



Computing Generalized Rank Invariant for 2-Parameter Persistence Modules via Zigzag Persistence and Its Applications

Tamal K. Dey¹ · Woojin Kim² · Facundo Mémoli³

Received: 14 July 2022 / Revised: 20 August 2023 / Accepted: 23 August 2023 /
Published online: 7 October 2023
© The Author(s) 2023

Abstract

The notion of generalized rank in the context of multiparameter persistence has become an important ingredient for defining interesting homological structures such as generalized persistence diagrams. However, its efficient computation has not yet been studied in the literature. We show that the generalized rank over a finite interval I of a \mathbf{Z}^2 -indexed persistence module M is equal to the generalized rank of the zigzag module that is induced on a certain path in I tracing mostly its boundary. Hence, we can compute the generalized rank of M over I by computing the barcode of the zigzag module obtained by restricting to that path. If M is the homology of a bifiltration F of t simplices (while accounting for multi-criticality) and I consists of t points, this computation takes $O(t^\omega)$ time where $\omega \in [2, 2.373)$ is the exponent of matrix multiplication. We apply this result to obtain an improved algorithm for the following problem. Given a bifiltration inducing a module M , determine whether M is interval decomposable and, if so, compute all intervals supporting its indecomposable summands.

Editor in Charge: Kenneth Clarkson

Tamal K. Dey
tamaldey@purdue.edu

Woojin Kim
woojin.kim.math@gmail.com

Facundo Mémoli
facundo.memoli@gmail.com

¹ Department of Computer Science, Purdue University, West Lafayette, IN, USA

² Department of Mathematical Sciences, KAIST, Daejeon, South Korea

³ Department of Mathematics and Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

Keywords Persistence diagram · Persistent homology · Möbius inversion · Quiver representations · Multiparameter persistence

Mathematics Subject Classification 16G20 · 55N31 · 62R40 · 68U05

1 Introduction

In Topological Data Analysis (TDA) one of the central tasks is that of decomposing persistence modules into direct sums of *indecomposables*. In the case of a persistence module M over the integers \mathbf{Z} , which implies that M is isomorphic to a direct sum of *interval* modules $\mathbb{I}([b_\alpha, d_\alpha])$, for integers $b_\alpha \leq d_\alpha$ and α in some index set A [1, 2]. The multiset of intervals $\{[b_\alpha, d_\alpha], \alpha \in A\}$ that appear in this decomposition constitutes the *persistence diagram*, or equivalently, the *barcode* of M —a central object in TDA [3, 4].

There are many situations in which data naturally induce persistence modules over posets which are different from \mathbf{Z} [5–14]. Unfortunately, the situation already becomes ‘wild’ when the domain poset is \mathbf{Z}^2 . In that situation, one must contend with the fact that a direct analogue of the notion of persistence diagrams may not exist [8], namely it may not be possible to obtain a lossless up-to-isomorphism representation of the module as a direct sum of interval modules.

Much energy has been put into finding ways in which one can extract incomplete but still stable invariants from persistence modules $M : \mathbf{Z}^d$ (or \mathbf{R}^d) $\rightarrow \mathbf{vec}$ (which we will refer to as a \mathbf{Z}^d -module). Biasotti et al. [15] considered the restriction of an \mathbf{R}^d -module to lines with direction vectors in the positive orthant.¹ This was further developed by Lesnick and Wright in the RIVET project [16] which facilitates the interactive visualization of \mathbf{Z}^2 -modules. Cai et al. [17] considered a certain elder-rule on the \mathbf{Z}^2 -modules which arise in multiparameter clustering. Other efforts have attempted to identify algebraic conditions guaranteeing that M can be decomposed into interval modules of varying degrees of complexity. This has been fully realized in [18, 19] in the case when the intervals in question are rectangles. The case of more complicated intervals has been approached in [20–22] through the lens of Möbius inversion.

The notion of Möbius inversion was first explicitly recognized in the TDA literature by Patel in [23] through the reinterpretation of the persistence diagram of a \mathbf{Z} -module as the Möbius inversion of its rank function. Patel’s work was then extended by Kim and Mémoli [22] to the setting of modules defined over any suitable locally finite poset. They generalized the rank invariant via the *limit-to-colimit map* over subposets and then conveniently expressed its Möbius inversion. In fact the limit-to-colimit map was suggested by Amit Patel to the authors of [22] who in [24] used it to define a notion of rank invariant for zigzag modules. Chambers and Letscher [25] also considered a notion of persistent homology over directed acyclic graphs using the limit-to-colimit map. Asashiba et al. [20] study the case of modules defined on an $m \times n$ grid and propose a high-level algorithm for computing both their generalized rank function and

¹ When $d = 2$ this means lines with positive slope.

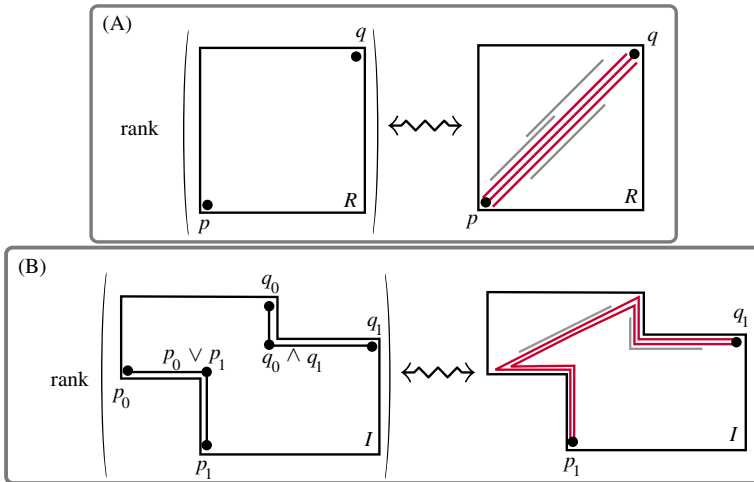


Fig. 1 Generalized rank via zigzag persistence. Let M be a \mathbf{Z}^2 -module. (A) Standard rank: Let $p \leq q$ in \mathbf{Z}^2 . The rank of the structure map from p to q coincides with the multiplicity of full bars (red) over the diagonal path, which is three. (B) Generalized rank: Let I be an interval of \mathbf{Z}^2 . Let us consider the zigzag poset $\partial I : p_1 \leq (p_0 \vee p_1) \geq p_0 \leq q_0 \geq (q_0 \wedge q_1) \leq q_1$. The generalized rank of M over I is equal to the multiplicity of full bars (red) in the zigzag module $M_{\partial I}$, which is two (Theorem 3.12) (note: by definition, the zigzag poset ∂I does not fully inherit the partial order on \mathbf{Z}^2 . For example, the partial order on ∂I does not contain the pair (p_1, q_1) whereas $p_1 \leq q_1$ in \mathbf{Z}^2)

their Möbius inversions with the goal of providing an approximation of a given module by interval decomposables. Asashiba et al. [26] tackle the interval decomposability of a given \mathbf{Z}^d -module via quiver representation theory.

One fundamental algorithmic problem is that of determining whether a given \mathbf{Z}^2 -module is interval decomposable, and if so, computing the intervals. There are some existing solutions to this problem in the literature. Suppose that the input \mathbf{Z}^2 -module is induced by a bifiltration (Definition 4.1) comprising insertions of at most t simplices on a grid of cardinality $O(t)$. First, the decomposition algorithm by Dey and Xin [27] can produce all indecomposables from such a module in $O(t^{2\omega+1})$ time (see [28] for comments about its implementation) where $\omega \in [2, 2.373)$ is the exponent of matrix multiplication. Given these indecomposables, one could then test whether they are indeed interval modules. However, the algorithm requires that the input module be such that no two generators or relations in the module have the same grade. Then, Asashiba et al. [26] give an algorithm which requires enumerating an exponential number (in t) of intervals. Finally, the algorithm by Meataxe sidesteps both of the above issues, but incurs a worst-case cost of $O(t^{18})$ as explained in [27].

See also [29–33] for related recent work.

Contributions.

One of our key results is as follows. Let M be a \mathbf{Z}^2 -module. We prove that the generalized rank of M over any finite interval I in \mathbf{Z}^2 can be computed using the zigzag persistence barcode of the restriction of M to a certain zigzag path within I (Theorem 3.12). See Fig. 1 for an illustration. By utilizing this theorem, we establish two algorithms described subsequently.

Assume that the input is a filtration \mathcal{F} of a simplicial complex K over an interval I of \mathbf{Z}^2 such that I and K consist of $O(t)$ points and simplices, respectively. The filtration can be *multicritical*, i.e. for any simplex $\sigma \in K$, the subset of points in I at which σ is present can admit multiple minimal points. When the filtration \mathcal{F} is multicritical, we assume that $\sum_{\sigma \in K} n(\sigma) = O(t)$ where $n(\sigma)$ is the number of minimal points in I at which σ is present.

Let $M_{\mathcal{F}}$ be the persistence module that is obtained by applying the homology functor to the filtration \mathcal{F} .

- Given that $M_{\mathcal{F}}$ is interval decomposable, we provide an algorithm INTERVAL (page 19) to compute the barcode of $M_{\mathcal{F}}$ in time $O(t^{\omega+2})$ (Proposition 4.15).
- We provide an algorithm ISINTERVALDECOMP (page 23) to decide the interval decomposability of $M_{\mathcal{F}}$ in time $O(t^{3\omega+2})$ (Proposition 4.17).

2 Preliminaries

In Sect. 2.1, we review the notion of interval decomposability of persistence modules. In Sect. 2.2, we review the notions of generalized rank invariant and generalized persistence diagram. In Sect. 2.3, we discuss how to compute the limit and the colimit of a given functor $P \rightarrow \mathbf{vec}$.

2.1 Persistence Modules and Their Decompositions

We fix a certain field \mathbb{F} and every vector space in this paper is over \mathbb{F} . Let \mathbf{vec} denote the category of *finite dimensional* vector spaces and linear maps.

Let P be a poset. We regard P as the category that has points of P as objects. Also, for any $p, q \in P$, there exists a unique morphism $p \rightarrow q$ if and only if $p \leq q$. For a positive integer d , let \mathbf{Z}^d be given the partial order defined by $(a_1, a_2, \dots, a_d) \leq (b_1, b_2, \dots, b_d)$ if and only if $a_i \leq b_i$ for $i = 1, 2, \dots, d$.

A (P -indexed) **persistence module** is any functor $M : P \rightarrow \mathbf{vec}$ (which we will simply refer to as a P -module). In other words, to each $p \in P$, a vector space M_p is associated, and to each pair $p \leq q$ in P , a linear map $\varphi_M(p, q) : M_p \rightarrow M_q$ is associated. Importantly, whenever $p \leq q \leq r$ in P , it must be that $\varphi_M(p, r) = \varphi_M(q, r) \circ \varphi_M(p, q)$.

We say that a pair of $p, q \in P$ is **comparable** if either $p \leq q$ or $q \leq p$.

Definition 2.1 ([34]) An **interval** I of P is a subset $I \subseteq P$ such that:

- I is nonempty.
- If $p, q \in I$ and $p \leq r \leq q$, then $r \in I$.
- I is **connected**, i.e. for any $p, q \in I$, there is a sequence $p = p_0, p_1, \dots, p_\ell = q$ of elements of I with p_i and p_{i+1} comparable for $0 \leq i \leq \ell - 1$.

By $\mathbf{Int}(P)$ we denote the set of all *finite* intervals of P . When P is finite and connected, $P \in \mathbf{Int}(P)$ will be referred to as **the full interval**.

For an interval I of P , the **interval module** $\mathbb{I}_I : P \rightarrow \mathbf{vec}$ is defined as

$$\mathbb{I}_I(p) = \begin{cases} \mathbb{F} & \text{if } p \in I, \\ 0 & \text{otherwise,} \end{cases} \quad \varphi_{\mathbb{I}_I}(p, q) = \begin{cases} \text{id}_{\mathbb{F}} & \text{if } p, q \in I, \ p \leq q, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.2 Let M be any P -module. A **submodule** N of M is defined by subspaces $N_p \subseteq M_p$ such that $\varphi_M(p, q)(N_p) \subseteq N_q$ for all $p, q \in P$ with $p \leq q$. These conditions guarantee that N itself is a P -module, with the structure maps given by the restrictions $\varphi_M(p, q)|_{N_p}$. In this case we write $N \leq M$.

A submodule N is a **summand** of M if there exists a submodule N' which is complementary to N , i.e. $M_p = N_p \oplus N'_p$ for all p . In that case, we say that M is a *direct sum* of N, N' and write $M \cong N \oplus N'$. Note that this direct sum is an *internal* direct sum.

The (external) **direct sum** $M_1 \oplus M_2$ of given P -modules M_1 and M_2 is the P -module where $(M \oplus N)_p = M_p \oplus N_p$ for $p \in P$ and $\varphi_{M \oplus N}(p, q) = \varphi_M(p, q) \oplus \varphi_N(p, q)$ for $p \leq q$ in P . A nonzero P -module M is **indecomposable** if whenever $M = M_1 \oplus M_2$ for some P -modules M_1 and M_2 , either $M_1 = 0$ or $M_2 = 0$.

Definition 2.3 A P -module M is called **interval decomposable** if M is isomorphic to a direct sum of interval modules, i.e. there exists an indexing set \mathcal{J} such that $M \cong \bigoplus_{j \in \mathcal{J}} \mathbb{I}_{I_j}$ (external direct sum). In this case, the multiset $\{I_j : j \in \mathcal{J}\}$ is called the **barcode** of M , which will be denoted by $\text{barc}(M)$.

The Azumaya–Krull–Remak–Schmidt theorem [35] guarantees that $\text{barc}(M)$ is well-defined, i.e. the multiset $\{I_j : j \in \mathcal{J}\}$ is unique.

Consider a *zigzag poset* of n points, $\bullet_1 \leftrightarrow \bullet_2 \leftrightarrow \dots \leftrightarrow \bullet_{n-1} \leftrightarrow \bullet_n$ where \leftrightarrow stands for either \leq or \geq . A functor from a zigzag poset to \mathbf{vec} is called a **zigzag module** [6]. Any zigzag module is interval decomposable [1] and thus admits a barcode.

The following proposition directly follows from the Azumaya–Krull–Remak–Schmidt theorem and will be useful in Sect. 4.

Proposition 2.4 Let $M : P \rightarrow \mathbf{vec}$ be interval decomposable and let $N \leq M$ be a summand of M (Definition 2.2). Then, M/N is interval decomposable.

Proof It suffices to show that $M \cong N \oplus (M/N)$ (since M is interval decomposable, this isomorphism and Azumaya–Krull–Remak–Schmidt’s theorem [35] guarantee that M/N is also interval decomposable). Let $\iota : N \hookrightarrow M$ be the canonical inclusion. Since N is a *summand* of M , there exists a complementary submodule $N' \leq M$ with $M = N \oplus N'$ (internal direct sum). For the inclusion $\iota : N' \hookrightarrow M$ and the projection $\pi : M \rightarrow M/N$, the composition $\pi \circ \iota$ is an isomorphism, completing the proof. \square

2.2 Generalized Rank Invariant and Generalized Persistence Diagrams

Throughout this section, let P be a finite connected poset. Consider any P -module M . Then M admits a limit $\varprojlim M = (L, (\pi_p : L \rightarrow M_p)_{p \in P})$ and a colimit $\varinjlim M = (C, (i_p : M_p \rightarrow C)_{p \in P})$; see Appendix 1. For every $p \leq q$ in P , we have

$$\varphi_M(p \leq q) \circ \pi_p = \pi_q \text{ and } i_q \circ \varphi_M(p \leq q) = i_p.$$

By the connectedness of P , we have $i_p \circ \pi_p = i_q \circ \pi_q : L \rightarrow C$ for any $p, q \in P$. This equality ensures the validity of the following definition.

Definition 2.5 ([22]) The **canonical limit-to-colimit map** $\psi_M : \varprojlim M \rightarrow \varinjlim M$ is the linear map $i_p \circ \pi_p$ for any $p \in P$. The **generalized rank** of M is $\text{rank}(\overline{M}) := \text{rank}(\psi_M)$.

The rank of M counts the multiplicity of the fully supported interval modules in a direct sum decomposition of M .

Theorem 2.6 ([25, Lem. 3.1]) *The rank of M is equal to the number of indecomposable summands of M which are isomorphic to the interval module \mathbb{I}_P .*

Definition 2.7 The **(Int)-generalized rank invariant** of M is the map $\text{rk}_{\mathbb{I}}(M) : \mathbf{Int}(P) \rightarrow \mathbf{Z}_{\geq 0}$ defined as $I \mapsto \text{rank}(M|_I)$, where $M|_I$ is the restriction of M to I .

Remark 2.8 (Additivity) Given any P -modules M and N , it is not hard to show that $\text{rank}(M \oplus N) = \text{rank}(M) + \text{rank}(N)$. This implies that $\text{rk}_{\mathbb{I}}(M \oplus N) = \text{rk}_{\mathbb{I}}(M) + \text{rk}_{\mathbb{I}}(N)$.

Definition 2.9 The **(Int)-generalized persistence diagram** of M is the unique² function $\text{dgm}_{\mathbb{I}}(M) : \mathbf{Int}(P) \rightarrow \mathbf{Z}$ that satisfies, for any $I \in \mathbf{Int}(P)$,

$$\text{rk}_{\mathbb{I}}(M)(I) = \sum_{\substack{J \supseteq I \\ J \in \mathbf{Int}(P)}} \text{dgm}_{\mathbb{I}}(M)(J).$$

The following is a slight variation of [22, Thm. 3.14] and [20, Thm. 5.10].

Theorem 2.10 *If a given $M : P \rightarrow \mathbf{vec}$ is interval decomposable, then for all $I \in \mathbf{Int}(P)$, $\text{dgm}_{\mathbb{I}}(M)(I)$ is equal to the multiplicity of I in $\text{barc}(M)$.*

Proof It is not hard to verify that for any $I, J \in \mathbf{Int}(P)$,

$$\text{rk}_{\mathbb{I}}(\mathbb{I}_J)(I) = \begin{cases} 1, & J \supseteq I \\ 0 & \text{else.} \end{cases}$$

For $J \in \mathbf{Int}(P)$, let $\mu_J \in \mathbf{Z}_{\geq 0}$ be the multiplicity of J in $\text{barc}(M)$. We have $M \cong \bigoplus_{J \in \mathbf{Int}(P)} \mathbb{I}_J^{\mu_J}$ where $\mathbb{I}_J^{\mu_J}$ is the direct sum of μ_J copies of \mathbb{I}_J . Then, by Theorem 2.6 and Remark 2.8, we have that for all $I \in \mathbf{Int}(P)$

$$\text{rk}_{\mathbb{I}}(M)(I) = \text{rk}_{\mathbb{I}}\left(\bigoplus_{J \in \mathbf{Int}(P)} \mathbb{I}_J^{\mu_J}\right)(I) = \sum_{J \in \mathbf{Int}(P)} \mu_J \cdot \text{rk}_{\mathbb{I}}(\mathbb{I}_J)(I) = \sum_{\substack{J \supseteq I \\ J \in \mathbf{Int}(P)}} \mu_J.$$

² The existence and uniqueness are guaranteed by properties of the Möbius inversion formula [36, 37].

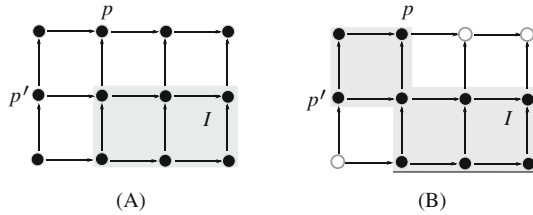


Fig. 2 **A** For the given interval I of \mathbf{Z}^2 , we have $\text{nbdl}_{\mathbb{I}}(I) = \{p, p'\}$. For $A = \{p, p'\}$, the set $A \cup I$ is not an interval of \mathbf{Z}^2 . **B** The points in the shaded region form $\overline{A \cup I}$

By the uniqueness of $\text{dgm}_{\mathbb{I}}(M)$ mentioned in Definition 2.9, $\text{dgm}_{\mathbb{I}}(M)(I) = \mu_I$ for all $I \in \text{Int}(P)$. □

In the rest of this subsection, we consider P to be a finite interval of the product poset \mathbf{Z}^2 . Recall that $\text{Int}(\mathbf{Z}^2)$ denotes the set of finite intervals of \mathbf{Z}^2 .

Definition 2.11 For any $I \in \text{Int}(\mathbf{Z}^2)$, we define $\text{nbdl}_{\mathbb{I}}(I) := \{p \in \mathbf{Z}^2 \setminus I : I \cup \{p\} \in \text{Int}(\mathbf{Z}^2)\}$.

Note that $\text{nbdl}_{\mathbb{I}}(I)$ is nonempty [20, Prop. 3.2]. When $A \subseteq \text{nbdl}_{\mathbb{I}}(I)$ contains more than one point, $A \cup I$ is not necessarily an interval of \mathbf{Z}^2 . However, there always exists a unique smallest interval that contains $A \cup I$, which is denoted by $\overline{A \cup I}$. See Fig. 2 for an illustrative example.

Remark 2.12 ([20, Thm. 5.3]) If in Definition 2.9 we assume that P is a finite interval of \mathbf{Z}^2 , then we have that for every $I \in \text{Int}(P)$,³

$$\text{dgm}_{\mathbb{I}}(M)(I) = \text{rk}_{\mathbb{I}}(M)(I) + \sum_{\substack{A \subseteq \text{nbdl}_{\mathbb{I}}(I) \cap P \\ A \neq \emptyset}} (-1)^{|A|} \text{rk}_{\mathbb{I}}(M)\left(\overline{A \cup I}\right). \tag{1}$$

2.3 Canonical Constructions of Limits and Colimits

Let M be any P -module.

Notation 2.13 Let $p, q \in P$ and let $v_p \in M_p$ and $v_q \in M_q$. We write $v_p \sim v_q$ if p and q are comparable, and either v_p is mapped to v_q via $\varphi_M(p, q)$ or v_q is mapped to v_p via $\varphi_M(q, p)$.

The following proposition gives a standard way of constructing a limit and a colimit of a P -module M . Since it is well-known, we do not prove it (see for example [38, Chap. 5]).

³ In [20], only the case $P = \{1, \dots, m\} \times \{1, \dots, n\} \subset \mathbf{Z}^2$ was considered. However, it is not difficult to check that Eq. (1) is still valid for any finite interval P in \mathbf{Z}^2 and any subinterval $I \subseteq P$.

Proposition 2.14 (i) *The limit of M is (isomorphic to) the pair $(W, (\pi_p)_{p \in P})$ where*

$$W := \left\{ (v_p)_{p \in P} \in \prod_{p \in P} M_p : \forall p \leq q \text{ in } P, v_p \sim v_q \right\} \quad (2)$$

and for each $p \in P$, the map $\pi_p : W \rightarrow M_p$ is the canonical projection. An element of W is called a **section** of M .⁴

(ii) *The colimit of M is (isomorphic to) the pair $(U, (i_p)_{p \in P})$ described as follows: For $p \in P$, let the map $j_p : M_p \hookrightarrow \bigoplus_{p \in P} M_p$ be the canonical injection. U is the quotient $(\bigoplus_{p \in P} M_p) / T$, where T is the subspace of $\bigoplus_{p \in P} M_p$ that is generated by the vectors of the form $j_p(v_p) - j_q(v_q)$, $v_p \sim v_q$, the map $i_p : M_p \rightarrow U$ is the composition $\rho \circ j_p$, where ρ is the quotient map $\bigoplus_{p \in P} M_p \rightarrow U$.*

Setup 1 In the rest of the paper, limits and colimits of a P -module M will all be constructed as in Proposition 2.14.

3 Computing Generalized Rank via Boundary Zigzags

In Sect. 3.1 we introduce the notions of lower and upper fences of a poset. In Sect. 3.2, we introduce the notion of boundary cap ∂I of a finite interval I of \mathbf{Z}^2 , which is a path, a certain sequence of points in I . In Sect. 3.3, we show that the rank of any functor $M : I \rightarrow \mathbf{vec}$ can be obtained by computing the barcode of the zigzag module over the path ∂I .

3.1 Lower and Upper Fences of a Poset

Let P be any connected poset. Given any $p \in P$, by p^\downarrow , we denote the set of all elements of P that are less than or equal to p . Dually p^\uparrow is defined as the set of all elements of P that are greater than or equal to p .

Definition 3.1 A subposet $L \subset P$ (resp. $U \subset P$) is called a **lower (resp. upper) fence** of P if L is connected, and for any $q \in P$, the intersection $L \cap q^\downarrow$ (resp. $U \cap q^\uparrow$) is nonempty and connected.

The following proposition is a crucial milestone towards establishing our main result, Theorem 3.12.

Proposition 3.2 *Let L and U be a lower and an upper fences of a connected poset P respectively. Given any P -module M , we have $\varprojlim M \cong \varprojlim M|_L$ and $\varinjlim M \cong \varinjlim M|_U$.*

⁴ When P is finite, $\prod_{p \in P} M_p = \bigoplus_{p \in P} M_p$.

Although this proposition directly follows from the fact that the inclusions $L \hookrightarrow P$ and $U \hookrightarrow P$ are respectively *initial* and *final* functors (see [38, Sect. 9.3]), an elementary and tangible proof is given below, which utilizes the following proposition.

Proposition 3.3 *Let M be a P -module and let $L \subset P$ be a lower fence. Let $\mathbf{v} \in \bigoplus_{p \in I} M_p$. The tuple \mathbf{v} belongs to $\varprojlim M$ if and only if for every $p \in P$, it holds that $\mathbf{v}_p = \varphi_M(q, p)(\mathbf{v}_q)$ for every $q \in L$ such that $q \leq p$.*

Proof The forward direction is straightforward from the description of $\varprojlim M$ given in Proposition 2.14 (i). Let us show the backward direction. Again, by Proposition 2.14 (i) it suffices to show that for all $r \leq p$ in P it holds that $\mathbf{v}_p = \varphi_M(r, p)(\mathbf{v}_r)$. Fix $r \leq p$ in P . Since L is a lower fence, there exists $q \in L$ such that $q \leq r \leq p$. Then, by assumption and by functoriality of M , we have

$$\begin{aligned} \mathbf{v}_p &= \varphi_M(q, p)(\mathbf{v}_q) = (\varphi_M(r, p) \circ \varphi_M(q, r))(\mathbf{v}_q) \\ &= \varphi_M(r, p)(\varphi_M(q, r)(\mathbf{v}_q)) = \varphi_M(r, p)(\mathbf{v}_r), \end{aligned}$$

as desired. □

Proof of Proposition 3.2 We only prove $\varprojlim M \cong \varprojlim M|_L$. It suffices to prove that the section extension map $e : \varprojlim M|_L \rightarrow \varprojlim M$ in (3) is bijective. The injectivity is clear by definition. The surjectivity follows from the forward direction of the statement in Proposition 3.3. □

The canonical isomorphism $\varprojlim M \cong \varprojlim M|_L$ in Proposition 3.2 is given by the canonical *section extension* $e : \varprojlim M|_L \rightarrow \varprojlim M$. Namely,

$$e : (\mathbf{v}_p)_{p \in L} \mapsto (\mathbf{w}_q)_{q \in P}, \tag{3}$$

where for any $q \in P$, the vector \mathbf{w}_q is defined as $\varphi_M(p, q)(\mathbf{v}_p)$ for any $p \in L \cap q^\downarrow$; the connectedness of $L \cap q^\downarrow$ guarantees that \mathbf{w}_q is well-defined. Also, if $q \in L$, then $\mathbf{w}_q = \mathbf{v}_q$. The inverse $r := e^{-1}$ is the canonical section restriction. The other isomorphism $\varprojlim M \cong \varinjlim M|_U$ in Proposition 3.2 is given by the map $i : \varinjlim M|_U \rightarrow \varinjlim M$ defined by $[v_p] \mapsto [v_p]$ for any $p \in U$ and any $v_p \in M_p$; the fact that this map i is well-defined will become clear from Proposition 3.10. Let us define $\xi : \varprojlim M|_L \rightarrow \varinjlim M|_U$ by $i^{-1} \circ \psi_M \circ e$. By construction, the following diagram commutes

$$\begin{array}{ccc} \varprojlim M|_L & \xrightarrow{\xi} & \varinjlim M|_U \\ \cong \downarrow e & & \cong \downarrow i \\ \varprojlim M & \xrightarrow{\psi_M} & \varinjlim M, \end{array} \tag{4}$$

where ψ_M is the canonical limit-to-colimit map of M . Hence we have the fact

$$\text{rank}(\psi_M) = \text{rank}(\xi), \tag{5}$$

which is useful for proving Theorem 3.12.

3.2 Boundary Cap of an Interval in \mathbf{Z}^2

Let $I \in \mathbf{Int}(\mathbf{Z}^2)$, i.e. I is a finite interval of \mathbf{Z}^2 (Definition 2.1). By $\min(I)$ and $\max(I)$, we denote the collections of minimal and maximal elements of I , respectively. In other words,

$$\begin{aligned}\min(I) &:= \{p \in I : \text{there is no } q \in I \text{ s.t. } q < p\}, \\ \max(I) &:= \{p \in I : \text{there is no } q \in I \text{ s.t. } p < q\}.\end{aligned}$$

Note that $\min(I)$ and $\max(I)$ are nonempty and that $\min(I)$ and $\max(I)$ respectively form an *antichain* in I , i.e. any two different points in $\min(I)$ (or in $\max(I)$) are *not* comparable.

Remark 3.4 (i) The least upper bound and the greatest lower bound of $p, q \in \mathbf{Z}^2$ are denoted by $p \vee q$ and $p \wedge q$ respectively. Let $p = (p_x, p_y)$ and $q = (q_x, q_y)$ in \mathbf{Z}^2 . Then,

$$p \vee q = (\max\{p_x, q_x\}, \max\{p_y, q_y\}), \quad p \wedge q = (\min\{p_x, q_x\}, \min\{p_y, q_y\}).$$

(ii) Let $I \in \mathbf{Int}(\mathbf{Z}^2)$. Since $\min(I)$ is a finite antichain, we can list the elements of $\min(I)$ in ascending order of their x -coordinates, i.e. $\min(I) := \{p_0, \dots, p_k\}$ and such that for each $i = 0, \dots, k$, the x -coordinate of p_i is less than that of p_{i+1} . Similarly, let $\max(I) := \{q_0, \dots, q_\ell\}$ be ordered in ascending order of q_j 's x -coordinates. We have that $p_0 \leq q_0$ (Fig. 3).

Based on Remark 3.4 (ii) above, we define the following:

Definition 3.5 (Lower and upper zigzags of an interval) Let I , $\min(I)$, and $\max(I)$ be as in Remark 3.4 (ii). We define the following two zigzag posets (Fig. 3):

$$\begin{aligned}\min_{\mathbf{ZZ}}(I) &:= \{p_0 < (p_0 \vee p_1) > p_1 < (p_1 \vee p_2) > \dots < (p_{k-1} \vee p_k) > p_k\} \\ &= \min(I) \cup \{p_i \vee p_{i+1} : i = 0, \dots, k-1\},\end{aligned}\tag{6}$$

$$\begin{aligned}\max_{\mathbf{ZZ}}(I) &:= \{q_0 > (q_0 \wedge q_1) < q_1 > (q_1 \wedge q_2) < \dots > (q_{\ell-1} \wedge q_\ell) > q_\ell\} \\ &= \max(I) \cup \{q_i \wedge q_{i+1} : i = 0, \dots, \ell-1\}.\end{aligned}\tag{7}$$

Note that $\min_{\mathbf{ZZ}}(I)$ and $\max_{\mathbf{ZZ}}(I)$ are lower and upper fences of I respectively.

For $p, q \in P$, let us write $p \triangleleft q$ if $p < q$ and there is no $r \in P$ such that $p < r < q$. Similarly, we write $p \triangleright q$ if $p > q$ and there is no $r \in P$ such that $p > r > q$.

Definition 3.6 Given a poset P , a **path** Γ between two points $p, q \in P$ is a sequence of points $p = p_0, \dots, p_k = q$ in P such that either $p_i \leq p_{i+1}$ or $p_i \geq p_{i+1}$ for every $i \in [1, k-1]$ (note that there can be a pair $i \neq j$ such that $p_i = p_j$). The path Γ is said to be **monotonic** if $p_i \leq p_{i+1}$ for each i . The path Γ is called **faithful** if either $p_i \triangleleft p_{i+1}$ or $p_i \triangleright p_{i+1}$ for each i .

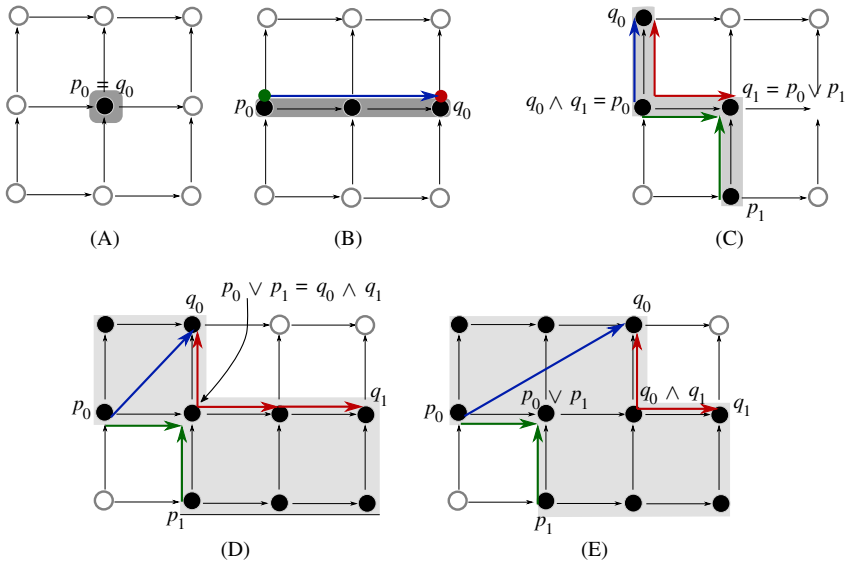


Fig. 3 Five different intervals I of \mathbb{Z}^2 . Relations in $\min_{\mathbb{Z}\mathbb{Z}}(I)$ and $\max_{\mathbb{Z}\mathbb{Z}}(I)$ are indicated by green and red arrows, respectively. The inequality $p_0 \leq q_0$ is indicated by blue arrows unless $p_0 = q_0$. Notice that ∂I , as defined in equation (8), has cardinality 2, 2, 6, 6, 6 in that order ((A),(B),(C),(D),(E))

Definition 3.7 (Boundary cap of an interval) We define the **boundary cap** ∂I of $I \in \mathbf{Int}(\mathbb{Z}^2)$ as the path obtained by concatenating $\min_{\mathbb{Z}\mathbb{Z}}(I)$ and $\max_{\mathbb{Z}\mathbb{Z}}(I)$ in Eqs. (6) and (7).

$$\partial I := \underbrace{p_k < (p_k \vee p_{k-1}) > p_{k-1} < \cdots > p_0 \leq q_0}_{2k+1 \text{ terms from } \min_{\mathbb{Z}\mathbb{Z}}(I)} \underbrace{> (q_0 \wedge q_1) < q_1 > \cdots < q_\ell}_{2\ell+1 \text{ terms from } \max_{\mathbb{Z}\mathbb{Z}}(I)} \tag{8}$$

We remark that ∂I can contain multiple copies of the same point. Namely, there can be $i \in [0, k]$ and $j \in [0, \ell]$ such that either $p_i = q_j$ (Fig. 3 (A)), $p_i = q_j \wedge q_{j+1}$ (Fig. 3 (C)), $p_i \vee p_{i+1} = q_j$ (Figure 3 (C)), or $p_i \vee p_{i+1} = q_j \wedge q_{j+1}$ (Fig. 3 (D)).

Consider the following zigzag poset of the same length as ∂I :

$$\mathbb{Z}\mathbb{Z}_{\partial I} : \underbrace{\bullet_1 < \bullet_2 > \bullet_3 < \cdots > \bullet_{2k+1}}_{2k+1} < \underbrace{\circ_1 > \circ_2 < \circ_3 > \cdots < \circ_{2\ell+1}}_{2\ell+1} . \tag{9}$$

Still using the notation in Eqs. (8) we have the following order-preserving map

$$\iota_I : \mathbb{Z}\mathbb{Z}_{\partial I} \rightarrow I \tag{10}$$

whose image is ∂I : \bullet_1 is sent to p_k , \bullet_2 is sent to $p_k \vee p_{k-1}$, ..., and $\circ_{2\ell+1}$ is sent to q_ℓ .

3.3 Generalized Rank Invariant via Boundary Zigzags

The goal of this section is to establish one of our main results, Theorem 3.12. Let P be a poset and let M be any P -module.

Definition 3.8 Let $\Gamma : p_0, \dots, p_k$ be a path in P . A $(k + 1)$ -tuple $\mathbf{v} \in \bigoplus_{i=0}^k M_{p_i}$ is called the **section of M along Γ** if $\mathbf{v}_{p_i} \sim \mathbf{v}_{p_{i+1}}$ for each i (Notation 2.13).

We remark that a path Γ can contain multiple copies of the same point in P . The next example shows that a section \mathbf{v} of M along $\Gamma : p_0, \dots, p_k$ does not necessarily belong to the limit of the restriction of M to the subposet $\{p_0, \dots, p_k\} \subseteq P$.

Example 3.9 Consider $M : \{(1, 1), (1, 2), (2, 2), (2, 1)\} \subset \mathbf{Z}^2 \rightarrow \mathbf{vec}$ given as follows.

$$\begin{array}{ccc}
 M_{(1,2)} & \longrightarrow & M_{(2,2)} \\
 \uparrow & & \uparrow \\
 M_{(1,1)} & \longrightarrow & M_{(2,1)}
 \end{array}
 =
 \begin{array}{ccc}
 \mathbb{F} & \xrightarrow{1} & \mathbb{F} \\
 \uparrow 1 & & \uparrow (1 \ 1) \\
 \mathbb{F} & \xrightarrow{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} & \mathbb{F}^2
 \end{array}$$

Consider the path $\Gamma : (1, 1), (1, 2), (2, 2), (2, 1)$ which contains all points in the indexing poset. Then, $\mathbf{v} := (1, 1, 1, (0, 1)) \in M_{(1,1)} \oplus M_{(1,2)} \oplus M_{(2,2)} \oplus M_{(2,1)}$ is a section of M along Γ , while $\mathbf{v} \notin \varprojlim M$.

By Proposition 2.14 (ii), we directly have:

Proposition 3.10 Let $p, q \in P$. For any vectors $v_p \in M_p$ and $v_q \in M_q$, $[v_p] = [v_q]$ in⁵ the colimit $\varinjlim M$ if and only if there exist a path $\Gamma : p = p_0, p_1, \dots, p_n = q$ in P and a section \mathbf{v} of M along Γ such that $\mathbf{v}_p = v_p$ and $\mathbf{v}_q = v_q$.

Setup 2 In the rest of Sect. 3.3, we fix both $I \in \mathbf{Int}(\mathbf{Z}^2)$ and a functor $M : I \rightarrow \mathbf{vec}$. Also, we identify points in (8) and (9) via ι_I .

Definition 3.11 (Zigzag module along ∂I) For the map $\iota_I : \mathbf{ZZ}_{\partial I} \rightarrow I$ in Eq. (10), we define the zigzag module $M_{\partial I} : \mathbf{ZZ}_{\partial I} \rightarrow \mathbf{vec}$ by the composition $M \circ \iota_I$.

We remark that, by Setup 1 and Definition 3.8, each $\mathbf{v} = (\mathbf{v}_x)_{x \in \mathbf{ZZ}_{\partial I}} \in \varprojlim M_{\partial I}$ is an element of $\bigoplus_{x \in \mathbf{ZZ}_{\partial I}} M_{\iota_I(x)}$ that is a section M along ∂I .

One of our main results is the following.

Theorem 3.12 $\text{rank}(M)$ is equal to the multiplicity of the full interval in $\text{barc}(M_{\partial I})$.

⁵ For simplicity, we write $[v_p]$ and $[v_q]$ instead of $[j_p(v_p)]$ and $[j_q(v_q)]$ respectively where $j_p : M_p \rightarrow \bigoplus_{r \in P} M_r$ and $j_q : M_q \rightarrow \bigoplus_{r \in P} M_r$ are the canonical inclusion maps.

Proof By Theorem 2.6, it suffices to show that

$$\text{rank}(\psi_M : \varprojlim M \rightarrow \varinjlim M) = \text{rank}(\psi_{M_{\partial I}} : \varprojlim M_{\partial I} \rightarrow \varinjlim M_{\partial I}).$$

Let $L := \min_{\mathbf{ZZ}}(I)$ and $U := \max_{\mathbf{ZZ}}(I)$ which are lower and upper fences of I respectively. Let us define the maps e, r, i and ξ as described in the paragraph after Proposition 3.2. Then, by equation (5), it suffices to prove that the rank of ξ equals the rank of $\psi_{M_{\partial I}}$. To this end, we show that there exist a surjective linear map $f : \varinjlim M_{\partial I} \rightarrow \varprojlim M|_L$ and an injective linear map $g : \varinjlim M|_U \rightarrow \varinjlim M_{\partial I}$ such that $\psi_{M_{\partial I}} = g \circ \xi \circ f$. We define f as the canonical section restriction $(\mathbf{v}_q)_{q \in \partial I} \mapsto (\mathbf{v}_q)_{q \in L}$. We define g as the canonical map, i.e. $[v_q] \mapsto [v_q]$ for any $q \in U$ and any $v_q \in M_q$. By Proposition 3.10 and by construction of $M_{\partial I}$, the map g is well-defined.

We now show that $\psi_{M_{\partial I}} = g \circ \xi \circ f$. Let $\mathbf{v} := (\mathbf{v}_q)_{q \in \partial I} \in \varprojlim M_{\partial I}$. Then, by definition of $\psi_{M_{\partial I}}$ (Setup 1), the image of \mathbf{v} via $\psi_{M_{\partial I}}$ is $[\mathbf{v}_{q_0}]$ where $q_0 \in U$ is defined as in Remark 3.4 (ii). Also, we have

$$\mathbf{v} \xrightarrow{f} (\mathbf{v}_q)_{q \in L} \xrightarrow{\xi} [\mathbf{v}_{q_0}] (\in \varinjlim M|_U) \xrightarrow{g} [\mathbf{v}_{q_0}] (\in \varinjlim M_{\partial I}),$$

which proves the equality $\psi_{M_{\partial I}} = g \circ \xi \circ f$.

We claim that f is surjective. Let $r' : \varprojlim M \rightarrow \varprojlim M_{\partial I}$ be the canonical section restriction map $(\mathbf{v}_q)_{q \in I} \mapsto (\mathbf{v}_q)_{q \in \partial I}$. Then, the restriction $r : \varprojlim M \rightarrow \varprojlim M|_L$, can be seen as the composition of two restrictions $r = f \circ r'$. Since r is the inverse of the isomorphism e in diagram (4), r is surjective and thus so is f .

Next we claim that g is injective. Let $i' : \varinjlim M_{\partial I} \rightarrow \varinjlim M$ be defined by $[v_q] \mapsto [v_q]$ for any $q \in \partial I$ and any $v_q \in M_q$. By Proposition 3.10 and by construction of $M_{\partial I}$, the map i' is well-defined. Then, for the isomorphism i in diagram (4), we have $i = i' \circ g$. This implies that g is injective. □

Remark 3.13 In Definition 3.7 one may consider the “lower” boundary cap $\widehat{\partial I}$, as an alternative to ∂I :

$$\widehat{\partial I} : p_0 < p_0 \vee p_1 > p_1 < \cdots > p_k \leq q_\ell > q_\ell \wedge q_{\ell-1} < q_{\ell-1} > \cdots < q_0.$$

The value $\text{rank}(M)$ also equals the multiplicity of the full interval in the barcode of the zigzag module induced over $\widehat{\partial I}$.

By Theorem 3.12, we can utilize algorithms for zigzag persistence in order to compute the generalized rank invariant and the generalized persistence diagram of any \mathbf{Z}^2 -module that is obtained by applying the homology functor to a simplicial filtration \mathcal{F} over (a subset of) \mathbf{Z}^2 . For this, we complete the boundary cap of a given interval to a faithful path (i.e. we put the missing monotonic paths between every pair of consecutive points) and then simply run a zigzag persistence algorithm. Notice that, if \mathcal{F} satisfies the condition in Setup 3 below, the above mentioned faithful path admits only $O(t)$ insertions and deletions of simplices in the zigzag filtration and hence the zigzag persistence algorithms of Milosavljevic et al. [39] and of Dey and Hou [40] run in time $O(t^\omega)$.

Remark 3.14 To compute $\text{dgm}_{\mathbb{I}}(M)(I)$ by the formula in (1), one needs to consider terms whose number depends exponentially on the number of neighbors of I . However, for any interval that has at most $O(\log t)$ neighbors, we have $2^{O(\log t)} = t^c$ terms for some constant $c > 0$. It follows that using $O(t^\omega)$ zigzag persistence algorithm for computing generalized ranks, we obtain an $O(t^{\omega+c})$ algorithm for computing generalized persistence diagrams of intervals that have at most $O(\log t)$ neighbors.

4 Computing Intervals and Detecting Interval Decomposability

When a persistence module M admits a summand N that is isomorphic to an interval module, N will be called an **interval summand** of M . In this section, we apply Theorem 3.12 for computing generalized rank via zigzag to different problems that ask to find interval summands of an input finite \mathbf{Z}^2 -module: Problems 4.3, 4.6, and 4.16.

Let K be a finite abstract simplicial complex and let $\text{sub}(K)$ be the poset of all subcomplexes of K , ordered by inclusion.

Definition 4.1 Given any poset P , an order-preserving map $\mathcal{F} : P \rightarrow \text{sub}(K)$ is called a **(simplicial) filtration (of K over P)**. When P is either \mathbf{Z}^2 or an interval of \mathbf{Z}^2 , we call \mathcal{F} a **bifiltration**.

Setup 3 Throughout Sect. 4, \mathcal{F} denotes a filtration of a simplicial complex K over an interval P of \mathbf{Z}^2 such that P and K consist of $O(t)$ points and simplices, respectively. The filtration can be *multicritical*, i.e., for any simplex $\sigma \in K$, the subset of points in P at which σ is present can admit multiple minimal points. We assume that $\sum_{\sigma \in K} n(\sigma) = O(t)$ where $n(\sigma)$ denotes the number of minimal points in P at which σ is present.

By $M_{\mathcal{F}} : P \rightarrow \mathbf{vec}$ we denote the module induced by \mathcal{F} through the homology functor with coefficients in the field \mathbb{F} .

Computing the dimension function.

In all algorithms below, we utilize a subroutine $\text{DIM}(\mathcal{F}, P)$, which computes the dimension of the vector space $(M_{\mathcal{F}})_p$ for every $p \in P$.

Proposition 4.2 $\text{DIM}(\mathcal{F}, P)$ can be executed in $O(t^4)$ time.

Proof To implement DIM , we maintain a set C which we initialize to be the empty set. We consider each $p \in \min(P)$ iteratively and proceed as follows. We reach all points $q \in p^\uparrow \setminus C$ from p in a depth first search over paths that are both monotonic and faithful (Definition 3.6). These paths form a directed tree Q rooted at p (where the arrows do not zigzag). We run the standard persistence algorithm on the filtration \mathcal{F} restricted to Q with a slight modification so that the branchings in the tree are taken care of. We traverse the tree Q in a depth first manner and each time we come to a branching node, we start with the matrix which was computed during the previous visit of this node. This means that we leave a copy of the reduced matrix at each node

that we traverse. Let t_p equal the maximum of the cardinality of $p^\uparrow \setminus C$ and the number of insertions of simplices over a monotonic path in the directed tree Q . Then, it is easy to see that $t_p = O(t)$ by our assumption on t (based on the description in Setup 3). At each node q we spend $O(t_p^2)$ time to copy a matrix of size $O(t_p) \times O(t_p)$. Also, for each simplex insertion, we spend $O(t_p^2)$ time to reduce this matrix. Therefore, in total, each simplex insertion takes $O(t_p^2) = O(t^2)$ time. Considering all minimal points, we see that, in total we insert only $O(t^2)$ simplices. This is because the input filtration consists of $O(t^2)$ insertions (each simplex in the complex K is inserted at most $O(t)$ times in the filtration) and each insertion is processed only once over all minimal points. This gives us a total complexity of $O(t^2) \times O(t^2) = O(t^4)$. \square

4.1 Detecting Interval Modules

We consider the following problem. Recall that P is a finite interval of \mathbf{Z}^2 .

Problem 4.3 Determine whether $M_{\mathcal{F}}$ is isomorphic to the direct sum of a certain number of copies of \mathbb{I}_P and if so, report the number of such copies.

Algorithm ISINTERVAL solves Problem 4.3. The correctness of the algorithm follows from Proposition 4.4. Below, for $m \in \mathbf{Z}_{\geq 0}$ let $\mathbb{I}_P^m := \underbrace{\mathbb{I}_P \oplus \mathbb{I}_P \oplus \cdots \oplus \mathbb{I}_P}_m$. In

particular, \mathbb{I}_P^0 is defined to be the trivial module. Let us recall that $(M_{\mathcal{F}})_{\partial P}$ denotes the zigzag module along the boundary cap ∂P (Definition 3.11).

Algorithm ISINTERVAL(\mathcal{F}, P)

- Step 1. Compute zigzag barcode $\text{barc}((M_{\mathcal{F}})_{\partial P})$ and let m be the multiplicity of the full interval.
- Step 2. Call $\text{DIM}(\mathcal{F}, P)$ (Computes $\dim(M_{\mathcal{F}})_p$ for every $p \in P$)
- Step 3. If $\dim(M_{\mathcal{F}})_p = m$ for each point $p \in P$ return m , otherwise return 0 indicating $M_{\mathcal{F}}$ has a summand which is not an interval module supported over P .

Recall that \mathbf{vec} denotes the category of finite dimensional vector spaces and linear maps.

Proposition 4.4 Assume that a given $M : P \rightarrow \mathbf{vec}$ has the indecomposable decomposition $M \cong \bigoplus_{i=1}^m M_i$. Then, every summand M_i is isomorphic to the interval module \mathbb{I}_P if and only if $\text{rk}_{\mathbb{I}}(M)(P) = \dim M_p = m$ for all $p \in P$.

Proof By Theorem 2.6, the forward direction is straightforward. Let us show the backward direction. Since $\text{rk}_{\mathbb{I}}(M)(P) = m$, by Theorem 2.6, we have that $M \cong \mathbb{I}_P^m \oplus M'$ where M' admits no summand that is isomorphic to \mathbb{I}_P . Fix any $p \in P$. Then, $m = \dim(M_p) = \dim((\mathbb{I}_P^m)_p) + \dim(M'_p) = m + \dim(M'_p)$, and hence $\dim(M'_p) = 0$. Since p was chosen arbitrarily, M' must be trivial. \square

Proposition 4.5 Algorithm ISINTERVAL can be run in $O(t^4)$ time.

Proof As we commented earlier, Step 1 computing the zigzag barcode $\text{barc}((M_{\mathcal{F}})_{\partial P})$ can be implemented to run in $O(t^\omega)$ time because it runs on a filtration comprising

$O(t)$ insertions and deletions of simplices over an index set of size $O(t)$. Step 2 takes $O(t^4)$ time (Proposition 4.2). Step 3 takes only time $O(t)$. This implies that the overall complexity is $O(t^4)$ given that $\omega < 2.373$. \square

4.2 Interval Decomposable Modules and Its Summands

Setup 3 still applies in Sect. 4.2. Next, we consider the problem of computing all indecomposable summands of $M_{\mathcal{F}}$ under the assumption that $M_{\mathcal{F}}$ is interval decomposable (Definition 2.3).

Problem 4.6 Assume that $M_{\mathcal{F}} : P \rightarrow \mathbf{vec}$ is interval decomposable. Find $\text{barc}(M_{\mathcal{F}})$.

We present algorithm INTERVAL to solve Problem 4.6 in $O(t^{\omega+2})$ time. This algorithm is eventually used to detect whether a given module is interval decomposable or not (Problem 4.16). Before describing INTERVAL, we first describe another algorithm TRUEINTERVAL. The outcomes of both INTERVAL and TRUEINTERVAL are the same as the barcode of $M_{\mathcal{F}}$ in Problem 4.6 (Propositions 4.10 and 4.14). Whereas TRUEINTERVAL is more intuitive, real implementation is accomplished via INTERVAL.

Definition 4.7 Let $\mathcal{I}(M_{\mathcal{F}}) := \{I \in \mathbf{Int}(P) : \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > 0\}$. We call $I \in \mathcal{I}(M_{\mathcal{F}})$ **maximal** if there is no $J \supsetneq I$ in $\mathbf{Int}(P)$ such that $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(J)$ is nonzero.

Proposition 4.8 Assume that $M_{\mathcal{F}}$ is interval decomposable and let $I \in \mathcal{I}(M_{\mathcal{F}})$ be maximal. Then, I belongs to $\text{barc}(M_{\mathcal{F}})$ and the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$ is equal to $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$.

Proof By assumption, all summands in the sum

$$\sum_{\substack{A \subseteq \text{nb}_{\mathbb{I}}(I) \cap P \\ A \neq \emptyset}} (-1)^{|A|} \text{rk}_{\mathbb{I}}(M_{\mathcal{F}}) \left(\overline{I \cup A} \right)$$

corresponding to the second term of (1) are zero. Hence, $\text{dgm}_{\mathbb{I}}(M_{\mathcal{F}})(I) = \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > 0$. Since $M_{\mathcal{F}}$ is interval decomposable, by Theorem 2.10, $\text{dgm}_{\mathbb{I}}(M_{\mathcal{F}})(I)$ is equal to the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$. Therefore, not only does I belong to $\text{barc}(M_{\mathcal{F}})$, but also the value $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$ is equal to the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$. \square

Corollary 4.9 Assume that $M_{\mathcal{F}}$ is interval decomposable and let $I \in \mathcal{I}(M_{\mathcal{F}})$ be maximal. Let $\mu_I := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$. Then, $M_{\mathcal{F}}$ admits a summand N which is isomorphic to $\mathbb{I}_I^{\mu_I}$.

Let us now describe a procedure TRUEINTERVAL that outputs all indecomposable summands of a given interval decomposable module. It uses ‘true’ quotient operation justifying the name. For computational efficiency, we will implement TRUEINTERVAL differently with the algorithm INTERVAL avoiding the quotient operation.

Let $M := M_{\mathcal{F}}$. First we compute $\dim M_p$ for every point $p \in P$. Iteratively, we choose a point p with $\dim M_p \neq 0$ and compute a maximal interval $I \in \mathcal{I}(M)$ containing p . Since M is interval decomposable, by Proposition 4.8 and Corollary 4.9 we

have that $I \in \text{barc}(M)$ and that there is a summand $N \cong \mathbb{I}_I^{\mu_I}$ of M . Consider the quotient module $M' := M/N$. Clearly, this ‘peeling off’ of N reduces the total dimension of the input module. Namely, $\dim M'_p = \begin{cases} \dim M_p - \mu_I, & p \in I \\ \dim M_p, & p \notin I. \end{cases}$ We continue the process by replacing M with M' until there is no point $p \in P$ with $\dim M_p \neq 0$ (note that M' is interval decomposable by Proposition 2.4). Since $\dim M := \sum_{p \in P} \dim M_p$ is finite, this process terminates in finitely many steps. By Proposition 2.4 and Corollary 4.9, the outcome of TRUEINTERVAL is a list of all intervals in $\text{barc}(M)$ with accurate multiplicities:

Proposition 4.10 *Assume that $M_{\mathcal{F}}$ is interval decomposable. Let $I_i, i = 1, \dots, k$ be the intervals computed by TRUEINTERVAL. For each $i = 1, \dots, k$, let $\mu_{I_i} := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I_i)$. Then, we have $M_{\mathcal{F}} \cong \bigoplus_{i=1}^k \mathbb{I}_{I_i}^{\mu_{I_i}}$.*

Next, we describe an algorithm INTERVAL that simulates TRUEINTERVAL while avoiding explicit quotienting of $M_{\mathcal{F}}$ by its summands.

We associate two variables $d(p)$ and $\text{ptlist}(p)$ with each point $p \in P$. Their roles are described below.

The variable $d(p)$ is a number that equals the original dimension of $(M_{\mathcal{F}})_p$ minus the number of intervals peeled off so far (counted with their multiplicities) which contained p . It is initialized to $\dim(M_{\mathcal{F}})_p$. Each time we compute a maximal interval $I \in \mathcal{I}(M_{\mathcal{F}})$ with multiplicity μ_I that contains p , we update $d(p) := d(p) - \mu_I$. This current value of $d(p)$ keeps track of how many more intervals that contain p still need to be peeled off by TRUEINTERVAL.

The variable $\text{ptlist}(p)$ is a list that maintains the set of the intervals containing p that have been output so far. So, if we have already output intervals, say I_1, \dots, I_k , which contain p , then $\text{ptlist}(p) = \{I_1, \dots, I_k\}$. While searching for a maximal interval I , we maintain a variable idlist for I that contains the set of intervals common to all points in I . So, if $I = \{p_1, \dots, p_m\}$, then $\text{idlist} = \text{ptlist}(p_1) \cap \dots \cap \text{ptlist}(p_m)$ at the end of the search. Initializing idlist with $\text{ptlist}(p)$ of the initial point p , we update it as we explore expanding I . Every time we augment I with a new point q , we update idlist by taking its intersection with $\text{ptlist}(q)$ associated with q .

We assume a routine COUNT that takes an idlist as input and gives the total number of intervals in the idlist counted with their multiplicities. This means that if $\text{idlist} = \{I_1, \dots, I_k\}$, then $\text{COUNT}(\text{idlist})$ returns the number $c := \mu_{I_1} + \dots + \mu_{I_k}$.

Notice that, while searching for a maximal interval starting from a point, we keep considering the original given module $M_{\mathcal{F}}$ since we do not implement the true ‘peeling’ (i.e. quotient $M_{\mathcal{F}}$ by a submodule). However, we modify the condition for checking the maximality of an interval I . We check whether $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > c$, that is, whether the generalized rank of $M_{\mathcal{F}}$ over I is larger than the total number of intervals containing I that would have been peeled off so far by TRUEINTERVAL. This idea is implemented in the following algorithm.

Algorithm INTERVAL (\mathcal{F} : Filtration, P : Poset)

- Step 1. Call $\text{DIM}(\mathcal{F}, P)$ and set $d(p) := \dim(M_{\mathcal{F}})_p$; $\text{ptlist}(p) := \emptyset$ for every $p \in P$
- Step 2. While there exists a $p \in P$ with $d(p) > 0$ do
 - Step 2.1 Let $I := \{p\}$; $\text{idlist} := \text{ptlist}(p)$; unmark every $q \in P$
 - Step 2.2 If there exists unmarked $q \in \text{nbdl}_{\mathbb{I}}(I)$ with $d(q) > 0$,⁶ then
 - i. $\text{templist} := \text{idlist} \cap \text{ptlist}(q)$; $c := \text{COUNT}(\text{templist})$
 - ii. If $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$ then⁷ mark q ; set $I := I \cup \{q\}$; $\text{idlist} := \text{idlist} \cap \text{ptlist}(q)$
 - iii. go to Step 2.2
 - Step 2.3 Output I with multiplicity $\mu_I := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) - c$
 - Step 2.4 For every $q \in I$ set $d(q) := d(q) - \mu_I$ and $\text{ptlist}(q) := \text{ptlist}(q) \cup \{I\}$

The output of INTERVAL can be succinctly described as:

Output: $\{(I_i, \mu_{I_i}) : i = 1, \dots, k\}$ where $I_i \in \mathbf{Int}(P)$ and μ_{I_i} is a positive integer for each i .

Remark 4.11 For each $p \in P$, we have that $\dim(M_{\mathcal{F}})_p = \sum_{I_i \ni p} \mu_{I_i}$. This equality holds even when $M_{\mathcal{F}}$ is not interval decomposable.

We will show that if $M_{\mathcal{F}}$ is interval decomposable, then the output of INTERVAL coincides with the barcode of $M_{\mathcal{F}}$ (Propositions 4.10 and 4.14).

Example 4.12 (INTERVAL with interval decomposable input) Suppose that $M_{\mathcal{F}} \cong \mathbb{I}_{I_1} \oplus \mathbb{I}_{I_2} \oplus \mathbb{I}_{I_3}$ as depicted in Fig. 4 (A). The algorithm INTERVAL yields $\{(I_1, 1), (I_2, 1), (I_3, 1)\}$. In particular, since $I_1 \supset I_2 \supset I_3$, INTERVAL outputs $(I_1, 1)$, $(I_2, 1)$, and $(I_3, 1)$ in order, as depicted in Fig. 5 (A).

Details for Example 4.12 We illustrate how the barcode of $M_{\mathcal{F}}$ is obtained as the output of INTERVAL. In Step 1, for $p = (2, 2)$, we have $d(p) = 3$ and $\text{ptlist}(p) = \emptyset$. For the first round of the while loop in Step 2, note that $q = (2, 1)$ belongs to $\text{nbdl}_{\mathbb{I}}(\{p\})$ and thus q can be added to $I = \{p\}$. After several rounds of the while loop, we obtain I_1 with $\mu_{I_1} = 1$, and for every $q \in I_1$ (including $p = (2, 2)$), $d(q)$ is decreased by 1. Next, suppose that another search begins at $I = \{p\}$ and it tries to include $q = (2, 1)$. We obtain $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) = 1$ as in the previous round, but now $\text{COUNT}(\text{ptlist}(p) \cap \text{ptlist}(q))$ also returns 1 in Step 2.2(i) because the previously detected interval I_1 contains both p and q . Then, the test $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > (c = 1)$ in Step 2.2(ii) fails and the search proceeds with other points. Again, after several iterations of successful and unsuccessful attempts to expand I , we obtain the interval

⁶ When the input $M_{\mathcal{F}}$ is interval decomposable, the algorithm outputs the barcode of $M_{\mathcal{F}}$ even without the condition “with $d(q) > 0$ ”. But, when $M_{\mathcal{F}}$ is *not* interval decomposable, the condition “with $d(q) > 0$ ” ensures that the equality in Remark 4.11 holds.

⁷ to check $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$, we invoke Theorem 3.12 and run the zigzag persistence algorithm described beneath Remark 3.13. For efficiency, one can use zigzag update algorithm in [9] instead of computing zigzag persistence afresh every time.

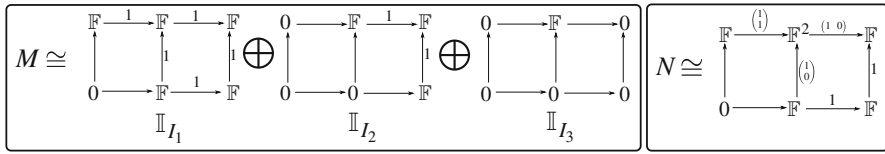


Fig. 4 Modules $M, N : \{1, 2, 3\} \times \{1, 2\} \rightarrow \mathbf{vec}$. M is interval decomposable, but N is not

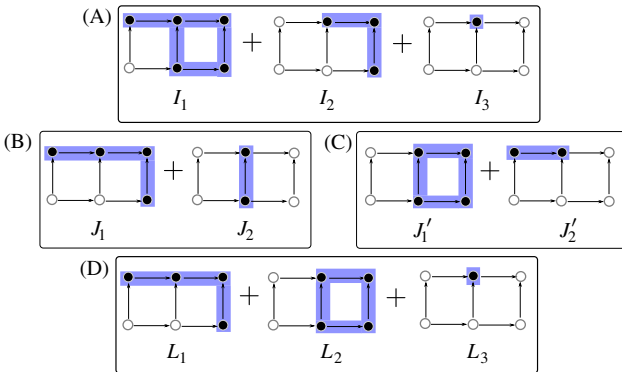


Fig. 5 An illustration for Examples 4.12 and 4.13

I_2 with $\mu_{I_2} = 1$. After obtaining (I_2, μ_2) , only $p = (2, 2)$ has $d(p) = 1$ because of which another round of while loop starting at p outputs the interval $I_3 = \{p\}$. \square

The algorithm INTERVAL outputs the barcode of an input module $M_{\mathcal{F}}$ when $M_{\mathcal{F}}$ is interval decomposable. If $M_{\mathcal{F}}$ is not interval decomposable, the output of the algorithm INTERVAL may not be unique and may not include some interval I even if the interval module \mathbb{I}_I is a summand of $M_{\mathcal{F}}$.

Example 4.13 (INTERVAL with non-interval-decomposable input) Consider the persistence module N depicted in Fig. 4, which is indecomposable and not isomorphic to an interval module.

- (i) When $M_{\mathcal{F}} \cong N$ is the input to INTERVAL, two possible final outputs of INTERVAL are $\{(J_1, 1), (J_2, 1)\}$ and $\{(J'_1, 1), (J'_2, 1)\}$ depicted in Fig. 5 (B) and (C).
- (ii) Let $M_{\mathcal{F}} \cong N \oplus \mathbb{I}_{I_2}$ be the input to INTERVAL, where \mathbb{I}_{I_2} is depicted in Fig. 4 (A). Then, one possible final output of INTERVAL is $\{(L_1, 1), (L_2, 1), (L_3, 1)\}$ as depicted in Fig. 5 (D). Note that the output does not contain interval I_2 , even though \mathbb{I}_{I_2} is a summand of $M_{\mathcal{F}}$.

Details for Example 4.13(ii) In Step 1 of INTERVAL, for $p = (2, 2)$, we have $d(p) = 3$ and $\text{ptlist}(p) = \emptyset$. In Step 2.2, $q = (1, 2)$ belongs to $\text{nbdl}_{\mathbb{I}}(\{p\})$ and thus q can be added to $I = \{p\}$. Once I becomes $\{p, q\}$, after multiple iterations within Step 2.2, I will expand to L_1 in Fig. 5 (D). Since $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(L_1) = 1$ and L_1 is a maximal interval (Definition 4.7), the pair of L_1 and $\mu_{L_1} = 1$ will be a part of the output. By continuing this process, one possible final output is $\{(L_1, 1), (L_2, 1), (L_3, 1)\}$ as depicted in Fig. 5 (D). \square

Proposition 4.14 *If $M_{\mathcal{F}}$ is interval decomposable, $\text{INTERVAL}(\mathcal{F}, P)$ computes all and only intervals in $\text{barc}(M_{\mathcal{F}})$ with correct multiplicities.*

Proof We prove that INTERVAL computes an interval I with multiplicity μ_I if and only if $\text{TRUEINTERVAL}(\mathcal{F}, P)$ computes it with the same multiplicity.

(‘if’): We induct on the list of intervals in the order they are computed by TRUEINTERVAL . We prove two claims by induction: (i) TRUEINTERVAL can be run to explore the points in P in the same order as INTERVAL while searching for maximal intervals, (ii) if $I_i, i = 1, \dots, k$, are the intervals computed by TRUEINTERVAL with this chosen order, then INTERVAL also outputs these intervals with the same multiplicities. Clearly, for $i = 1$, INTERVAL computes the maximal interval on the same input module $M_{\mathcal{F}}$ as TRUEINTERVAL does. So, clearly, TRUEINTERVAL can be made to explore P as INTERVAL does and hence their outputs are the same. Assume inductively that the hypotheses hold for $i \geq 1$. Then, TRUEINTERVAL operates next on the module $M_{i+1} := M_{\mathcal{F}} / (\mathbb{I}_{I_1}^{\mu_{I_1}} \oplus \dots \oplus \mathbb{I}_{I_i}^{\mu_{I_i}})$ (here each $\mathbb{I}_{I_i}^{\mu_{I_i}}$ stands for a summand of $M_{\mathcal{F}}$ that is isomorphic to $\mathbb{I}_{I_i}^{\mu_{I_i}}$ by Corollary 4.9). We let TRUEINTERVAL explore P in the same way as INTERVAL does. This is always possible because the outcome of the test for exploration remains the same in both cases as we argue. The variable $d(p)$ at this point has the value $\dim(M_{i+1})_p$ and thus both TRUEINTERVAL and INTERVAL can start exploring from the point p if $d(p) > 0$. So, we let TRUEINTERVAL compute the next maximal interval I_{i+1} starting from the point p if INTERVAL starts from p .

Now, when INTERVAL tests for a point q to expand the interval I , we claim that the result would be the same if TRUEINTERVAL tested for q . First of all, the condition whether $I \cup \{q\}$ is an interval or not does not depend on which algorithm we are executing. Second, the list supplied to COUNT in Step 2.2 (i) exactly equals the list of intervals containing $I \cup \{q\}$ that INTERVAL has already output. By the inductive hypothesis, this list is exactly equal to the list of intervals that TRUEINTERVAL had already ‘peeled off’. Therefore, the test $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$ that INTERVAL performs in Step 2.2 (ii) is exactly the same as the test $\text{rk}_{\mathbb{I}}(M_{i+1})(I \cup \{q\}) > 0$ that TRUEINTERVAL would have performed for the module M_{i+1} . This establishes that INTERVAL computes the same interval I_{i+1} with the same multiplicity as TRUEINTERVAL would have computed on M_{i+1} using the same order of exploration as the inductive hypothesis claims.

(‘only if’): We already know that INTERVAL computes all intervals that TRUEINTERVAL computes. We claim that it does not compute any other interval. For intervals computed by TRUEINTERVAL , one has that $\dim(M_{\mathcal{F}})_p = \sum_i (\mathbb{I}_{I_i}^{\mu_{I_i}})_p$, or equivalently $\dim(M_{\mathcal{F}})_p = \sum_{I \in \{I_1, \dots, I_k\}, s.t. I \ni p} \mu_I$. The algorithm INTERVAL decreases the variable $d(p)$ exactly by the amount on the right-hand side of the equation and is initialized to $\dim(M_{\mathcal{F}})_p$. Therefore, every $d(p)$ becomes equal to $\dim(M_{\mathcal{F}})_p$ after INTERVAL computes the intervals that TRUEINTERVAL computes. The condition $d(p) > 0$ in the *while* loop prohibits INTERVAL to compute any other interval. \square

Proposition 4.15 $\text{INTERVAL}(\mathcal{F}, P)$ runs in $O(t^{\omega+2})$ time.

Proof Each iteration in the *while* loop executes a traversal of the graph underlying the poset P starting from a point p . Each time, we reach a new point q in this traversal, we execute a zigzag persistence computation on the boundary cap $\partial(I \cup \{q\})$. This

means, the number of times a zigzag persistence is computed equals the number of times a point in the poset is considered by the *while* loop. We claim that this number is $O(t^2)$. Each time a point p is considered by the *while* loop, either we include it in an interval that is output (a successful attempt), or we don't include it in the expansion of the current interval (unsuccessful attempt) and q appears as a point in the neighborhood of an interval that is output. The number of times a point is involved in a successful attempt is at most t because a point can be contained in at most t intervals ($\dim(M_{\mathcal{F}})_p \leq t$). Similarly, the number of times a point is involved in an unsuccessful attempt is at most $4t$ because the point can be in the neighborhood of at most $4t$ intervals (at most t intervals for each of its ≤ 4 neighbors). Therefore, each point participates in at most $O(t)$ computations of zigzag persistence over the entire *while* loop. Each zigzag computation takes time $O(t^\omega)$ since the filtration $\mathcal{F}|_{\partial(t \cup \{q\})}$ restricted to the boundary cap has length at most t comprising at most t insertions and deletions of simplices. It follows that the total cost due to zigzag persistence computation is bounded by $O(t^{\omega+2})$.

Now, we analyze the cost of maintaining the lists with each point and with the intervals under construction. Notice that $\dim(M_{\mathcal{F}}) = \sum_{p \in P} \dim(M_{\mathcal{F}})_p = O(t^2)$ because there are at most t points in P with $\dim(M_{\mathcal{F}})_p \leq t$ for each $p \in P$ since \mathcal{F} has at most t simplices. Each *while* loop iteration maintains a global list, calls COUNT on this list, and updates $\text{ptlist}(q)$ for some points $q \in P$. The cost of this counting and updates cannot be more than the order of the final total size $\sum_p \text{ptlist}(p)$ of the lists, which in turn is no more than $\dim(M_{\mathcal{F}}) = O(t^2)$. Thus, over the entire *while* loop we incur $O(t^4)$ cost for maintenance of the lists and for the counting based on them. Also, step 1 takes $O(t^4)$ time to compute the dimensions. Thus, we have a worst-case complexity of $O(t^4 + t^{\omega+2}) = O(t^{\omega+2})$ because it is known that $\omega \geq 2$. \square

4.3 Interval Decomposability

Setup 3 still applies in Sect. 4.3. We consider the following problem.

Problem 4.16 Determine whether the module $M_{\mathcal{F}}$ is interval decomposable or not.

If the input module $M_{\mathcal{F}}$ is interval decomposable, then the algorithm INTERVAL computes all intervals in the barcode. However, if the module $M_{\mathcal{F}}$ is not interval decomposable, then the algorithm is not guaranteed to output all interval summands. We show that INTERVAL still can be used to solve Problem 4.16. For this we test whether each of the output intervals I with multiplicity μ_I indeed supports a summand $N \cong \mathbb{I}_I^{\mu_I}$ of $M_{\mathcal{F}}$.

To do this we run Algorithm 3 in Asashiba et al. [26] for each of the output intervals of INTERVAL. Call this algorithm TESTINTERVAL which with an input interval I , returns $\mu_I > 0$ if the module $\mathbb{I}_I^{\mu_I}$ is a summand of M and 0 otherwise.

For each of the intervals I with multiplicity μ_I returned by INTERVAL(\mathcal{F} , P) we test whether TESTINTERVAL(I) returns a non-zero μ_I . The first time the test fails, we declare that $M_{\mathcal{F}}$ is not interval decomposable. This gives us a polynomial time algorithm (with complexity $O(t^{3\omega+2})$) to test whether a module induced by a given

bifiltration is interval decomposable or not. It is a substantial improvement over the result of Asashiba et al. [26] who gave an algorithm for tackling the same problem. Their algorithm cleverly enumerates the intervals in the poset to test, but still tests exponentially many of them and hence may run in time that is exponential in t . Because of our algorithm INTERVAL, we can do the same test but only on polynomially many intervals.

Algorithm ISINTERVALDECOMP (\mathcal{F}, P)

- Step 1. $\mathcal{I} = \{(I_i, \mu_{I_i})\} \leftarrow \text{INTERVAL}(\mathcal{F}, P)$
- Step 2. For every $I_i \in \mathcal{I}$ do
 - Step 2.1 $\mu \leftarrow \text{TESTINTERVAL}(M_{\mathcal{F}}, I_i)$
 - Step 2.2 If $\mu \neq \mu_{I_i}$ then output false; quit
- Step 3. output true

Proposition 4.17 ISINTERVALDECOMP(\mathcal{F}, P)

1. returns true if and only if $M_{\mathcal{F}}$ is interval decomposable, and
2. takes $O(t^{3\omega+2})$ time.

Proof By the contrapositive of Proposition 4.10, if for any of the computed interval(s) $I_i, i = 1, \dots, k$ by INTERVAL, $\mathbb{I}_{I_i}^{\mu_{I_i}}$ is not a summand of $M_{\mathcal{F}}$, then $M_{\mathcal{F}}$ is not interval decomposable. On the other hand, if every such interval module is a summand of $M_{\mathcal{F}}$, then we have that $M_{\mathcal{F}} \cong \bigoplus_{i=1}^k \mathbb{I}_{I_i}^{\mu_{I_i}}$ because $\dim(M_{\mathcal{F}})_p = \sum_i^k \dim(\mathbb{I}_{I_i}^{\mu_{I_i}})_p$ for every $p \in P$.

Time complexity By Proposition 4.15, Step 1 runs in time $O(t^{\omega+2})$. We claim that $\dim(M_{\mathcal{F}}) = O(t^2)$ (see Proof of Proposition 4.15). Therefore, INTERVAL returns at most $O(t^2)$ intervals. According to the analysis in Asashiba et al. [26], each test in Step 2.1 takes $O(((\dim M_{\mathcal{F}})^{\omega} + t)t^{\omega}) = O(t^{3\omega})$ time and thus $O(t^{3\omega+2})$ in total over all $O(t^2)$ tests which dominates the time complexity of ISINTERVALDECOMP. \square

4.4 INTERVAL Produces Partial Sections of Indecomposable Summands

The algorithm INTERVAL produces all intervals of an input interval decomposable module. A natural question is what does the algorithm INTERVAL return on a module that is not interval decomposable (Fig. 5 (B)). We show that the intervals returned by the algorithm support “partial” sections of indecomposable summands:

Proposition 4.18 Let \mathcal{I} be the set of intervals computed by INTERVAL(\mathcal{F}, P). Then, for every $I \in \mathcal{I}$, there exists a section supported over I of an indecomposable summand of $M_{\mathcal{F}}$.

The above result follows from the proposition below since INTERVAL outputs an interval I only if $\text{rk}(M_{\mathcal{F}})(I) > 0$.

Proposition 4.19 Let P be a finite connected poset. Let M be a P -module with an indecomposable decomposition $M \cong \bigoplus_{j \in L} M_j$ for some finite set L . Let $I \in \text{Int}(P)$.

If $\text{rk}(M)(I) = c > 0$, there are $j_1, \dots, j_c \in L$ such that for each $t = 1, \dots, c$, there exists a section \mathbf{v} of $M_{j_t}|_I$ that is fully supported, i.e. $\mathbf{v}_p \neq 0$ for all $p \in I$.⁸

Proof By the assumption, we have

$$M|_I \cong \left(\bigoplus_{j \in L} M_j \right) \Big|_I \cong \bigoplus_{j \in L} M_j|_I.$$

Note that $M_j|_I$ can be decomposable for each j . Let $M_j|_I$ have an indecomposable decomposition

$$M_j|_I \cong \bigoplus_{k \in K_j} M_{jk},$$

i.e. each M_{jk} is an indecomposable I -module. Then $M|_I$ has an indecomposable decomposition

$$M|_I \cong \bigoplus_{j \in L} \bigoplus_{k \in K_j} M_{jk}.$$

Since I is a finite poset and M_p is finite dimensional for each $p \in P$, we have that $\varprojlim M|_I$ is finite dimensional. Hence the notions of direct product and direct sum coincide in the category of I -indexed modules. This implies that $\varprojlim(N_1 \oplus N_2) \cong \varprojlim N_1 \oplus \varprojlim N_2$ (and it is a standard fact that $\varinjlim(N_1 \oplus N_2) \cong \varinjlim N_1 \oplus \varinjlim N_2$ [38, Thm. V.5.1]). Therefore, we have that

$$c = \text{rk}(M)(I) = \text{rank}(M|_I) = \sum_{j \in L} \sum_{k \in K_j} \text{rank}(M_{jk}).$$

Since each M_{jk} is indecomposable, by Theorem 2.6,

$$\text{rank}(M_{jk}) = \begin{cases} 1, & \text{if } M_{jk} \cong \mathbb{I}_I \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, there exist exactly c distinct pairs (j, k) such that $M_{jk} \cong \mathbb{I}_I$. Hence, for each of such (j, k) , we can find a section $\mathbf{v} := \mathbf{v}^{jk}$ of M_{jk} that is fully supported over I . Since M_{jk} is a summand of $M_j|_I$, \mathbf{v} is also a section of $M_j|_I$, as desired. \square

4.5 Barcode Ensemble from INTERVAL

Fix a P -module $M = M_{\mathcal{F}}$ as an input to the algorithm INTERVAL. Recall that an output of INTERVAL with the input M is a collection $\{(I_i, \mu_{I_i})\}_i$. This collection may

⁸ We remark that these sections can be further extended using the structure maps to submodules which may have larger supports than the original sections.

change with different choices available during the exploration for computing maximal intervals. The collection $\mathbf{E}(M)$ of all possible outputs of INTERVAL will be called the **barcode ensemble** of M . Proposition 4.14 implies:

Corollary 4.20 *If $\mathbf{E}(M)$ contains more than one collection, then M is not interval decomposable.*

Suppose that a collection $\mathcal{C} := \{(I_i, \mu_{I_i})\}_i$ belongs to $\mathbf{E}(M)$. We consider the interval decomposable module $\mathbb{I}_{\mathcal{C}} := \bigoplus_i \mathbb{I}_{I_i}^{\mu_{I_i}}$ corresponding to \mathcal{C} . Fix any interval $J \subseteq P$. We define

$$\mathbf{E}(M)(J) := \max_{\mathcal{C} \in \mathbf{E}(M)} \text{rk}(\mathbb{I}_{\mathcal{C}})(J).$$

We remark that $\text{rk}(\mathbb{I}_{\mathcal{C}})(J)$ is equal to $\sum_i \mu_i$ where the sum is taken over all i such that $I_i \supseteq J$.

A single output of the algorithm INTERVAL may fail to capture the generalized rank invariant of M ; for instance, in Example 4.13 (i), the generalized rank invariant of the module $M_{\mathcal{F}} (\cong N)$ is different from the generalized rank invariants of $\bigoplus_{i=1}^2 \mathbb{I}_{J_i}$ or $\bigoplus_{i=1}^2 \mathbb{I}_{J'_i}$. Nevertheless, the barcode ensemble $\mathbf{E}(M)$ recovers the generalized rank invariant of the input module M :

Proposition 4.21 *For every interval $J \subseteq P$, we have*

$$\text{rk}(M)(J) = \mathbf{E}(M)(J).$$

By the design of the algorithm INTERVAL, the above equality is clearly true when $\text{rk}(M)(J) = 0$.

Proof (\geq) Let $m := \text{rk}(M)(J)$ and pick any $\mathcal{C} = \{(I_i, \mu_i)\}_i$ in $\mathbf{E}(M)$. By the design of the algorithm INTERVAL, it is not possible for the sum $\sum_{I_i \supseteq J} \mu_i (= \text{rk}(\mathbb{I}_{\mathcal{C}}))$ to be greater

than m . Since \mathcal{C} was arbitrarily chosen in $\mathbf{E}(M)$, we have $\text{rk}(M)(J) \geq \mathbf{E}(M)(J)$.

(\leq) Assume that $m := \mathbf{E}(M)(J)$. This implies that there exists $\mathcal{C} = \{(I_i, \mu_i)\}_i$ in $\mathbf{E}(M)$ such that $\text{rk}(\mathbb{I}_{\mathcal{C}})(J) = m$. We prove the desired inequality by contradiction. Suppose that $n := \text{rk}(M)(J) > m$. This implies that we can run the while loop (Step 2) at least n times starting from the singleton interval $I = \{p\}$ for any point $p \in J$. Let \mathcal{C}' be any final output obtained in this strategy. Then, the output \mathcal{C}' contains intervals I'_j containing J with multiplicity μ'_j and we have $n = \sum_j \mu'_j$. Then, by Theorem 2.6, we have that $\text{rk}(\mathbb{I}_{\mathcal{C}'})(J) \geq n > m$, contradicting the definition of $\mathbf{E}(M)(J)$. \square

5 Discussion

Some open questions that follow are: (i) Can we generalize Theorem 3.12 to d -parameter persistent homology for $d > 2$? One obstacle is that, for $d > 2$, the (minimal) lower and upper fences of an interval $I \in \mathbf{Int}(\mathbf{Z}^d)$ are generally not zigzag

posets. (ii) Can the complexity of the algorithms be improved? (iii) In particular, can we improve the interval decomposability testing by improving TESTINTERVAL which currently uses an algorithm of Asashiba et al.? (iv) Finally, it would be interesting to see the use of our results in applications, see e.g. [41] for an application in machine learning.

Acknowledgements The authors thank the anonymous reviewers for constructive feedback and suggesting ideas that shortened the proof of Theorem 3.12. This work is supported by NSF Grants CCF-2049010, CCF-1740761, DMS-1547357, and IIS-1901360.

Funding This work is supported by NSF grants CCF-2049010, CCF-1740761, DMS-1547357, and IIS-1901360.

Data Availability Data sharing not applicable to this article.

Code Availability Code sharing not applicable to this article.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Limits and Colimits

We recall the notions of limit and colimit from category theory [38]. Recall that a poset P can be viewed as a category whose objects are the elements of P and morphisms are the comparable pairs $p \leq q$ in P . Although limits and colimits are defined for functors indexed by small categories we restrict our attention to poset-indexed functors. Let \mathcal{C} be any category.

Definition A.1 (Cone) Let $F : P \rightarrow \mathcal{C}$ be a functor. A *cone* over F is a pair $(L, (\pi_p)_{p \in P})$ consisting of an object L in \mathcal{C} and a collection $(\pi_p)_{p \in P}$ of morphisms $\pi_p : L \rightarrow F(p)$ that commute with the arrows in the diagram of F , i.e. if $p \leq q$ in P , then $\pi_q = F(p \leq q) \circ \pi_p$ in \mathcal{C} , i.e. the diagram below commutes.

$$\begin{array}{ccc}
 F(p) & \xrightarrow{F(p \leq q)} & F(q) \\
 & \swarrow \pi_p & \searrow \pi_q \\
 & L &
 \end{array} \tag{11}$$

In Definition A.1, the cone $(L, (\pi_p)_{p \in P})$ over F is sometimes denoted simply by L , suppressing the collection $(\pi_p)_{p \in P}$ of morphisms if no confusion can arise. A limit of $F : P \rightarrow \mathcal{C}$ is a terminal object in the collection of all cones over F :

Definition A.2 (Limit) Let $F : P \rightarrow \mathcal{C}$ be a functor. A *limit* of F is a cone over F , denoted by $(\varprojlim F, (\pi_p)_{p \in P})$ or simply $\varprojlim F$, with the following *terminal* property: If there is another cone $(L', (\pi'_p)_{p \in P})$ of F , then there is a *unique* morphism $u : L' \rightarrow \varprojlim F$ such that $\pi'_p = \pi_p \circ u$ for all $p \in P$.

It is possible that a functor does not have a limit at all. However, if a functor does have a limit then the terminal property of the limit guarantees its uniqueness up to isomorphism. For this reason, we sometimes refer to a limit as *the* limit of a functor.

Cocones and colimits are defined in a dual manner:

Definition A.3 (Cocone) Let $F : P \rightarrow \mathcal{C}$ be a functor. A *cocone* over F is a pair $(C, (i_p)_{p \in P})$ consisting of an object C in \mathcal{C} and a collection $(i_p)_{p \in P}$ of morphisms $i_p : F(p) \rightarrow C$ that commute with the arrows in the diagram of F , i.e. if $p \leq q$ in P , then $i_p = i_q \circ F(p \leq q)$ in \mathcal{C} , i.e. the diagram below commutes.

$$\begin{array}{ccc}
 & C & \\
 i_p \nearrow & & \nwarrow i_q \\
 F(p) & \xrightarrow{F(p \leq q)} & F(q)
 \end{array} \tag{12}$$

In Definition A.3, a cocone $(C, (i_p)_{p \in P})$ over F is sometimes denoted simply by C , suppressing the collection $(i_p)_{p \in P}$ of morphisms. A colimit of $F : P \rightarrow \mathcal{C}$ is an initial object in the collection of cocones over F :

Definition A.4 (Colimit) Let $F : P \rightarrow \mathcal{C}$ be a functor. A *colimit* of F is a cocone, denoted by $(\varinjlim F, (i_p)_{p \in P})$ or simply $\varinjlim F$, with the following *initial* property: If there is another cocone $(C', (i'_p)_{p \in P})$ of F , then there is a *unique* morphism $u : \varinjlim F \rightarrow C'$ such that $i'_p = u \circ i_p$ for all $p \in P$.

It is possible that a functor does not have a colimit at all. However, if a functor does have a colimit then the initial property of the colimit guarantees its uniqueness up to isomorphism. For this reason, we sometimes refer to a colimit as *the* colimit of a functor. It is well-known that if P is finite, then any functor $F : P \rightarrow \mathbf{vec}$ admits both limit and colimit in \mathbf{vec} . Assume that P is also connected. Then, by the commutativity in (11) and (12), once a cone and a cocone of F are specified, there exists the canonical map from the cone to the cocone, leading to Definition 2.5.

Remark A.5 Let P be a finite and connected poset. Let Q be a finite and connected subposet of P . Let us fix any $F : P \rightarrow \mathbf{vec}$.

- (i) For any cone $(L', (\pi'_p)_{p \in P})$ over F , its restriction $(L', (\pi'_p)_{p \in Q})$ is a cone over the restriction $F|_Q : Q \rightarrow \mathbf{vec}$. Therefore, by the terminal property of the limit

- $(\varprojlim F|_Q, (\pi_q)_{q \in Q})$, there exists the unique morphism $u : L' \rightarrow \varprojlim F|_Q$ such that $\pi'_q = \pi_q \circ u$ for all $q \in Q$.
- (ii) For any cocone $(C', (i'_p)_{p \in P})$ over F , its restriction $(C', (i'_p)_{p \in Q})$ is a cocone over the restriction $F|_Q : Q \rightarrow \mathbf{vec}$. Therefore, by the initial property of $\varprojlim F|_Q$, there exists the unique morphism $u : \varprojlim F|_Q \rightarrow C'$ such that $i'_q = u \circ i_q$ for all $q \in Q$.
- (iii) By the previous two items, there exist linear maps $\pi : \varprojlim F \rightarrow \varprojlim F|_Q$ and $\iota : \varprojlim F|_Q \rightarrow \varprojlim F$ such that $\psi_F = \iota \circ \psi_{F|_Q} \circ \pi$.

Therefore, $\text{rank}(F) = \text{rank}(\psi_F) \leq \text{rank}(\psi_{F|_Q}) = \text{rank}(F|_Q)$.

References

- Gabriel, P.: Unzerlegbare darstellungen i. *Manuscr. Math.* **6**, 71–103 (1972)
- Crawley-Boevey, W.: Decomposition of pointwise finite-dimensional persistence modules. *J. Algebra Appl.* **14**(05), 1550066 (2015)
- Dey, T.K., Wang, Y.: *Computational Topology for Data Analysis*. Cambridge University Press, Cambridge, UK (2022)
- Edelsbrunner, H., Harer, J.: *Computational Topology: An Introduction*. American Mathematical Society, Providence, RI (2010)
- Bauer, U., Botnan, M.B., Oppermann, S., Steen, J.: Cotorsion torsion triples and the representation theory of filtered hierarchical clustering. *Adva. Math.* **369**, 107171 (2020)
- Carlsson, G., de Silva, V.: Zigzag persistence. *Found. Comput. Math.* **10**(4), 367–405 (2010)
- Carlsson, G., Mémoli, F.: Multiparameter hierarchical clustering methods. In: *Classification as a Tool for Research*, pp. 63–70. Springer, Heidelberg (2010)
- Carlsson, G., Zomorodian, A.: The theory of multidimensional persistence. *Discrete Comput. Geom.* **42**(1), 71–93 (2009)
- Dey, T.K., Hou, T.: Updating zigzag persistence and maintaining representatives over changing filtrations. *CoRR arXiv:2112.02352* (2021)
- Escolar, E.G., Hiraoka, Y.: Persistence modules on commutative ladders of finite type. *Discrete Comput. Geom.* **55**(1), 100–157 (2016)
- Kim, W., Mémoli, F.: Spatiotemporal persistent homology for dynamic metric spaces. *Discrete Comput. Geom.* **66**(3), 831–875 (2021)
- Lesnick, M.: *Multidimensional interleavings and applications to topological inference*. PhD thesis, Stanford University (2012)
- Lesnick, M.: The theory of the interleaving distance on multidimensional persistence modules. *Found. Comput. Math.* **15**(3), 613–650 (2015)
- Miller, E.: Modules over posets: commutative and homological algebra. *arXiv preprint arXiv:1908.09750* (2019)
- Biasotti, S., Cerri, A., Frosini, P., Giorgi, D., Landi, C.: Multidimensional size functions for shape comparison. *J. Math. Imaging Vis.* **32**(2), 161–179 (2008)
- Lesnick, M., Wright, M.: Interactive visualization of 2-d persistence modules. *arXiv preprint arXiv:1512.00180* (2015)
- Cai, C., Kim, W., Mémoli, F., Wang, Y.: Elder-rule-staircodes for augmented metric spaces. *SIAM J. Appl. Algebra Geom.* **5**(3), 417–454 (2021)
- Botnan, M., Lebovici, V., Oudot, S.: On Rectangle-Decomposable 2-Parameter Persistence Modules. In: *36th International Symposium on Computational Geometry (SoCG 2020)*, vol. 164 (2020). <https://doi.org/10.4230/LIPIcs.SoCG.2020.22>
- Cochoy, J., Oudot, S.: Decomposition of exact pfd persistence bimodules. *Discrete Comput. Geom.* **63**(2), 255–293 (2020)
- Asashiba, H., Escolar, E.G., Nakashima, K., Yoshiwaki, M.: On approximation of $2d$ -persistence modules by interval-decomposables. *arXiv preprint arXiv:1911.01637* (2019)

21. Clause, N., Kim, W., Memoli, F.: The discriminating power of the generalized rank invariant. arXiv preprint [arXiv:2207.11591](https://arxiv.org/abs/2207.11591) (2022)
22. Kim, W., Mémoli, F.: Generalized persistence diagrams for persistence modules over posets. *J. Appl. Comput. Topol.* **5**(4), 533–581 (2021)
23. Patel, A.: Generalized persistence diagrams. *J. Appl. Comput. Topol.* **1**(3), 397–419 (2018)
24. Kim, W., Mémoli, F.: Rank invariant for zigzag modules. arXiv preprint [arXiv:1810.11517v1](https://arxiv.org/abs/1810.11517v1) (2018)
25. Chambers, E., Letscher, D.: Persistent homology over directed acyclic graphs. In: *Research in Computational Topology*, pp. 11–32. Springer, (2018)
26. Asashiba, H., Buchet, M., Escolar, E.G., Nakashima, K., Yoshiwaki, M.: On interval decomposability of 2d persistence modules. *Comput. Geom.* **105**, 101879 (2022)
27. Dey, T.K., Xin, C.: Generalized persistence algorithm for decomposing multiparameter persistence modules. *J. Appl. Comput. Topol.* **9**, 1–52 (2022)
28. Kerber, M.: Multi-parameter persistent homology is practical. In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond* (2020)
29. Bethausser, L., Bubenik, P., Edwards, P.B.: Graded persistence diagrams and persistence landscapes. *Discrete Comput. Geom.* **8**, 1–28 (2021)
30. Botnan, M., Oppermann, S., Oudot, S.: Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions. arXiv preprint [arXiv:2107.06800](https://arxiv.org/abs/2107.06800) (2021)
31. Bubenik, P., Elchesen, A.: Virtual persistence diagrams, signed measures, and wasserstein distance. arXiv preprint [arXiv:2012.10514](https://arxiv.org/abs/2012.10514) (2020)
32. Kim, W., Moore, S.: Bigraded betti numbers and generalized persistence diagrams. arXiv preprint [arXiv:2111.02551v3](https://arxiv.org/abs/2111.02551v3) (2021)
33. McCleary, A., Patel, A.: Edit distance and persistence diagrams over lattices. arXiv preprint [arXiv:2010.07337](https://arxiv.org/abs/2010.07337) (2020)
34. Botnan, M., Lesnick, M.: Algebraic stability of zigzag persistence modules. *Algebraic Geometric Topol.* **18**(6), 3133–3204 (2018)
35. Azumaya, G.: Corrections and supplementaries to my paper concerning Krull–Remak–Schmidt’s theorem. *Nagoya Math. J.* **1**, 117–124 (1950)
36. Rota, G.C.: On the foundations of combinatorial theory i. theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **2**(4), 340–368 (1964)
37. Stanley, R.P.: *Enumerative Combinatorics*. Cambridge Studies in Advanced Mathematics, vol. 1, 2nd edn. Cambridge University Press, Cambridge (2011)
38. Mac Lane, S.: *Categories for the Working Mathematician*, vol. 5. Springer, New York (2013)
39. Milosavljević, N., Morozov, D., Skraba, P.: Zigzag persistent homology in matrix multiplication time. In: *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, pp. 216–225 (2011)
40. Dey, T.K., Hou, T.: Fast computation of zigzag persistence. In: *30th Annual European Symposium on Algorithms, ESA 2022. LIPIcs*, vol. 244 (2022). <https://doi.org/10.4230/LIPIcs.ESA.2022.43>
41. Xin, C., Mukherjee, S., Samaga, S., Dey, T.K.: GRIL: a 2-parameter persistence based vectorization for machine learning. *CoRR* [arXiv:2304.04970](https://arxiv.org/abs/2304.04970) (2023)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.