $\phi : \mathbb{V} \to \mathbb{W}$ defined as:

$$\begin{tikzcd}[column sep=small]
\mathbb{V}: \arrow[d, "\phi"'] & V_1 \arrow[r, leftrightarrow] \arrow[d, "\mathbb{1}"'] & \cdots \arrow[r, leftrightarrow] & V \arrow[r, "\mathbb{1}"] \arrow[d, "\mathbb{1}"'] & V \arrow[d, "f"'] & V \arrow[l, "\mathbb{1}"'] \arrow[r, leftrightarrow] \arrow[d, "\mathbb{1}"'] & \cdots \arrow[r, leftrightarrow] & V_n \arrow[d, "\mathbb{1}"'] \\
\mathbb{W}: & V_1 \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V \arrow[r, "f"] & W & V \arrow[l, "f"'] \arrow[r, leftrightarrow] & \cdots \arrow[r, leftrightarrow] & V_n
\end{tikzcd}$$

The first isomorphism theorem implies that:

$$\mathbb{V} / \ker \phi \cong \mathrm{im}\ \phi = \mathbb{W}$$

by surjectivity of ϕ. We conclude that:

$$\mathbb{W} \cong \left(\bigoplus_{j=1}^{p} \mathbb{I}[b_j; d_j] \right) / \ker \phi \oplus \mathbb{U}.$$

$\square$

Consequently, we assume in the following that all the intervals in the decomposition of $\mathbb{V}$ contain index i and contribute to $\ker f$. Specifically, the decomposition of $\mathbb{V}$ is $\mathbb{V} \cong \bigoplus_{j=1}^{p} \mathbb{I}[b_j; d_j]$ and it is represented by $\{\ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)}\ \}_{j=1,\ldots,p}$, with $b_j < i < d_j$ for all j and $p = \dim V$.

Formal Sums of Sequences. We consider the restrictions $\mathbb{V}[1; i-1]$ and $\mathbb{V}[i+1; n]$ of the representation $\mathbb{V}$. By mean of the restriction principle 1.2, their interval decompositions are:

$$\mathbb{V}[1; i-1] \cong \bigoplus_{j=1}^{p} \mathbb{I}[b_j; i-1] \text{ and } \mathbb{V}[i+1; d_j] \cong \bigoplus_{j=1}^{p} \mathbb{I}[i+1; d_j]$$

and are represented by $\{\ u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(i-1)}\ \}_{j=1,\ldots,p}$ and $\{\ u_j^{(i+1)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)}\ \}_{j=1,\ldots,p}$ respectively. For short, we denote by $\mathbf{x}_j$ the sequence $u_j^{(b_j)} \leftrightarrow \cdots \leftrightarrow u_j^{(i-1)}$ and by $\mathbf{y}_j$ the sequence $u_j^{(i+1)} \leftrightarrow \cdots \leftrightarrow u_j^{(d_j)}$. These sequences also represent interval decompositions for $\mathbb{W}[1; i-1]$ and $\mathbb{W}[i+1; n]$ as $\mathbb{V}$ and $\mathbb{W}$ differ only by a vector space at index i. We call a representative sequence in $\mathbb{V}[1; i-1]$ whose death is $i-1$ a *right sequence*, and we call a representative sequence in $\mathbb{V}[i+1; n]$ whose birth is $i+1$ a *left sequence*.

Let $\mathcal{U}$ be the set of all *formal sums* of $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ with coefficients in $\mathbb{F}$, i.e.

$$\mathcal{U} = \{\ \mathbf{x}\ :\ \mathbf{x} = \gamma_1 \mathbf{x}_1 + \ldots + \gamma_p \mathbf{x}_p, \quad \gamma_j \in \mathbb{F}\ \},$$

and let $\mathcal{V}$ be the set of all formal sums of $\{\mathbf{y}_1, \ldots, \mathbf{y}_p\}$ with coefficients in $\mathbb{F}$, i.e.

$$\mathcal{V} = \{\ \mathbf{y}\ :\ \mathbf{y} = \gamma_1 \mathbf{y}_1 + \ldots + \gamma_p \mathbf{y}_p, \quad \gamma_j \in \mathbb{F}\ \}.$$

Naturally, $\mathcal{U}$ and $\mathcal{V}$ are $\mathbb{F}$-vector spaces. Recall that $\leq_{\mathbf{b}}$ and $\leq_{\mathbf{d}}$ are total order relations defined on the indices $\{1, \ldots, n\}$ of the bottom quiver in diagram 7.12. Recall also that "$\cdot$" is a scalar multiplication for representative sequences and $*$ is a binary operator.

PROPOSITION 7.1. *Every formal sum $\gamma_1 \mathbf{x}_1 + \ldots + \gamma_p \mathbf{x}_p \neq 0$ of $\mathcal{U}$ defines a right sequence:*

$$\gamma_1 u_1^{(b)} + \ldots + \gamma_p u_p^{(b)} \leftrightarrow \cdots \leftrightarrow \gamma_1 u_1^{(i-1)} + \ldots + \gamma_p u_p^{(i-1)}$$

where the birth b is equal to $\max_{\leq_{\mathbf{b}}} \{b_j : \gamma_j \neq 0\}_{j=1,\ldots,p}$. Similarly, every formal sum $\gamma_1 \mathbf{y}_1 + \ldots + \gamma_p \mathbf{y}_p \neq 0$ of $\mathcal{V}$ defines a left sequence:

$$\gamma_1 v_1^{(i+1)} + \ldots + \gamma_p v_p^{(i+1)} \leftrightarrow \cdots \leftrightarrow \gamma_1 v_1^{(d)} + \ldots + \gamma_p v_p^{(d)}$$

where the death d is equal to $\max_{\leq_{\mathbf{d}}} \{d_j : \gamma_j \neq 0\}_{j=1,\ldots,p}$.

Proof. We prove the result for $\mathcal{U}$ only, the case of $\mathcal{V}$ being symmetric. Let $i_1, \ldots, i_k$ be the indices such that $\gamma_{i_j} \neq 0$. We prove the proposition by induction on j for the formal sum $\gamma_{i_1}\mathbf{x}_{i_1} + \ldots + \gamma_{i_j}\mathbf{x}_{i_j}$. For $j = 1$, $\gamma_{i_1}\mathbf{x}_{i_1}$ defines the right sequence $\gamma_{i_1} \cdot \mathbf{x}_{i_1}$, where " $\cdot$ " is the scalar multiplication of definition 1.2. Suppose the property true up to j: $\gamma_{i_1}\mathbf{x}_{i_1} + \ldots + \gamma_{i_j}\mathbf{x}_{i_j}$ defines a right sequence $\mathbf{x} =$

$$\gamma_{i_1} u_{i_1}^{(b)} + \ldots + \gamma_{i_j} u_{i_j}^{(b)} \leftrightarrow \gamma_{i_1} u_{i_1}^{(i-1)} + \ldots + \gamma_{i_j} u_{i_j}^{(i-1)}$$

We prove that $\mathbf{x}$ and $\gamma_{i_{j+1}} \cdot \mathbf{x}_{i_j}$ satisfy conditions *(a)*, *(b)* and *(c)* of definition 1.2 and show that $\mathbf{x} * (\gamma_{i_{j+1}} \cdot \mathbf{x}_{i_j})$ is the right sequence associated to $\gamma_{i_1}\mathbf{x}_{i_1} + \ldots + \gamma_{i_{j+1}}\mathbf{x}_{i_{j+1}}$. Conditions *(a)* and *(c)* are directly satisfied considering the two sequences have same death $i-1$, and $i-1$ is the largest index in the restricted module $\mathbb{V}[1; i-1]$. We prove condition *(b)* by contradiction: suppose there exists and index $k \in [b; i-1] \cap [b_{i_{j+1}}; i-1]$ such that, for some $\alpha, \beta \in \mathbb{F}$, not both 0, we have:

$$\alpha(\gamma_{i_1} u_{i_1}^{(k)} + \ldots + \gamma_{i_j} u_{i_j}^{(k)}) + \beta \gamma_{i_{j+1}} u_{i_{j+1}}^{(k)} = 0$$

By hypothesis on k, not all $u_i^{(k)}$ in the sum are 0 and hence we have a contradiction with the fact that the family $\{u_j^{(k)} : u_j^{(k)} \neq 0\}$ is free in V_k.

Consequently, $\mathbf{x} * (\gamma_{i_{j+1}} \cdot \mathbf{x}_{i_j})$ is a well-defined right sequence. Its birth is $\max_{\leq_{\mathbf{b}}} \{b_{i_\ell}\}_{\ell=1,\ldots,j+1}$ and its death is $i-1$.

Finally, using the commutativity and associativity of $*$ (lemma 1.2), we conclude that the representative sequence constructed does not depend on the order in which the sequences are added.

□

By a small abuse of notation, we use the same notation $\mathbf{x}$ for the formal sum and for the representative sequence it defines.

Recall that $\xi = \alpha_1 u_1^{(i)} + \cdots + \alpha_p u_p^{(i)}$ generates the kernel of f. We define the right and left sequences $\xi^{\mathcal{U}}$ and $\xi^{\mathcal{V}}$ to be equal to $\alpha_1\mathbf{x}_1 + \cdots + \alpha_p\mathbf{x}_p$ and $\alpha_1\mathbf{y}_1 + \cdots + \alpha_p\mathbf{y}_p$ respectively.

The Join Operator. Let $u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(i-1)}$ and $v^{(i+1)} \leftrightarrow \cdots \leftrightarrow v^{(d)}$ be respectively a right sequence in $\mathbb{V}[1; i-1]$ and a left sequence in $\mathbb{V}[i+1; n]$. Recall that f is the surjective map in the diamond. If the sequences satisfy $f(u^{(i-1)}) = f(v^{(i+1)})$ (we denote this element by w), we define the *join* of these right and left sequences by the representative sequence of $\mathbb{W}$ equal to:

$$(\ u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(i-1)}\ ) \bowtie (\ v^{(i+1)} \leftrightarrow \cdots \leftrightarrow v^{(d)}\ ) :=$$
$$u^{(b)} \leftrightarrow \cdots \leftrightarrow u^{(i-1)} \rightarrow w \leftarrow v^{(i+1)} \leftrightarrow \cdots \leftrightarrow v^{(d)}$$

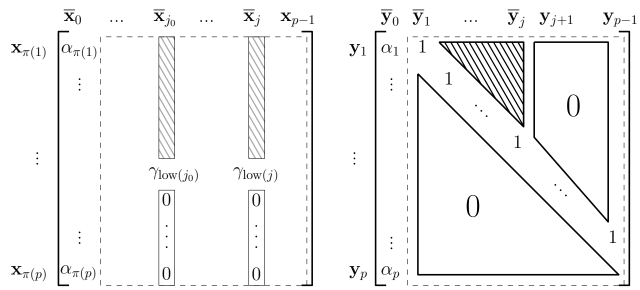

Figure 4: Matrices $\mathbf{X}$ and $\mathbf{Y}$ during the j^{th} iteration of algorithm 7.

The following lemma is a direct consequence of the linearity of f:

LEMMA 7.2. *Let $\mathbf{x}, \mathbf{x}' \in \mathcal{U}$, $\mathbf{y}, \mathbf{y}' \in \mathcal{V}$ such that $\mathbf{x} \bowtie \mathbf{y}$ and $\mathbf{x}' \bowtie \mathbf{y}'$ are well-defined. For any $\alpha, \beta \in \mathbb{F}$, the join $(\alpha\mathbf{x} + \beta\mathbf{x}') \bowtie (\alpha\mathbf{y} + \beta\mathbf{y}')$ is well-defined.*

Greedy Pairing of Births and Deaths. Up to a reordering of the indices $\{1, \ldots, p\}$, suppose that the deaths satisfy $d_1 \leq_{\mathbf{d}} \ldots \leq_{\mathbf{d}} d_p$. Define π to be the permutation of $\{1, \cdots, p\}$ satisfying $b_{\pi(1)} \leq_{\mathbf{b}} \ldots \leq_{\mathbf{b}} b_{\pi(p)}$, breaking ties arbitrarily. In particular d_p is the maximum w.r.t. $\leq_{\mathbf{d}}$ among all the death indices $\{d_1, \cdots, d_p\}$ and $b_{\pi(p)}$ is the maximum w.r.t. $\leq_{\mathbf{b}}$ among all the birth indices $\{b_1, \cdots, b_p\}$.

We construct new bases $\{\overline{\mathbf{x}}_0, \ldots, \overline{\mathbf{x}}_{p-1}\}$ for $\mathcal{U}$ and $\{\overline{\mathbf{y}}_0, \cdots, \overline{\mathbf{y}}_{p-1}\}$ for $\mathcal{V}$ such that:

1. the p right sequences $\mathbf{x}_j$, $0 \leq j \leq p-1$, have births exactly the set $\{b_1, \ldots, b_p\}$,
2. the p left sequences $\mathbf{y}_j$, $0 \leq j \leq p-1$, have deaths exactly the set $\{d_1, \ldots, d_p\}$,
3. for every j such that $1 \leq j \leq p-1$, the join between $\mathbf{x}_j$ and $\mathbf{y}_j$ is well-defined,
4. the joins $\mathbf{x}_0 \bowtie 0$ and $0 \bowtie \mathbf{y}_0$ are well-defined, where 0 refers to the sequence where all elements are 0. $\mathbf{x}_0 \bowtie 0$ has death $i-1$ and $0 \bowtie \mathbf{y}_0$ has birth $i+1$.

We pick $\overline{\mathbf{x}}_0 = \xi^{\mathcal{U}}$ and $\overline{\mathbf{y}}_0 = \xi^{\mathcal{V}}$ as first elements for the new bases. They satisfy condition (4.) above. Note that $\{\overline{\mathbf{x}}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{p-1}\}$ and $\{\overline{\mathbf{y}}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{p-1}\}$ are bases for $\mathcal{U}$ and $\mathcal{V}$ respectively. Note also that, by definition of the $\mathbf{x}_j$ and the $\mathbf{y}_j$, conditions (2.) and (3.) are satisfied, but not condition (1.) as $\xi^{\mathcal{U}}$ does not have, in general, birth b_p. The construction is algorithmic[6]

[6]Note that these algorithm are theoretical and are not used in the zigzag persistence algorithm presented in section 3. They are meant to prove the surjective diamond principle, which offers a simpler computational rule for updating the interval decomposition.

Algorithm 7: Reduction to column echelon form of $\mathbf{X}$ and $\mathbf{Y}$.

Data: Matrices $\mathbf{X}$ and $\mathbf{Y}$
for $j = 1$ *to* $p-1$ **do**
  while *there exists* $j_0 < j$ *with* $\mathrm{low}_{\mathbf{X}}(j_0) = \mathrm{low}_{\mathbf{X}}(j)$ **do**
    $\overline{\mathbf{x}}_j \leftarrow \overline{\mathbf{x}}_j + \left(\frac{-\gamma_{\mathrm{low}_{\mathbf{X}}(j)}}{\gamma_{\mathrm{low}_{\mathbf{X}}(j_0)}}\,\overline{\mathbf{x}}_{j_0}\right)$;
    if $j_0 \neq 0$ **then** $\overline{\mathbf{y}}_j \leftarrow \overline{\mathbf{y}}_j + \left(\frac{-\gamma_{\mathrm{low}_{\mathbf{X}}(j)}}{\gamma_{\mathrm{low}_{\mathbf{X}}(j_0)}}\,\overline{\mathbf{y}}_{j_0}\right)$;
  end
end

and consists of a simultaneous basis change in the vector spaces $\mathcal{U}$ and $\mathcal{V}$. Let $\mathbf{X}$ be the $(p \times p)$ matrix, with $\mathbb{F}$ coefficients, representing the basis $\{\xi^{\mathcal{U}}, \mathbf{x}_1, \cdots, \mathbf{x}_{p-1}\}$ of $\mathcal{U}$ into the basis $\{\mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(p)}\}$, and let $\mathbf{Y}$ be the $(p\times p)$ matrix, with $\mathbb{F}$ coefficients, representing the basis $\{\xi^{\mathcal{V}}, \mathbf{y}_1, \cdots, \mathbf{y}_{p-1}\}$ of $\mathcal{V}$ into the basis $\{\mathbf{y}_1, \cdots, \mathbf{y}_p\}$, as depicted in figure 4. Note that columns are labeled from 0 to $p-1$ and rows are labeled from 1 to p, in order to match with the indices of the bases elements.

For a matrix $\mathbf{M}$, we define $\mathrm{low}_{\mathbf{M}}(j)$ to be the row index of the lowest non-zero coefficient of the j^{th} column, for $0 \leq j \leq p-1$. Let $\gamma_{\mathrm{low}(j)} \in \mathbb{F}$ be the value of this lowest coefficient in matrix $\mathbf{X}$. We denote the reduced columns of $\mathbf{X}$ and $\mathbf{Y}$ by $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{y}}_i$ respectively. The algorithm is a reduction to column echelon form of the matrix $\mathbf{X}$, meaning that at the end of the algorithm, any row index j admits a unique column k such that $j = \mathrm{low}_{\mathbf{X}}(k)$. The process is presented in algorithm 7. The reduction is done by mean of elementary column operations, that are reproduced almost identically in $\mathbf{Y}$ (**if** condition). In the algorithm, for any $0 \leq j \leq p-1$, we denote by $\gamma_{\mathrm{low}_{\mathbf{X}}(j)}$ the value of the lowest non-zero coefficient of column j in matrix $\mathbf{X}$.

Because the algorithm consist in a change of basis in vector spaces $\mathcal{U}$ and $\mathcal{V}$, the algorithm is valid and, at the end of the reduction, $\mathbf{X}$ is in column echelon form. The **if** condition of the reduction ensures that the function $\mathrm{low}_{\mathbf{Y}}$ does not change and $\mathbf{Y}$ is in column echelon form as well. Because the rows are sorted by increasing birth values w.r.t. $\leq_{\mathbf{b}}$ in $\mathbf{X}$, the birth of $\overline{\mathbf{x}}_j$ is the birth $b_{\pi(k)}$ such that $k = \mathrm{low}_{\mathbf{X}}(j)$. Consequently, as the function $\mathrm{low}_{\mathbf{X}}$ is bijective ($\mathbf{X}$ is in column echelon form and is full-rank), condition (1.) is satisfied. With a similar argument on $\mathbf{Y}$, condition (2.) is satisfied. Finally, condition (3.) is satisfied because, for any $j \geq 1$ and all along the procedure, the vector at index $i-1$ in the right sequence $\overline{\mathbf{x}}_j$ and the vector at index $i+1$ in the left sequence $\overline{\mathbf{y}}_j$ are either identical, or differ by $\gamma\xi$. They consequently have the same image by f.

Proposition 7.2. *The family* $\{\overline{\mathbf{x}}_0 \bowtie 0, 0 \bowtie \overline{\mathbf{y}}_0, \overline{\mathbf{x}}_1 \bowtie \overline{\mathbf{y}}_1, \cdots, \overline{\mathbf{x}}_{p-1} \bowtie \overline{\mathbf{y}}_{p-1}\}$ *of representative sequences of* $\mathbb{W}$ *represents an interval decomposition of* $\mathbb{W}$.

Proof. For any index j, $0 \leq j \leq p-1$, denote $\mathbf{x}_j$ and $\overline{\mathbf{x}}_j$ by, respectively, $u^{(b_j)} \leftrightarrow \cdots \leftrightarrow u^{(i-1)}$ and $\overline{u}_j^{(b_{\ell_j})} \leftrightarrow \cdots \leftrightarrow \overline{u}_j^{(i-1)}$. We prove that, for any index $k \leq i-1$, the family $\{\overline{u}_j^{(k)} : \overline{u}_j^{(k)} \neq 0\}_{j=0,\ldots,p-1}$ is a basis for V_k. By virtue of proposition 7.1, the family $\{\overline{u}_j^{(k)} : \overline{u}_j^{(k)} \neq 0\}_{j=0,\ldots,p-1}$ is a subset of the family $\{\mathbf{X}u_j^{(k)} : u_j^{(k)} \neq 0\}_{j=0,\ldots,p-1}$, where $\mathbf{X}$ is in column echelon form and is invertible, and the set of non-zero $u_j^{(k)}$ is free. Consequently, for any index $k \leq i-1$, the family of non-zero $u_j^{(k)}$ is free. Because, the births of the representative sequences $\{\overline{\mathbf{x}}_0 \bowtie 0, 0 \bowtie \overline{\mathbf{y}}_0, \overline{\mathbf{x}}_1 \bowtie \overline{\mathbf{y}}_1, \cdots, \overline{\mathbf{x}}_{p-1} \bowtie \overline{\mathbf{y}}_{p-1}\}$ are exactly the set $\{b_1, \ldots, b_p\}$, the set $\{\overline{u}_j^{(k)} : \overline{u}_j^{(k)} \neq 0\}_{j=0,\ldots,p-1}$ contains $\dim V_k$ elements and is a basis of V_k. Denoting $\overline{\mathbf{y}}_j = \overline{v}_j^{(i+1)} \leftrightarrow \cdots \leftrightarrow \overline{v}_j^{(d_j)}$, we prove similarly that the family of non-zero $\overline{v}_j^{(k)}$ is a basis for V_k, for any index $k \geq i+1$.

Finally, at index $k = i$, the non zero vectors $\overline{u}_j^{(i)}$ form a basis because the decomposition is aligned with the kernel of f. We conclude using proposition 1.2. $\square$

Finally, we study a bit closer the structure of the matrix $\mathbf{X}$ during the reduction, in order to deduce the simpler pairing rule described in algorithm 1 of the surjective diamond principle 2.3.

Lemma 7.3. *The reduction algorithm 7 satisfies the following properties.*

(i) During its reduction, any column col *of* $\mathbf{X}$ *is either equal to* $[0, \ldots, 0, 1, 0, \ldots 0]^T$ *or to* $\gamma \times [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(\mathrm{low}_{\mathbf{X}}(\mathsf{col}))}, 0, \ldots, 0]^T$ *for some* $\gamma \neq 0$.

(ii) The reduction of $\mathbf{X}$ *performs* $O(p)$ *column operations in* $\mathbf{X}$.

(iii) The pairing procedure in algorithm 1 of the surjective diamond principle 2.3 is correct.

Proof. **(i)** We prove the result by induction on the column index j. For $j = 0$, $\overline{\mathbf{x}}_0 = [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(p-1)}]^T$ and the property is satisfied. Suppose all columns $\overline{\mathbf{x}}_k$, for $k \leq j$, satisfy the property. We reduce column $\mathbf{x}_{j+1}$, originally equal to $[0, \ldots, 0, 1, 0, \ldots, 0]$. If there is no column $\overline{\mathbf{x}}_k$, with $k < j+1$, such that $\mathrm{low}_{\mathbf{X}}(k) = \mathrm{low}_{\mathbf{X}}(j+1)$, then the column $\mathbf{x}_{j+1}$ is reduced and satisfies the property. Otherwise, let $\overline{\mathbf{x}}_k$ be such that $k \leq j$ and $\mathrm{low}_{\mathbf{X}}(k) = \mathrm{low}_{\mathbf{X}}(j+1)$. By induction, $\overline{\mathbf{x}}_k$ satisfies the property. Because the matrix has full-rank, $\overline{\mathbf{x}}_k$ must be equal to $\gamma \times [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(\mathrm{low}_{\mathbf{X}}(j+1))}, 0, \ldots, 0]^T$ for some $\gamma \neq 0$.

We reduce $\mathbf{x}_{j+1}$ using $\overline{\mathbf{x}}_k$ as in the **while** loop. Denote $\mathrm{low}_{\mathbf{X}}(j+1)$ by ℓ. We get $\overline{\mathbf{x}}_{j+1} = \gamma' \times [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(\ell-1)}, 0, \ldots, 0]^T$ for some $\gamma' \neq 0$. By induction, and because $\mathbf{X}$ has full-rank, the remainder of the reduction of $\overline{\mathbf{x}}_{j+1}$ involves only columns of the form $[0, \ldots, 0, 1, 0, \ldots, 0]^T$. Consequently, the reduced column $\overline{\mathbf{x}}_{j+1}$ satisfies the property.

(ii) The property follows by noticing that, during the whole reduction, a column $\mathbf{x}_k$ is picked at most once in the **while** loop condition for reducing another column (i.e. $j_0 \leftarrow k$).

(iii) For each column operation $\overline{\mathbf{x}}_j \leftarrow \overline{\mathbf{x}}_j + \gamma \overline{\mathbf{x}}_{j_0}$, the index $\mathrm{low}_{\mathbf{X}}(j)$ decreases by exactly 1. Recall that the birth of the representative sequence defined by $\overline{\mathbf{x}}_j$ is $b_{\pi(\mathrm{low}_{\mathbf{X}}(j))}$. Recall also that, by definition of the permutation π, the rows are ordered by increasing birth w.r.t. $\leq_{\mathbf{b}}$, i.e. $b_{\pi(1)} \leq_{\mathbf{b}} \ldots \leq_{\mathbf{b}} b_{\pi(p)}$. We prove by induction on j that algorithm 1 of the surjective diamond principle 2.3 is correct. More precisely, we prove that every reduced column $\overline{\mathbf{x}}_j$ equal to $[0, \ldots, 0, 1, 0, \ldots, 0]^T$ has birth $b_{\ell_j} = b_j$ and every reduced column $\overline{\mathbf{x}}_j$ equal to $\gamma \times [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(\mathrm{low}_{\mathbf{X}}(j))}, 0, \ldots, 0]^T$ has birth b_{ℓ_j} equal to the maximal birth index, w.r.t. $\leq_{\mathbf{b}}$, available during its reduction. For $j = 0$, $\overline{\mathbf{x}}_0$ has birth $b_{\pi(p)}$ which is the maximum w.r.t. $\leq_{\mathbf{b}}$. Suppose the property true for all $k \leq j$. Consider column $\overline{\mathbf{x}}_{j+1}$. If its reduced form is $[0, \ldots, 0, 1, 0, \ldots, 0]^T$, then its representative sequence has indeed birth b_{j+1}. Otherwise, there exists a column $\overline{\mathbf{x}}_k = \gamma \times [\alpha_{\pi(1)}, \ldots, \alpha_{\pi(\mathrm{low}_{\mathbf{X}}(j+1))}, 0, \ldots, 0]^T$ for $k \leq j$. Hence, by induction, none of the births larger than b_{j+1} are available. By reducing $\mathrm{low}_{\mathbf{X}}(j+1)$ by exactly 1 for each column operation, $\overline{\mathbf{x}}_{j+1}$ is assigned the largest birth w.r.t. $\leq_{\mathbf{b}}$ that is available. □

This concludes the proof of the surjective diamond principle.

Acknowledgments.

The authors wish to thank the reviewers for their insightful comments. They acknowledge the support of ANR through the TopData project, and of ERC through the Gudhi project.

References

[1] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Proc. TopoInVis*, 2013.

[2] I. N. Bernstein, I. M. Gelfand, and V. A. Ponomarev. Coxeter functors and Gabriel's theorem. *Russian Mathematical Surveys*, 28(2):17–32, 1973.

[3] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In *Proc. European Symposium on Algorithms*, pages 695–706, 2013.

[4] Jean-Daniel Boissonnat and Clément Maria. Computing persistent homology with various coefficient fields in a single pass. In *Proc. European Symposium on Algorithms*, 2014. To appear.

[5] G. Carlsson, A. Zomorodian, A. Collins, and L. J. Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(2):149–187, 2005.

[6] Gunnar Carlsson, Tigran Ishkhanov, Vin Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, 76:1–12, January 2008.

[7] Gunnar E. Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

[8] Gunnar E. Carlsson, Vin de Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Symposium on Computational Geometry*, pages 247–256, 2009.

[9] Huang-Wei Chang, Sergio Bacallado, Vijay S. Pande, and Gunnar E. Carlsson. Persistent topology and metastable state in conformational dynamics. *PLoS ONE*, 8(4):e58699, 04 2013.

[10] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-Hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum (proc. SGP 2009)*, pages 1393–1403, 2009.

[11] F. Chazal and S. Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Proc. 24th. Annu. Sympos. Comput. Geom.*, pages 231–241, 2008.

[12] Frédéric Chazal, Leonidas J. Guibas, Steve Oudot, and Primoz Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6):41:1–41:38, November 2013.

[13] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Symposium on Computational Geometry*, pages 119–126, 2006.

[14] Vin de Silva and Gunnar Carlsson. Topological estimation using witness complexes. *IEEE Symposium on Point-based Graphic*, pages 157–166, 2004.

[15] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *CoRR*, abs/1107.5665, 2011.

[16] Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. In *Symposium on Computational Geometry*, page 345, 2014.

[17] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.

[18] Nikola Milosavljevic, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Symposium on Comp. Geom.*, 2011.

[19] Dmitriy Morozov. `Dionysus`. http://www.mrzv.org/software/dionysus/.

[20] Dmitriy Morozov. *Homological Illusions of Persistence and Stability*. Ph.D. dissertation, Department of Computer Science, Duke University, 2008.

[21] James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984.

[22] Steve Y. Oudot and Donald R. Sheehy. Zigzag Zoology: Rips Zigzags for Homology Inference. *J. Foundations of Computational Mathematics*, 2014. To appear.

[23] Afra Zomorodian and Gunnar E. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.