Efficient Approximation of Multiparameter Persistence Modules

David Loiseaux, Mathieu Carriere, Andrew Blumberg

Efficient Approximation of Multiparameter Persistence Modules

David Loiseaux, Mathieu Carrière, Andrew J. Blumberg

June 7, 2022

Abstract

Topological Data Analysis is a growing area of data science, which aims at computing and characterizing the geometry and topology of data sets, in order to produce useful descriptors for subsequent statistical and machine learning tasks. Its main computational tool is *persistent homology*, which amounts to track the topological changes in growing families of subsets of the data set itself, called *filtrations*, and encode them in an algebraic object, called *persistence module*. Even though algorithms and theoretical properties of modules are now well-known in the single-parameter case, that is, when there is only one filtration to study, much less is known in the multi-parameter case, where several filtrations are given at once. Though more complicated, the resulting persistence modules are usually richer and encode more information, making them better descriptors for data science.

In this article, we present the first approximation scheme, which is based on *fibered barcodes* and *exact matchings*, two constructions that stem from the theory of single-parameter persistence, for computing and decomposing general multi-parameter persistence modules. Our algorithm has controlled complexity and running time, and works in arbitrary dimension, i.e., with an arbitrary number of filtrations. Moreover, when restricting to specific classes of multi-parameter persistence modules, namely the ones that can be decomposed into *intervals*, we establish theoretical results about the approximation error between our estimate and the true module in terms of interleaving distance. Finally, we present empirical evidence validating output quality and speed-up on several data sets.

1. Introduction

Topological Data Analysis (TDA) [Car09, EH10] is an area of data science that has been developing quite fast and that has gathered the interest of many practitioners in the last few years, due to its success in various applications. At its core is the use of computational tools from algebraic topology to capture multiscale shape information from data, that require only mild assumptions about the data (e.g., a metric or similarity measure between points) in order to be applied. Moreover, a centerpiece of the formal foundations of TDA are mathematical guarantees that ensure the resulting descriptors are reasonably efficient to compute and robust to perturbations. As such, TDA has been applied successfully in a wide range of scientific fields, including bioinformatics, computer graphics, and machine learning, among others.

Persistent homology. The main computational tool of TDA is *persistent homology* (PH). Whereas homology is a descriptor of a topological space X, the core idea of PH is to study how the homology groups change when computed on a specific family of subspaces of X called a *filtration* of X. A filtration is a family $\mathcal{F}$ of subspaces of X indexed over a partially ordered set I: $\mathcal{F} = \{X_i \subseteq X\}_{i \in I}$, that is *nested w.r.t. inclusion*, i.e., it satisfies $X_i \subseteq X_j$ for any $i \leq j$. Then, the functoriality of homology and these inclusion induces morphisms between the corresponding homology groups $H_*(X_i) \to H_*(X_j)$ for each pair $i \leq j$, which allows to detect the differences in homology when going from index i to index j. One of the most common ways to produce such filtrations is to study the *sublevel sets* of a continuous *filter* function $f : X \to \mathbb{R}^n$, defined with $\mathcal{F} = \{x \in X \,:\, f(x) \leq u\}_{u \in \mathbb{R}^n}$; where the partial order on the poset $\mathbb{R}^n$ (denoted by $\leq$) is defined, for any $a, b \in \mathbb{R}^n$, as $a \leq b$ if and only if $a_i \leq b_i$ for any $1 \leq i \leq n$.

Single-parameter PH. When $\mathcal{I}$ is totally ordered, e.g., when $\mathcal{I} \subseteq \mathbb{R}$, then applying the homology functor $H_*(-; k)$ for a field k to a (single-parameter) filtration results in a sequence of vector spaces connected by linear transformations. This sequence is called a *single-parameter persistence module* and has been studied extensively in the TDA literature [Car09, CdGO16, EH10, Oud15]. Notably, one can show that such persistence modules can always be decomposed into a direct sum of simple summands, which intuitively represent the appearances (birth) and disappearances (death) of topological structures detected by homology as the index increases. Moreover, single-parameter persistence modules can be efficiently represented in a compact descriptor called the *persistence barcode*, and several vectorization methods, as well as kernels and machine learning classifiers, have been proposed for such barcodes in the literature [Bub15, AEK+17, RHBK15, CCO17, CCI+20]. As a consequence, most applications of TDA use single-parameter persistence modules, and often use the sublevel sets of, e.g., the data set scale, as the corresponding single-parameter filtration.

Multi-parameter PH. However, many data sets come with not just one, but multiple, possibly intertwined, salient filtrations. For example, image data typically has both a spatial filtration and an intensity filtration. Arbitrary point

cloud data can be filtered both by feature scale and density. Unfortunately, in general, the resulting *multi-parameter persistence modules* obtained by applying the homology functor to a filtration indexed over $\mathbb{R}^n$ [Car09, Les15] are much less tractable; contrary to the single-parameter case, there is no general decomposition theorem that can break down any module into a direct sum of simple summands such as, e.g., interval modules.

Contributions. In this article, we build on the heuristic construction of [CB20] based on the fibered barcode [LW15] and propose the first approximate decomposition of general multi-parameter persistence modules in arbitrary dimension.

1. We introduce a new candidate approximate decomposition, parameterized by an approximation parameter $\delta > 0$, that can be computed for any multi-parameter persistence module with running time
$$O\left(N^3 + \frac{1}{\delta^n}(N + n \cdot 2^{n-1})\right),$$
where N is the number of simplices and n is the number of filtrations (Algorithm 1 in Section 3.3),
2. When computed over *interval decomposable* modules, we prove that the interleaving distance between our construction $\tilde{M}$ and the module M it approximates is upper bounded under mild assumptions (Proposition5.5):
$$d_I(M, \tilde{M}) \leq d_b(M, \tilde{M}) \leq \delta.$$
Even though our theoretical result only applies to interval decomposable modules, our candidate approximation $\tilde{M}$ always has the same fibered barcode and rank invariant than the module M it approximates. Moreover, we hypothesize that our approximate decomposition is also stable for modules that are close to being interval decomposable, while being more powerful than the rank invariant.
3. We perform numerical experiments that showcase the performance of this approximation and exhibit the trade-off between computation time and approximation error (Section 7).

Related work. There are several works in the literature that focused on the problem of computing or approximating multi-parameter persistence modules.

When restricted to filtrations indexed over $\mathbb{R}^2$, decomposition theorems have been provided under strong assumptions about the filter functions [ABE+21, BLO20, BLO22, BL18, CO19, Les15], as well as efficient algorithms for comparing these decompositions [KLO19, KN20, Vip20a]. Minimal presentations of bimodules of simplicial complexes can also be computed with `Rivet` [LW15] in $O(N^3\kappa + (N + \log\kappa)\kappa^2)$ operations, where N is the number of simplices, and $\kappa = \kappa_x\kappa_y$ is the product of unique x and y coordinates in the support of the module. Approximation schemes and methods to produce estimate modules have also been proposed with polynomial complexity, that are based on, e.g, Möbius inversions [AENY19], or rectangle summands [DX21]. While these running times are comparable to ours, we substantially generalize these approaches since our approximation can be computed for any number of filtrations.

For general multi-parameter persistence modules in dimension n, i.e., computed from filtrations indexed over $\mathbb{R}^n$, alternative descriptors of the multi-parameter persistence modules (that are complete under specific assumptions) have been presented [BOO21, CB20, CFK+19, Vip20b], and a decomposition algorithm for modules computed on simplicial complexes and indexed over a grid has been proposed [DX22]. This algorithm has complexity $O(N^{n(2\omega+1)})$, where N is the number of simplices (which can be typically more than cubic in the number of data points, depending on the homology dimension) and $\omega < 2.373$, and is thus very limited by the size of the input. Hence, computing approximate decompositions for multi-parameter persistence modules indexed over $\mathbb{R}^n$ for arbitrary $n \in \mathbb{N}^*$ with controlled complexity, running time, and approximation error is still an open and important question, which we tackle in this article.

Outline. Section 2 provides a concise review of multi-parameter persistence modules. In Section 3, we present our approximation scheme for general multi-parameter persistence modules, and in Sections 4 and 5, we study the theoretical properties of our construction for interval decomposable modules. We also discuss in depth the exact matching parameter of our construction in Section 6. Finally, we illustrate the performances of our candidate in Section 7.

2. Background

In this section, we recall the basics of multi-parameter persistence modules. This section only contains the necessary background and notations, and can be skipped if the reader is already familiar with persistence theory. A more complete treatment of persistence modules can be found in [Oud15, CdGO16, DW22]. A more precise description of multi-parameter persistence modules computed from filtered simplicial complexes can also be found in Section 6.3.

2.1. Multi-parameter persistence modules

In their most general form, i.e., regardless of whether they are computed from simplicial complexes, topological spaces, etc, multi-parameter persistence modules are nothing but k-vector spaces indexed by $\mathbb{R}^n$ and connected by linear transformations (where k denotes a field).

Definition 2.1 (Multi-parameter persistence module)**.** An n-multi-parameter persistence module (or n-multipersistence module for short) is a covariant functor M from $\mathbb{R}^n$ to the category of k-vector spaces, $M : x \in \mathbb{R}^n \mapsto M_x$. The linear transformations $\{\varphi_x^y : M_x \to M_y\}_{x,y\in\mathbb{R}^n, x\leq y}$ are called the *transition maps* of M. In particular, functoriality imposes the following property on the transition maps: $\varphi_x^z = \varphi_y^z \circ \varphi_x^y$ for any $x \leq y \leq z$.

A *morphism* between two n-multipersistence modules M, N with transition maps $\varphi_\cdot^\cdot$ and $\psi_\cdot^\cdot$ respectively, is a collection of linear maps $f = \{f_x : M_x \to N_x\}_{x\in\mathbb{R}^n}$, called an *$n$-multipersistence morphism*, that commutes with transitions maps, i.e., one has $f_y \circ \varphi_x^y = \psi_x^y \circ f_x$, for all $x \leq y$.

Multipersistence modules can be compared with the *interleaving distance* [Les15], which is one of the most commonly used distances in TDA, and which is based on the *shift functor*.

Definition 2.2 (Shift functor)**.** Let $v \in \mathbb{R}^n$. The *v-shift functor* is the endofunctor $(\cdot)(v)$ that maps an n-multipersistence module M (resp. an n-multipersistence morphism f) to $M(v)$ (resp. $f(v)$) defined, for any $x \in \mathbb{R}^n$, as $M(v)_x := M_{x+v}$ (resp. $f(v)_x := f_{x+v}$).

Definition 2.3 (Interleaving distance)**.** Given $\varepsilon > 0$, two n-multipersistence modules M and N are *$\boldsymbol{\varepsilon}$-interleaved* if there exist two morphisms $f\colon M \to N(\boldsymbol{\varepsilon})$ and $g\colon N \to M(\boldsymbol{\varepsilon})$ such that $g(\boldsymbol{\varepsilon}) \circ f = \varphi_\cdot^{\cdot+2\boldsymbol{\varepsilon}}$ and $f(\boldsymbol{\varepsilon}) \circ g = \psi_\cdot^{\cdot+2\boldsymbol{\varepsilon}}$, where $\boldsymbol{\varepsilon} = (\varepsilon, \ldots, \varepsilon) \in \mathbb{R}^n$, and φ and ψ are the transition maps of M and N respectively.

The interleaving (pseudo)distance between two multipersistence modules M and N is then defined as

$$d_I(M, N) = \inf\{\varepsilon \geq 0 : M \text{ and } N \text{ are } \boldsymbol{\varepsilon}\text{-interleaved}\}.$$

Another usual distance is the *bottleneck distance* [BL18, Section 2.3]. Intuitively, it relies on decompositions of the modules into direct sums of summands, and is defined as the interleaving distance between these summands. As such, it first requires the definition of a *matching* between summands.

Definition 2.4 (Matching)**.** Given two multisets A and B, $\mu\colon A \nrightarrow B$ is called a *matching* if there exist $A' \subseteq A$ and $B' \subseteq B$ such that $\mu\colon A' \to B'$ is a bijection. In the following, we let $\mathrm{im}(\mu) = B'$ and $\mathrm{coim}(\mu) = A'$.

Moreover, in order to define meaningful decompositions, summands are required to be *indecomposable modules*.

Definition 2.5 (Indecomposable module)**.** A multipersistence module M is *indecomposable* if

$$M \cong A \oplus B \Longrightarrow M \simeq A \text{ or } M \simeq B$$

Since decompositions of multipersistence modules are unique [Oud15], the following distance is well-defined.

Definition 2.6 (Bottleneck distance)**.** Let $M \cong \bigoplus_{i\in\mathcal{I}} M_i$ and $N \cong \bigoplus_{j\in\mathcal{J}} N_j$ be two multipersistence modules decomposed into indecomposable summands. Given $\varepsilon \geq 0$, the modules M and N are *$\boldsymbol{\varepsilon}$-matched* if there exists a matching $\sigma\colon \mathcal{I} \nrightarrow \mathcal{J}$ such that

1. for all $i \in \mathcal{I}\backslash\mathrm{coim}(\sigma)$, M_i is $\boldsymbol{\varepsilon}$-interleaved with the null module $\mathbf{0}$,
2. for all $j \in \mathcal{J}\backslash\mathrm{im}(\sigma)$, N_j is $\boldsymbol{\varepsilon}$-interleaved with the null module $\mathbf{0}$,
3. for all $i \in \mathrm{coim}(\sigma)$, M_i and $N_{\sigma(i)}$ are $\boldsymbol{\varepsilon}$-interleaved.

The *bottleneck distance*, denoted by d_b, between two multipersistence modules M and N is then defined as

$$d_b(M, N) = \inf\{\varepsilon \geq 0 \ : \ M \text{ and } N \text{ are } \boldsymbol{\varepsilon}\text{-matched}\}.$$

Since a matching between the decompositions of two multipersistence modules induces an interleaving between the modules themselves, it follows that $d_I \leq d_b$. Note that the bottleneck distance can actually be arbitrarily larger than the interleaving distance, as showcased in [BL18, Section 9].

2.2. Interval modules

The study of multipersistence modules is easier when restricted to a specific class called the *interval modules*. For instance, all of our theoretical results presented in Sections 4 and 5 are stated for modules that can be decomposed into intervals. Hence, in this section, we define such interval modules. Intuitively, they are modules that are trivial, except on a subset of $\mathbb{R}^n$ called an *interval*.

Definition 2.7 (Interval)**.** A subset I of $\mathbb{R}^n$ is called an *interval* if it satisfies:

- (convexity) if $p, q \in I$ and $p \leq r \leq q$ then $r \in I$, and
- (connectivity) if $p, q \in I$, then there exists a finite sequence $r_1, r_2, \ldots, r_m \in I$, for some $m \in \mathbb{N}$, such that $p \sim r_1 \sim r_2 \sim \cdots \sim r_m \sim q$, where $\sim$ can be either $\leq$ or $\geq$.

Definition 2.8 (Indicator module, Interval module)**.** An n-multipersistence module M is an *n-indicator module* if there exists a set $S \subseteq \mathbb{R}^n$, called the *support* of M and denoted by $\mathrm{supp}(M)$, such that:

$$\forall x \in \mathbb{R}^n,\ M_x = \begin{cases} k \text{ if } x \in S \\ \mathbf{0} \text{ otherwise} \end{cases} \qquad \text{and} \quad \forall x, y \in \mathbb{R}^n,\ \varphi_x^y = \begin{cases} \mathrm{id}_{k \to k} \text{ if } x \leq y \in S \\ 0 \text{ otherwise} \end{cases}$$

where $\varphi_{\cdot}^{\cdot}$ are the transition maps of M. If S is an interval, M is called an *n-interval module*.

An important consequence of modules is that whenever two points are in the support of an indicator module, then the whole rectangle induced by those points must be in the support as well, as stated by the following lemma.

Lemma 2.9. *Let I be an n-indicator module. Then one has $a, b \in \mathrm{supp}(I) \Leftrightarrow R_{a,b} \subseteq \mathrm{supp}(I)$, where, given two points $a, b \in \mathbb{R}^n$, the corresponding rectangle $R_{a,b}$ is defined as $R_{a,b} := \{x \in \mathbb{R}^n : a \leq x \leq b\}$, and $R_{a,b} = \emptyset$ if $b < a$ or if a and b are not comparable in $\mathbb{R}^n$.*

Proof. Since $\Leftarrow$ is trivial, we only prove $\Rightarrow$. If $x \not\leq y$, then $R_{x,y}$ is empty and the result holds. Otherwise, if $R_{x,y}$ is not empty, let $z \in R_{x,y}$, i.e., $x \leq z \leq y$, and let $\varphi_{\cdot}^{\cdot}$ be the transition maps of I. Since $\varphi_x^y = \varphi_z^y \circ \varphi_x^z = \mathrm{id}$, one has $1 \geq \dim I_z \geq \dim I_x = \dim I_y = 1$ (see Definition 2.8), which means that $z \in \mathrm{supp}(I)$. □

Note that one cannot use any set S for defining an indicator module, since transition maps of modules must satisfy some properties coming from functoriality (see Definition 2.1). However, one can defined a module *induced from a set* using the following definition.

Definition 2.10 (Induced module)**.** Given a subset $S \subseteq \mathbb{R}^n$, the indicator module $\mathbf{Ind}(S)$ induced by S is defined as the indicator module with support $\{x \in \mathbb{R}^n : \exists a, b \in S \text{ such that } a \leq x \leq b\}$.

Finally, *interval decomposable modules* are then defined as those multipersistence modules that are made of intervals.

Definition 2.11 (Interval decomposable module)**.** An *interval decomposable module M* is a multipersistence module that is isomorphic to a direct sum of interval modules.

Note that for *rectangle decomposable modules*, i.e., interval decomposable modules whose supports are rectangles in $\mathbb{R}^n$, it is possible to control the bottleneck distance more precisely with $d_b \leq (2n-1)d_I$ [Bje20, Theorem 4.3]. In the following, we present a few properties of interval modules that are often very useful for their theoretical analysis.

Definition 2.12 (Discretely presented interval module)**.** An n-interval module I is *discretely presented* if its support is a locally finite union of rectangles in $\mathbb{R}^n$, and whose boundary is an $(n-1)$-submanifold of $\mathbb{R}^n$. More precisely, there exist two families of points, the *birth* and *death critical points* of I, denoted by $C_B(I)$ and $C_D(I)$ respectively, such that:

$$I = \mathbf{Ind}\left(\bigcup_{c \in C_B(I)} \bigcup_{c' \in C_D(I)} R_{c,c'} \right). \tag{1}$$

A useful property of interval modules is that they can be described with their *upper-* and *lower-boundaries*, also called *upsets* and *downsets* [Mil20, Section 1.4].

Definition 2.13 (Upper- and lower-boundaries)**.** Given an interval I, its *upper-boundary* $U[I]$ and *lower-boundary* $L[I]$ are defined as:

$$L[I] := \left\{x \in \bar{I} : \forall y \in \mathbb{R}^n,\ y < x \Rightarrow y \notin I\right\},\ \ U[I] := \left\{x \in \bar{I} : \forall y \in \mathbb{R}^n,\ y > x \Rightarrow y \notin I\right\}$$

Moreover, the boundary of $\mathrm{supp}(I)$ can be decomposed with $\partial\mathrm{supp}(I) = L[I] \cup U[I]$. See Figure 1 for an illustration.

When interval modules are discretely presented, their lower- and upper-boundaries are made of flat parts, which are the faces of the corresponding rectangles forming the module. Hence, we call *facets* the subsets of the lower- and upper-boundaries that are included in some hyperplanes of $\mathbb{R}^n$.

Definition 2.14 (Facet)**.** A *lower* (resp. *upper*) *facet* of an interval I is an $(n-1)$-submanifold of $\partial\mathrm{supp}(I)$ written as $\{x \in \mathbb{R}^n : x_i = c\} \cap L[I]$ (resp. $\{x \in \mathbb{R}^n : x_i = c\} \cap U[I]$) for some $c \in \mathbb{R}$ and some dimension $i \in [\![1, n]\!]$ that is called the facet *codirection*. In particular, the upper- and lower-boundaries of a discretely presented interval module is a (locally) finite union of facets.

2.3. Interval morphisms

For indicator modules, there is only one possible transition map, the identity (up to an invertible scalar). This induces a canonical way to define morphisms between indicator modules (and thus between interval modules as well).

Definition 2.15 (Indicator module morphisms)**.** Let I and $\tilde{I}$ be two indicator modules. The collections of linear maps $\varphi^{(\cdot)}_{I\to\tilde{I}}$ and $\varphi^{(\cdot)}_{\tilde{I}\to I}$ between I and $\tilde{I}$ are called *indicator module morphisms*, and defined with

$$\begin{array}{ccc} \tilde{I} & \xrightarrow{\varphi^{(2\varepsilon)}_{\tilde{I}}} & \tilde{I}[2\boldsymbol{\varepsilon}] \\ & {\scriptstyle \varphi^{(\varepsilon)}_{\tilde{I}\to I}} \searrow \quad \nearrow {\scriptstyle \varphi^{(\varepsilon)}_{I\to\tilde{I}}} & \\ & I[\boldsymbol{\varepsilon}] & \end{array} \quad \text{and} \quad \begin{array}{ccc} I & \xrightarrow{\varphi^{(2\varepsilon)}_{I}} & I[2\boldsymbol{\varepsilon}] \\ & {\scriptstyle \varphi^{(\varepsilon)}_{I\to\tilde{I}}} \searrow \quad \nearrow {\scriptstyle \varphi^{(\varepsilon)}_{\tilde{I}\to I}} & \\ & \tilde{I}[\boldsymbol{\varepsilon}] & \end{array} \tag{2}$$

where $\boldsymbol{\varepsilon} = (\varepsilon, \dots, \varepsilon) \in \mathbb{R}^n$ and where $\varphi^{(\varepsilon)}_{I\to\tilde{I}}$ is defined, for $x \in \mathbb{R}^n$, by

$$\begin{aligned} (\varphi^{(\varepsilon)}_{I\to\tilde{I}})_x : I_x \simeq k \text{ or } \{0\} &\longrightarrow \tilde{I}_{x+\boldsymbol{\varepsilon}} \simeq k \text{ or } \{0\} \\ y &\longmapsto \begin{cases} y \text{ if } x + \boldsymbol{\varepsilon} \in \operatorname{supp}(\tilde{I}) \\ 0 \text{ otherwise} \end{cases} , \end{aligned}$$

and vice-versa for $\varphi^{(\varepsilon)}_{\tilde{I}\to I}$. Note that $\varphi^{(\varepsilon)}_{I\to\tilde{I}}$ and $\varphi^{(\varepsilon)}_{\tilde{I}\to I}$ define an $\boldsymbol{\varepsilon}$-interleaving between I and $\tilde{I}$ if they commute.

2.4. Fibered barcode

The *fibered barcode* [LW15] is a centerpiece of our approximation scheme, and is defined, given an n-multipersistence module M, as a map that takes as input a line (or segment) l in $\mathbb{R}^n$, and outputs the persistence barcode associated to the single-parameter persistence module obtained by restricting M along l. Hence, in the following, we formalize and define intersections between multipersistence modules and lines in $\mathbb{R}^n$.

Definition 2.16. Given a line $l \subseteq \mathbb{R}^2$, we let ι_l denote the induced functor $\iota_l : \mathbf{L} \to \mathbb{R}^2$, where $\mathbf{L}$ is the full subcategory of $\mathbb{R}^2$ associated to l. The module $M\big|_l := M \circ \iota_l$ is called the *restriction of M to l*.

Remark 2.17. When $M = \bigoplus_{i\in\mathcal{I}} M_i$ is decomposable into indicator modules, the support of the restriction of M to l is a set of segments called *bars*, and aggregated in a *barcode*: $\mathcal{B}(M\big|_l) := \operatorname{supp}(M\big|_l) := \left(\operatorname{supp}(M_i\big|_l)\right)_{i\in\mathcal{I}}$. Note that this barcode corresponds exactly to the barcode defined in the theory of single-parameter persistence, computed on the single-parameter filtration induced by $l \subseteq \mathbb{R}^2$.

Definition 2.18 (Fibered Barcode)**.** Let $M = \bigoplus_{i\in\mathcal{I}} M_i$ be a pointwise finite-dimensional n-multipersistence module. The *complete fibered barcode* of M is defined as the family of barcodes $\mathcal{CFB}(M) = \{\mathcal{B}(M\big|_l) : l \in \mathcal{L}\}$, where $\mathcal{L}$ denotes the set of *diagonal* lines in $\mathbb{R}^n$, i.e., those lines with direction vector $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$. Given a (possibly discrete) family of diagonal lines $L \subseteq \mathcal{L}$, we let the *L-fibered barcode* (or *fibered barcode* for short when L is clear) be the restriction of the complete fibered barcode to L, i.e., $\mathcal{FB}(M)_L = \{\mathcal{B}(M\big|_l) : l \in L\}$.

It is also useful to characterize intersections between modules and lines using the *endpoints* of lines.

Definition 2.19 (Birthpoint, Deathpoint)**.** Given a point $x \in \overline{\mathbb{R}}^n$ and an indicator module I, we call $b^I_x := \{x + \boldsymbol{\delta} : \delta \in \mathbb{R}\} \cap L[I]$ (resp. $d^I_x = \{x + \boldsymbol{\delta} : \delta \in \mathbb{R}\} \cap U[I]$) the *birthpoint* (resp. *deathpoint*) associated to x and I (see Figure 1 for an illustration), where $\boldsymbol{\delta} = (\delta, \dots, \delta) \in \mathbb{R}^n$. Since it follows from the definition of $L[I]$ and $U[I]$ that b^I_x and d^I_x are singletons, we slightly abuse notations and use b^I_x and d^I_x to also refer to the unique element these sets contain. When these sets are empty, or $b^I_x = d^I_x$, we say b^I_x and d^I_x are *trivial*.

Similarly, given a diagonal line $l \subseteq \mathbb{R}^n$ (i.e., a line with direction vector $(1, \dots, 1) \in \mathbb{R}^n$), we define the *birthpoint* (resp. *deathpoint*) associated to l and I as $b^I_l := b^I_x$ (resp. $d^I_l := d^I_x$) for any $x \in l$.

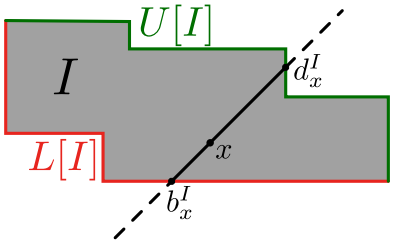

Figure 1: Lower- and upper-boundaries of a 2-interval (Definition 2.13); and birthpoints and deathpoints b^I_x and d^I_x (Definition 2.19) of a point $x \in \mathbb{R}^2$.

Remark 2.20. The rectangle $R_{a,b}$ induced by two birthpoints or deathpoints a, b of the same indicator module is always flat, i.e., at least one of its sides has length zero, as demonstrated by Figure 2.

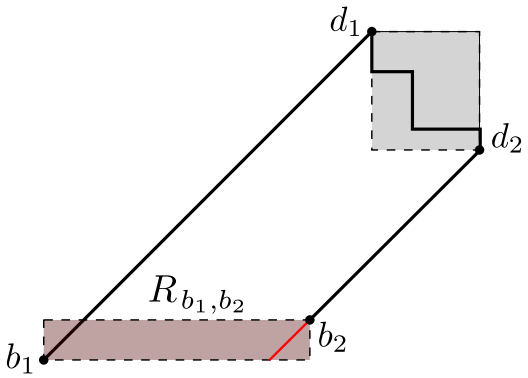

Figure 2: Two bars $[b_1, d_1]$ and $[b_2, d_2]$ of some indicator module; if R_{b_1,b_2} is not flat then, by Lemma 2.9, b_2 cannot be a birthpoint since it would be possible to find a smaller birthpoint w.r.t. the partial order of $\mathbb{R}^n$ along the diagonal line passing through b_2.

Remark 2.21. Using birthpoints and deathpoints, the L-fibered barcode of an interval decomposable multipersistence module $M = \bigoplus_{i \in I} M_i$ is written as:

$$\mathcal{FB}(M)_L = \{\mathcal{B}(M|_l) : l \in L\} = \{([b_l^{M_i}, d_l^{M_i}])_{i \in I} : l \in L\}. \tag{3}$$

Note also that bars of the fibered barcode that are associated to lines that are close to each other must have similar length, as stated in the lemma below, which is very similar to [Lan18, Lemma 2].

Lemma 2.22. *Let I be an indicator module, let $l_1, l_2 \subseteq \mathbb{R}^n$ be two diagonal lines and let $\overrightarrow{v} \in \mathbb{R}^n$ be a positive or negative vector (i.e., the coordinates of $\overrightarrow{v}$ are either all positive or all negative) such that $l_2 = l_1 + \overrightarrow{v}$. Assume that the barcodes $\mathcal{B}(I|_{l_1})$ and $\mathcal{B}(I|_{l_2})$ are non empty, and let $[b^I_{l_1}, d^I_{l_1}]$ and $[b^I_{l_2}, d^I_{l_2}]$ be the corresponding bars in $\mathbb{R}^n$. Then, one has*

$$\left\| d^I_{l_1} - d^I_{l_2} \right\|_\infty \leq \left\| \overrightarrow{v} \right\|_\infty \text{ and } \left\| b^I_{l_1} - b^I_{l_2} \right\|_\infty \leq \left\| \overrightarrow{v} \right\|_\infty,$$

where we used the conventions $(+\infty) - (+\infty) = (-\infty) - (-\infty) = 0$.

Proof. If one of the endpoints is infinite, the result holds trivially, so we now assume that the endpoints of the bars are all finite. Without loss of generality, assume that $l_2 = l_1 + \overrightarrow{v}$ where $\overrightarrow{v}$ is positive. Now, since both $d^I_{l_2}$ and $d^I_{l_1} + \overrightarrow{v}$ belong to l_2, they are comparable, so one has either $d^I_{l_2} > d^I_{l_1} + \overrightarrow{v}$ or $d^I_{l_2} \leq d^I_{l_1} + \overrightarrow{v}$. However, the first possibility would lead to $d^I_{l_2} > d^I_{l_1} + \overrightarrow{v} > d^I_{l_1}$, hence $d^I_{l_1}$ and $d^I_{l_2}$ would be (strictly) comparable in $\mathbb{R}^n$, which contradicts Remark 2.20. Thus, one must have $d^I_{l_2} \leq d^I_{l_1} + \overrightarrow{v}$. Furthermore, and using the exact same arguments, $d_{l_2} - \overrightarrow{v} + \left\| \overrightarrow{v} \right\|_\infty \cdot \mathbf{1}$ is on l_1, and one must have $d^I_{l_2} - \overrightarrow{v} + \left\| \overrightarrow{v} \right\|_\infty \cdot \mathbf{1} \geq d^I_{l_1}$. Finally, by combining the two previous inequalities, one has

$$d^I_{l_1} - \left\| \overrightarrow{v} \right\|_\infty \cdot \mathbf{1} \leq d^I_{l_1} + \overrightarrow{v} - \left\| \overrightarrow{v} \right\|_\infty \cdot \mathbf{1} \leq d^I_{l_2} \leq d^I_{l_1} + \overrightarrow{v} \leq d^I_{l_1} + \left\| \overrightarrow{v} \right\|_\infty \cdot \mathbf{1},$$

which leads to the desired inequality for deathpoints. The proof extends straightforwardly to birthpoints. □

3. General multipersistence module approximation

In this section, we present the first approximation scheme that works for any multipersistence module in arbitrary dimension, i.e., with arbitrary number of filtrations. In particular, one does not have to assume that the underlying module is decomposable in order to apply our method. Our candidate approximation works by sampling the underlying module with an ordered family of diagonal lines, computing the associated fibered barcode, and finally connecting the endpoints of bars in consecutive barcodes in a specific way. Note however that the theoretical properties that we show for our approximation in Sections 4 and 5 are valid only when the underlying module is interval decomposable. We first provide in Section 3.1 a few specific examples of approximation schemes based on fibered barcodes that we present to gather intuition and motivation for our main construction, that we then present in Section 3.2. The corresponding pseudo-code and algorithm are given in Section 3.3.

3.1. Motivation

The goal of this section is to frame the general question of reconstructing a multipersistence module from its fibered barcode. There are many ways of doing so, but the most natural ones are not necessarily the easiest computable ones.

For the sake of simplicity, assume that the underlying module is a single interval module $M = I$ (see Definition 2.8). Since interval modules are characterized by their supports, the goal is to recover $\mathrm{supp}(I)$. Moreover, if I is discretely presented, one can find an exact sequence of graded modules

$$R \longrightarrow G \twoheadrightarrow I \longrightarrow 0,$$

such that the critical points of $\mathrm{supp}(I)$ (see Equation (1)) provide bases for the modules G and R (we recall that, intuitively, G and R represent the homology generators and relations respectively). Technical details for finding such exact sequences can be found in, e.g., [DX22, Appendix A], and an example of such a sequence is given in Figure 3. Hence, only the facets or critical points of $\mathrm{supp}(I)$ need to be captured or approximated in order to recover I when it is discretely presented.

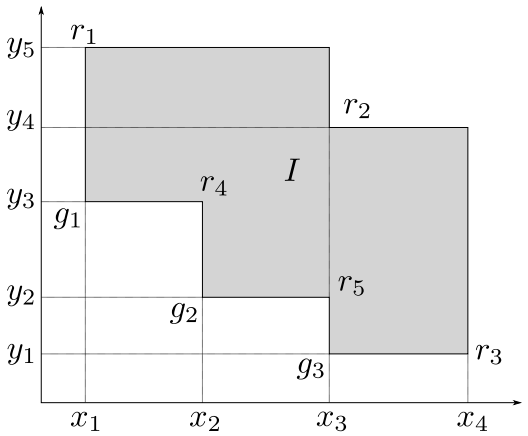

Figure 3: Let I be the interval whose support is colored in grey, and let $\varphi_{\cdot}^{\cdot}$ denote the transition maps of I. The graded modules R and G can be constructed as follows: G is defined as the free graded module $G := \langle g_1, g_2, g_3 \rangle$ (where the grades of g_1, g_2 and g_3 are (x_1, y_3), (x_2, y_2) and (x_3, y_1) respectively, i.e., they are given by their positions in the figure), and R is defined as the (not necessarily free) graded module $R := \langle r_1, \ldots, r_5 \rangle$ (where the grades of the r_i's, are also given by the figure). Since one has $r_1 = \varphi_{(x_1,y_3)}^{(x_1,y_3)+(y_5-y_3)e_2}(g_1)$, $r_2 = \varphi_{(x_3,y_1)}^{(x_3,y_1)+(y_4-y_1)e_2}(g_3)$, $r_3 = \varphi_{(x_3,y_1)}^{(x_3,y_1)+(x_4-x_3)e_1}(g_3)$, $r_4 = \varphi_{(x_1,y_3)}^{(x_1,y_3)+(x_2-x_1)e_1}(g_1) - \varphi_{(x_2,y_2)}^{(x_2,y_2)+(y_3-y_2)e_2}(g_2)$, and $r_5 = \varphi_{(x_2,y_2)}^{(x_2,y_2)+(x_3-x_2)e_1}(g_2) - \varphi_{(x_3,y_1)}^{(x_3,y_1)+(y_2-y_1)e_2}(g_3)$, it follows that $R \longrightarrow G \twoheadrightarrow I \longrightarrow 0$ is an exact sequence.

There are many different ways, for a given n-interval module I, to define candidate critical points, that we call *corners*, using the endpoints of its fibered barcode, e.g., by using the minimum and maximum of consecutive endpoint coordinates. Hence, it is natural to define our candidate approximation $\tilde{I}$ with model selection, i.e., by minimizing some penalty cost $\mathrm{pen}\colon S \longrightarrow \mathbb{R}_+$, where S is the set of discretely presented interval modules having the same fibered barcode as I, or a subset thereof. See Figure 4 for examples of sets S and corresponding candidate approximations. This penalty would forbid, e.g., overly complicated approximations that have a lot of corners. For instance, minimizing the penalty

$$\mathrm{pen} : \tilde{I} \mapsto \#\text{corners of } \mathrm{supp}(I). \tag{4}$$

would provide a sparse approximation of I. Actually, when one assumes that the underlying interval module I is discretely presented with facets that are large enough with respect to the family of lines L of the fibered barcode (see Proposition 4.4 for precise statements), it is easy to show that I minimizes penalty (4). Indeed, as all the facets are detected by some endpoints of the fibered barcode, any candidate has at least the same number of facets than I.

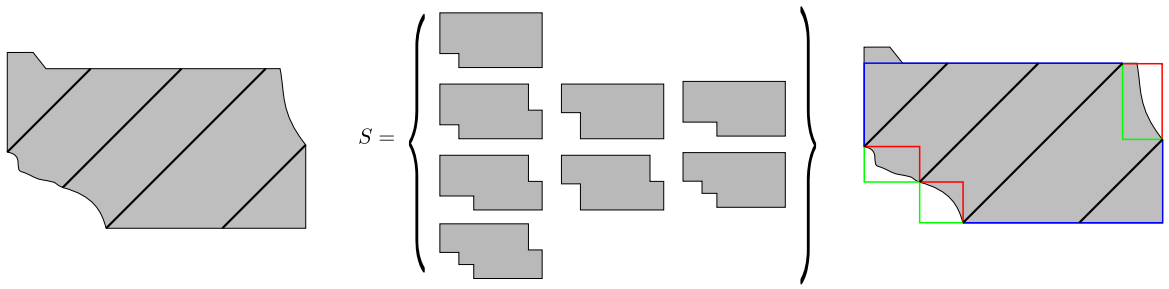

Figure 4: Example of candidates for a 2-interval module I with support in $\mathbb{R}^2$. **(Left)** Given the L-fibered barcode of I, where L is the family of the four black lines, we want to approximate I with an element of S, i.e., an interval module with the same fibered barcode. **(Middle)** When one further constrains the set S by asking to have at most one corner between two consecutive endpoints of the fibered barcode, the whole set S can be computed explicitly. **(Right)** The set S can also be described as the set of intervals which have to go through the blue path, and which can arbitrarily choose between the red or green path at three different locations. Hence, the cardinality of S is 2^3.

Remark 3.1. For n-interval modules, S is generally a set of cardinal c^d, where c is the number of possible corners between 2^{n-1} birthpoints or deathpoints, and d is the number of corners. For instance, in Figure 4, one has $n = 2$, $c = 2$ and $d = 3$. Unfortunately, c grows exponentially with the dimension n, and d is difficult to control in practice, since it heavily depends on the number of lines in the fibered barcode and the regularity of the underlying interval module I. Minimizing a penalty over S is thus practical only for low dimension n and small number of lines in the fibered barcode. Hence, our general approximation scheme presented in Section 3.3 does not use penalty minimization, but is rather defined with arbitrary corner choices.

Note also that there are cases when the corner choices are canonical. For instance, any 2-multipersistence module M with transition maps $\varphi_{\cdot}^{\cdot}$ that is *weakly exact*, i.e., that satisfy, for any $x \leq y$

$$\operatorname{im}\left(\varphi_x^y\right) = \operatorname{im}\left(\varphi_{(y_1,x_2)}^y\right) \cap \operatorname{im}\left(\varphi_{(x_1,y_2)}^y\right) \text{ and } \ker\left(\varphi_x^y\right) = \ker\left(\varphi_x^{(y_1,x_2)}\right) + \ker\left(\varphi_x^{(x_1,y_2)}\right),$$

is rectangle decomposable [BLO22]. Hence, a canonical approximation of a summand I of M is given by the interval module whose support is the rectangle with corners $(\min_l(b_l^I)_1, \min_l(b_l^I)_2)$ and $(\max_l(b_l^I)_1, \max_l(b_l^I)_2)$, where l goes through the family of lines L of the fibered barcode.

3.2. Line families, corners and exact matchings

In this section, we provide three additional definitions that turn out very useful for describing our approximation scheme in Section 3.3, as well as for proving associated guarantees for interval decomposable modules in Sections 4 and 5.

We first introduce a few notations: we let $(e_1, \dots, e_n)$ be the canonical basis of $\mathbb{R}^n$, d_∞ denote the $\|\cdot\|_\infty$ distance in $\mathbb{R}^n$, and, given a set $A \subseteq \mathbb{R}^n$ and $\delta > 0$, we let A^δ denote the δ-*offset* of A, defined as $A^\delta := \{x \in \mathbb{R}^n : d_\infty(x, A) \leq \delta\}$, and we let $\operatorname{conv}(A)$ denote the convex hull of A. Moreover, given a hyperplane $H \subseteq \mathbb{R}^n$ and its two associated vectors $a_H, b_H \in \mathbb{R}^n$ which satisfy $H = b_H + \{x \in \mathbb{R}^n : \langle x, a_H \rangle = 0\}$, we call a_H the *codirection* of H (similarly to the codirection of facets, see Definition 2.14). Finally, when a_H is a vector in the canonical basis of $\mathbb{R}^n$, i.e., there exists $i \in [\![1, n]\!]$ such that $a_H = e_i$, we slightly abuse notation and also call i the codirection of H.

Our first definition characterize those families of lines that evenly cover compact sets in $\mathbb{R}^n$.

Definition 3.2 (δ-regularly distributed lines filling a compact set)**.** Let L be a set of diagonal lines in $\mathbb{R}^n$ and $K \subseteq \mathbb{R}^n$ be a compact set. Then, we say that :

1. two diagonal lines $l, l' \in L$ are δ-*consecutive* (or simply *consecutive* when δ is clear) if there exists $\mathbf{u} \in \{0,1\}^n \setminus \{\mathbf{0}, \mathbf{1}\}$ such that $l' = l \pm \delta\mathbf{u}$.
2. two diagonal lines $l, l' \in L$ are δ-*comparable* if there exists a positive or negative vector $\overrightarrow{u} \in \mathbb{R}^n$ with $\left\|\overrightarrow{u}\right\|_\infty \leq \delta$ such that $l' = l + \overrightarrow{u}$, where $\overrightarrow{u}$ is said to be positive (resp. negative), written as $\overrightarrow{u} \geq 0$ (resp. $\overrightarrow{u} \leq 0$), if and only if $(\overrightarrow{u})_i \geq 0$ (resp. $(\overrightarrow{u})_i \leq 0$) for all $i \in [\![1, n]\!]$. If $\overrightarrow{u}$ is positive (resp. negative), we write $l' \geq l$ (resp. $l' \leq l$).
3. L is δ-*regularly distributed* if, for any pair of lines $(l, l') \in L$, there exists a sequence of δ-consecutive lines $\{l_1, \dots, l_k\}$ in L such that $l = l_1$ and $l' = l_k$.
4. for a given line l in a δ-regularly distributed family of lines L, we call $L_l := L \cap \{l + \overrightarrow{v} : \overrightarrow{v} \in \{0,1\}^{n-1} \times \{0\}\}$ the L-*surrounding set* of l. In particular, one has $|L_l| \leq 2^{n-1}$.
5. L δ-*fills* K if any point of K is at distance at most $\delta/2$ from some line in L. More formally, K is included in the offset $L^{\delta/2}$. When δ is clear from the context, we simply say that L *fills* K.

Remark 3.3. Let L be a set of δ-regularly distributed diagonal lines that δ-fills some compact set $K \subseteq \mathbb{R}^n$. Then L is actually distributed over a grid (along the canonical axes of $\mathbb{R}^n$) on K. More precisely, assume that there is a $l \in L$ such that $\mathbf{0} \in l$. Now, assume that there exist integers $\alpha_1, \dots, \alpha_n \in \mathbb{Z}$ such that $x = (\alpha_1\delta, \dots, \alpha_n\delta) \in K$. Then, using items (3) and (5) of Definition 3.2, there must exist some line $l_x \in L$ such that $x \in l_x$. Hence, to be more concise, we will call such a set of lines a *δ-grid* of K.

Families of lines that are δ-grids of K allow to formally define candidate critical points, or *corners*, that can be used to approximate the critical points of the true underlying interval module (see Section 3.1 above). In the following definition, we introduce points called *corners* that can be defined solely from the fibered barcode of an interval module I, and that we will use as proxies for the critical points of I (as per Equation (1)) in our approximation scheme.

Definition 3.4 (Birth and death corners)**.** Given a discretely presented interval I with birth and death corners included in a compact set $K \subseteq \mathbb{R}^n$, and a δ-grid L of the offset $K^{2\delta}$, we say that b is a *finite (L-)birth corner* (resp. d is a *finite (L-)death corner*) if:

1. for each dimension $i \in [\![1, n]\!]$, there exists an hyperplane H_i of codirection i intersecting K, and the family $(H_i)_i$ satisfies $\bigcap_i H_i = b$ (resp. $\bigcap_i H_i = d$),
2. there exists a line $l_0 \in L$ such that
   (a) $b \in \mathrm{conv}(L_{l_0})$ (resp. $d \in \mathrm{conv}(L_{l_0})$), where L_{l_0} is the L-surrounding set of l_0 (see Definition 3.2)
   (b) for each line $l \in L_{l_0}$, the endpoint b_l^I (resp. d_l^I) is non trivial,
   (c) for each dimension $i \in [\![1, n]\!]$, there exists $l_i \in L_{l_0}$ such that $b_{l_i}^I \in H_i$.

and we say that b is a *pseudo (L-)birth corner* (resp. d is a *pseudo (L-)death corner*) if:

1. there exists a set $\mathcal{J} \subseteq [\![1, n]\!]$, called the *codirection* of b (resp. d) and denoted with $\mathrm{codir}(b)$ (resp. $\mathrm{codir}(d)$), such that for each dimension $j \in \mathcal{J}$, there exists a hyperplane of codirection j intersecting K such that $\bigcap_j H_j \ni b$ (resp. $\bigcap_j H_j \ni d$). The set $[\![1, n]\!] \backslash \mathcal{J}$ is called the *direction* of b (resp. d) and is denoted with $\mathrm{dir}(b)$ (resp. $\mathrm{dir}(d)$).
2. there exists a line $l_0 \in L$ such that
   (a) $b \in \mathrm{conv}(L_{l_0}) \cap K^{2\delta} \backslash K$ (resp. $d \in \mathrm{conv}(L_{l_0}) \cap K^{2\delta} \backslash K$),
   (b) for each line $l \in L_{l_0}$, the endpoint b_l^I (resp. d_l^I) is non trivial,
   (c) for each dimension $j \in \mathcal{J}$, there exists $l_j \in L_{l_0}$ such that $b_{l_j}^I \in H_j$.

A pseudo birth (resp. death) corner b is said to be *minimal* (resp. *maximal*) if for any other pseudo birth corner b' (resp. pseudo death corner d') such that $\mathrm{codir}(b') \subseteq \mathrm{codir}(b)$ (and thus $\mathrm{dir}(b') \supseteq \mathrm{dir}(b)$), there exists a dimension i such that $b_i < b'_i$ (resp. $d_i > d'_i$).

Finally, we say that b (resp. d) is an *infinite (L-)birth (resp. death) corner* if there exists a minimal (resp. maximal) pseudo birth (resp. death) corner b' (resp. d') such that $b_i = -\infty$ (resp. $d_i = +\infty$) if $i \in \mathrm{dir}(b')$ (resp. $i \in \mathrm{dir}(d')$) and $b_i = b'_i$ (resp. $d_i = d'_i$) if $i \in \mathrm{codir}(b')$.

Finally, in Section 3.1 and Definitions 3.2 and 3.4 above, we have assumed that the true underlying module was a single interval module. In order to handle more general multipersistence modules, we need a way to be able to distinguish between the bars of the fibered barcodes of different summands (when the module is decomposable). This is precisely the role of *exact matchings*, which we define below. In other words, exact matchings are functions that connect bars of different barcodes from the fibered barcode in a way that is consistent with the decomposition of the underlying multipersistence module M.

Definition 3.5 (Exact matching)**.** Let $M = \bigoplus_{i \in \mathcal{I}} M_i$ be an n-multipersistence module. Let l, l' be two lines in $\mathbb{R}^n$. A map m between the corresponding barcodes $m \colon \mathcal{B}(M\big|_l) \to \mathcal{B}(M\big|_{l'}) \cup \{\varnothing\}$ is called a *matching* between l and l' if the restriction of m to $m^{-1}(\mathcal{B}(M\big|_{l'}))$ is injective.

Furthermore, if we also assume that M is interval decomposable, i.e., that the M_i's are interval modules, then we say that the matching m is *exact* if one has $\iota_1(b) = \iota_2(m(b))$ for any bar $b \in \mathcal{B}(M\big|_l)$, where $\iota_1 : \mathcal{B}(M\big|_l) \to \mathcal{I}$ and $\iota_2 : \mathcal{B}(M\big|_{l'}) \to \mathcal{I}$ are correspondences between the bars of barcodes in the fibered barcode and the interval summands of M, obtained by Equation (3). In other words, bars that are matched under m correspond to the same underlying interval summand of M.

There are many ways of defining exact matching functions. For instance, under some assumptions, matchings induced by the Wasserstein distance between barcodes are exact (see Section 6.1), and we prove that the matching given by the *vineyard* algorithm [CSEM06] is exact in Section 6.3.

We are now equipped for stating our approximation scheme, which constructs a candidate module by computing corners from the fibered barcode and exact matchings.

3.3. Algorithms for approximating modules

In this section, we provide Algorithm 1, that can approximate any multipersistence module using the fibered barcode of an appropriate set of lines. Roughly speaking, Algorithm 1 works in three steps:

1. compute the L-fibered barcode of the underlying module,
2. match bars that correspond to the same underlying summand together using an exact matching, and
3. for each summand, use the endpoints of the corresponding bars to build compute corners, using Algorithm 2.

Step 1 can be performed using any persistent homology software (such as, e.g., `Gudhi`, `Ripser`, `Phat`, etc), or with `Rivet` [LW15] when $n = 2$. Our code can be found at `https://gitlab.inria.fr/dloiseau/multipers`, and is based on the vineyard update algorithm [CSEM06], which allows to run steps 1 and 2 jointly (see Section 6.3).

Note that while we can guarantee that the output is close to the underlying multipersistence module M only when M is decomposable into interval summands (see Sections 4 and 5), Algorithm 1 makes no assumption about M at all and can be applied generally. Note however that since Algorithm 1 always returns a multipersistence module that is interval decomposable, the output decomposition is obviously wrong if M is not decomposable.

Algorithm 1: ApproximateModule

Input 1: Multipersistence module M,
Input 2: Family of lines L which is a δ-grid of the offset $K^{2\delta}$ of a compact set $K \subseteq \mathbb{R}^n$
Input 3: Exact matching m
Output: Interval decomposable multipersistence module $\tilde{M}$
Compute $\mathcal{FB}(M)_L$, i.e., the L-fibered barcode of M;
$S \leftarrow []$; # S is the set of interval summands, intialized as the empty set
for $l \in L$ **do**
  # If S is empty, populate it with the first barcode, each bar initializing a new summand
  if $S == []$ **then**
    for $[b_l^M, d_l^M] \in \mathcal{B}(M|_l)$ **do**
      $B \leftarrow \{[b_l^M, d_l^M]\}$;
      S.append(B);
    end
  end
  # If S is not empty, process each bar in the current barcode
  else
    for $[b_l^M, d_l^M] \in \mathcal{B}(M|_l)$ **do**
      # Check whether it is in the image of the exact matching
      if $\exists B \in S$ *and* $[b, d] \in B$ *s.t.* $[b_l^M, d_l^M] = m([b, d])$ **then**
        B.append($[b_l^M, d_l^M]$); # If it is, attach the bar to the corresponding summand
      end
      # Otherwise initialize a new summand with the bar
      else
        $B \leftarrow \{[b_l^M, d_l^M]\}$;
        S.append(B);
      end
    end
  end
end
# For each summand in S characterized by a set of bars, build an approximate interval summand by computing candidate corners
for $B \in S$ **do**
  $\tilde{I}(B) \leftarrow$ ApproximateInterval(B);
end
Return $\tilde{M} := \bigoplus_{B \in S} \tilde{I}(B)$;

We now describe the algorithm ApproximateInterval, which is used at the end of Algorithm 1. Our algorithm ApproximateInterval is defined in two steps:

1. first, we *label* birthpoints and deathpoints to identify the facets of I (Algorithm 3),
2. then, we use these labels to compute the interval *corners* (Algorithm 4).

Once corners are computed, one can use them as approximate critical points and in order to output a discretely presented interval module with Equation (1).

Algorithm 2: ApproximateInterval

Input: Set of bars $B = \{[b_l, d_l] : l \in L_B \subseteq L\}$
Output: Discretely presented interval module $\tilde{I}$ parameterized by a list of birth and death corners
labs $\leftarrow$ LabelEndpoints(B);
$C_B^L(I), C_D^L(I) \leftarrow$ ComputeCorners(B, labs);
Return $\tilde{I}(B) := \mathbf{Ind}\left(\bigcup_{c \in C_B^L(I)} \bigcup_{c' \in C_D^L(I)} R_{c,c'}\right)$;

We first describe LabelEndpoints. The core idea of the algorithm is, for a given bar in I, to look at its corresponding L-surrounding set (see item (4) in Definition 3.2). If there exists a hyperplane H such that all endpoints in this surrounding set belong to H, we identify H as a facet, and we label the bar with the codirection of H.

Algorithm 3: LabelEndpoints

Input: Set of bars $B = \{[b_l, d_l] : l \in L_B \subseteq L\}$
Output: List labs of labels for each endpoint in B
for $l \in L_B$ **do**
  labs$(b_l) \leftarrow []$;
  labs$(d_l) \leftarrow []$;
end
for $l \in L_B$ **do**
  if $\exists i \in [\![1, n]\!]$ *and* $c_i \in \mathbb{R}$, *such that* $\forall l' \in L_l$, $(b_{l'})_i = c_i$ **then**
    for $l' \in L_l$ **do**
      labs$(b_{l'})$.append(i, c_i);
    end
  end
  if $\exists i \in [\![1, n]\!]$ *and* $c_i \in \mathbb{R}$, *such that* $\forall l' \in L_l$, $(d_{l'})_i = c_i$ **then**
    for $l' \in L_l$ **do**
      labs$(d_{l'})$.append(i, c_i);
    end
  end
end
Return labs;

Note that endpoints can have zero or more than one label. For instance, an endpoint that belongs to the intersection of several facets might have multiple labels. However, if several labels are identified, they must be associated to different dimensions. See Figure 5 for examples of label assignments when the underlying interval module has rectangle support.

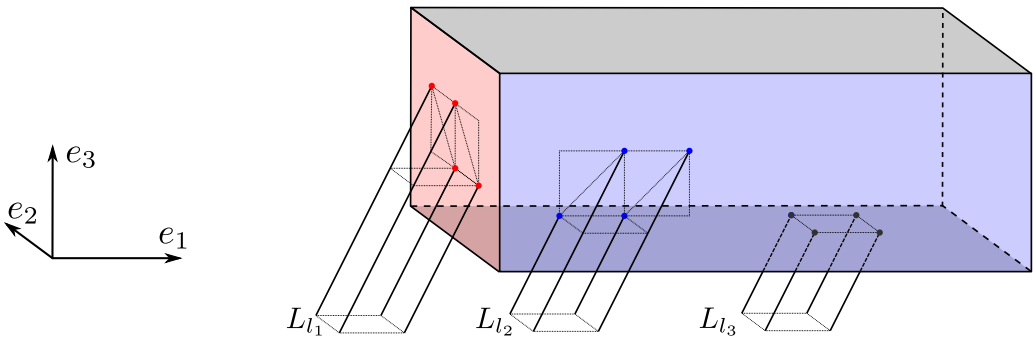

Figure 5: Example of birthpoint labelling for an interval module I with rectangle support with three surrounding sets of lines $L_{l_1}, L_{l_2}, L_{l_3}$ associated to three lines l_1, l_2, l_3. The labels of l_1, l_2, l_3 that are identified correspond to the red, blue and grey colored facets of I respectively.

Remark 3.6. Detecting facets with 2^{n-1} endpoints sharing the same labels is not necessarily optimal. For instance, a rectangle module can be recovered with only three bars passing through it in $\mathbb{R}^3$. However, it does allow for simpler proofs.

Finally, we describe ComputeCorners. The core idea of the algorithm is to use the labels identified by LabelEndpoints to compute candidate corners in the following way: if all birthpoints (resp. deathpoints) in a surrounding set

have at least one associated facet, i.e., have a non-empty list of labels, then a candidate corner can be defined using the minimum (resp. maximum) of all birthpoints (resp. deathpoints) coordinates. We only present the pseudo-code for birthpoints since the code for deathpoints is symmetric and can be obtained by replacing min by max and $-\infty$ by $+\infty$.

Algorithm 4: ComputeCorners

Input 1: Set of bars $B = \{[b_l, d_l] : l \in L_B \subseteq L\}$
Input 2: List labs of labels for each endpoint in B
Output: List of birth corners C_B
$C_B \leftarrow []$;
for $l \in L_B$ **do**
  $B_{L_l} \leftarrow \{b_{l'} : l' \in L_l \cap L_B\}$; # Note that $B_{L_l} \subseteq B$ by construction
  # Check whether all birthpoints in the surrounding set belong to the support K of the critical points of the underlying module
  if $B_{L_l} \subseteq K$ **then**
    # Compute birth corner if all the birthpoints are labelled
    if $\text{labs}(b) \neq \emptyset, \forall b \in B_{L_l}$ **then**
      $\{(j, c_j) : j \in \mathcal{J}\} \leftarrow \bigcup_{b \in B_{L_l}} \text{labs}(b)$; # $\mathcal{J} \subseteq [\![1, n]\!]$ is the corresponding set of codirections
      Define $C^l \in \mathbb{R}^n$ as
        - $(C^l)_j = c_j$ if $j \in \mathcal{J}$
        - $(C^l)_j = \min\left\{(b_{l'})_j : l' \in L_l \cap L_I\right\}$ otherwise
      C_B.append(C^l);
    end
    # If the birthpoints are not all labeled, we simply keep the birthpoints themselves as corners
    else
      for $l' \in L_l \cap L_B$ **do**
        C_B.append($b_{l'}$);
      end
    end
  end
  # If some birthpoints are not in K, they must correspond to infinite facets
  else
    Assert $B_{L_l} \cap K^{2\delta} \backslash K \neq \emptyset$;
    Assert $\text{labs}(b) \neq \emptyset$ for all $b \in B_{L_l}$;
    $\{(j, c_j) : j \in \mathcal{J}\} \leftarrow \bigcup_{b \in B_{L_l}} \text{labs}(b)$; # The cardinality of the set of codirections $\mathcal{J} \subsetneq [\![1, n]\!]$ must be strictly less than n
    Define $C^l \in \mathbb{R}^n$ as
      - $(C^l)_j = c_j$ if $j \in \mathcal{J}$
      - $(C^l)_j = -\infty$ otherwise
    C_B.append(C^l);
  end
end
Return C_B;

Note that, by construction, the candidate corners computed by ComputeCorners are all finite, pseudo or infinite L-corners, as defined in Definition 3.4. Now that we have defined how to compute an approximation, in the following sections, we will now show that the approximate multipersistence module $\tilde{M}$ provided by Algorithm 1 is a good approximation of the underlying module M when M is interval decomposable.

Complexity. Computing the L-fibered barcode $\mathcal{FB}(M)_L$ on a simplicial complex, as well as assigning the corresponding bars to their associated summands in the decomposition of M, can be done with the vineyard algorithm and matching [CSEM06] with complexity $O(N^3 + |L| \cdot N \cdot T)$, where N is the number of simplices in the simplicial complex, and T is the maximal number of transpositions required to update the single-parameter filtrations corresponding to the consecutive lines in L. In the worst case scenario, $T = N^2$. Note that T usually decreases as $|L|$ increases, and that this computation can be easily parallelized (see Section 7).

Now, adding the complexities of Algorithms 3 and 4, the total complexity of Algorithm 1 is

$$O(N^3 + |L| \cdot N \cdot T + |L| \cdot n \cdot 2^{n-1}).$$

Of importance, the dependence on n is much better than the (exact) decomposition algorithm proposed in [DX22] whose complexity is $O(N^{n(2\omega+1)})$. It is also better than Rivet [LW15] (which works only when $n = 2$), whose complexity is

$O(N^3\kappa + (N + \log\kappa)\kappa^2)$, where $\kappa = \kappa_x\kappa_y$ is the product of unique x and y coordinates in the support of the module. Moreover, our complexity can be controlled by the number of lines, which is user-dependent. Again, we illustrate this useful property in Section 7.

3.4. Endpoint properties

In this section, we prove a few preliminary results about endpoints of interval modules, that will turn out very useful for quantifying the error made by Algorithm 1 when approximating the true endpoints of a multipersistence module with L-corners, as we do in Sections 4 and 5. Roughly speaking, we prove in this section that the location of endpoints is related to the rectangle hull of other endpoints corresponding to lines in some specific surrounding sets.

Definition 3.7. Let $S \subseteq \mathbb{R}^n$. The *rectangle hull* of S, denoted by $\mathrm{recthull}[S]$, is defined with

$$\mathrm{recthull}[S] := \left\{ x \in \mathbb{R}^n : \forall i \in [\![1, n]\!], \min_{s\in S} s_i \leq x_i \leq \max_{s\in S} s_i \right\} = R_{\wedge S, \vee S},$$

where $(\wedge S)_i := \min_{s\in S} s_i$ and $(\vee S)_i := \max_{s\in S} s_i$.

Lemma 3.8 (Endpoints bound)**.** *Let I be an n-interval module. Let $\delta > 0$, K be a compact set of $\mathbb{R}^n$ and L be a δ-grid of K. Let $x \in K^\delta$, l_x be the diagonal line passing through x and $d_x^I \in U[I]$ be the associated deathpoint. Finally, let $L_{x,\delta} := \{l \in L : d_\infty(x, l) \leq \delta$ and l_x, l are δ-comparable$\}$, which is non-empty since L fills $K^{2\delta}$. Assume that for any line l in $L_{x,\delta}$, one has $\mathrm{supp}(I) \cap l \neq \emptyset$, and let $D^I_{x,\delta}$ be the set of the associated deathpoints: $D^I_{x,\delta} = \{d_l^I : l \in L_{x,\delta}\}$. Then, d_x^I belongs to the rectangle hull of $D^I_{x,\delta}$: one has $d_x^I \in \mathrm{recthull}[D^I_{x,\delta}]$.*

Similarly, if $b_x^I \in L[I]$ is a birthpoint, then $b_x^I \in \mathrm{recthull}[B^I_{x,\delta}]$ where $B^I_{x,\delta}$ is the set of birthpoints associated to $L_{x,\delta}$.

In other words, the endpoints of an interval module always belong to the rectangle hull of the endpoints associated to neighbouring lines. See Figure 6 for an illustration.

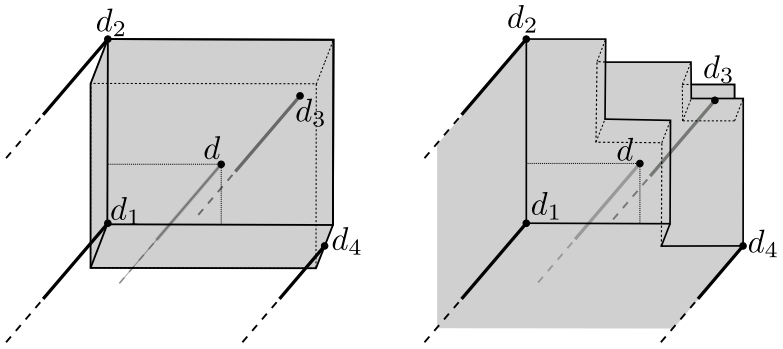

Figure 6: Example of deathpoint bound in $\mathbb{R}^3$, with $d \in U[I]$, and $D^I_{x,\delta} = \{d_1, d_2, d_3, d_4\}$. **(Left)** Rectangle hull of the deathpoints $D^I_{x,\delta}$. **(Right)** Upper-boundary $U[I]$.

Proof. We first prove the result for deathpoints. Note that the result is trivially satisfied if d_x^I and the deathpoints in $D^I_{x,\delta}$ are infinite, so we assume that they are finite in the following. Let $j \in [\![1, n]\!]$ be an arbitrary dimension. To alleviate notations, we let $d := d_x^I$. In order to prove the result, we will show that there exist two deathpoints $\underline{d}$ and $\overline{d}$ associated to consecutive lines of $L_{x,\delta}$ such that $\underline{d}_j \leq d_j \leq \overline{d}_j$.

Construction of $\underline{d}, \overline{d}$. Let H_j be the hyperplane $H_j = d + e_j^\perp$. Since L fills $K^{2\delta}$ and $x \in K^\delta$, there exists a diagonal line $l \in L$ such that $d_\infty(x, l) \leq \delta/2$. Moreover, since l and l_x (the line passing through x and d) are both diagonal, one has $d_\infty(d, l) = d_\infty(x, l) \leq \delta/2$. Let $\pi_l(d) \in l$ be the projection of d onto l that achieves $d_\infty(d, l)$, and let $d^j := l \cap H_j$. See Figure 7 for an illustration of these objects.

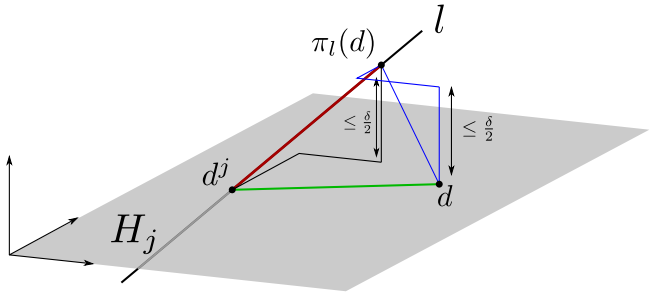

Figure 7: Illustration of H_j, d, l, d^j.

Since d^j and d belong to H_j, they have the same j-th coordinate: $d^j_j = d_j$. Moreover, both d^j and $\pi_l(d)$ belong to the diagonal line l, hence they are comparable, and $\|d^j - \pi_l(d)\|_\infty = |(d^j - \pi_l(d))_i|$ for any $i \in [\![1, n]\!]$. Then, one has $\|d^j - d\|_\infty \leq \|d^j - \pi_l(d)\|_\infty + \|\pi_l(d) - d\|_\infty = |(d^j - \pi_l(d))_j| + \|\pi_l(d) - d\|_\infty = |(d - \pi_l(d))_j| + \|\pi_l(d) - d\|_\infty \leq 2\|\pi_l(d) - d\|_\infty \leq \delta$.

Let $d^+ = d^j + \delta \sum_{j \in \mathcal{J}''} e_j$ and $d^- = d^j - \delta \sum_{j \in \mathcal{J}'} e_j$, where

$$\mathcal{J}' = \left\{ i \in [\![1, n]\!] \backslash \{j\} : d_i < d^j_i \right\} \text{ and } \mathcal{J}'' = \left\{ i \in [\![1, n]\!] \backslash \{j\} : d_i > d^j_i \right\}.$$

By construction, one has $d^- \leq d \leq d^+ \in H_j$ and $\|d^+ - d\|_\infty, \|d^- - d\|_\infty \leq \delta$. Since l and the diagonal lines $\underline{l}$ and $\overline{l}$ passing through d^- and d^+ respectively are δ-consecutive, and since $x \in K^\delta$, the projections of x onto $\underline{l}$ and $\overline{l}$ are in $K^{2\delta}$, and thus $\underline{l}, \overline{l}$ must belong to $L_{x,\delta}$, as by construction l_x is δ-comparable with the diagonal lines $\underline{l}$ and $\overline{l}$. Let $\overline{d} := d^I_{\underline{l}} \in \underline{l}$ and $\underline{d} := d^I_{\overline{l}} \in \overline{l}$ be their deathpoints (which exist by assumption).

Proof of inequalities. We now show that $\overline{d}_j \geq d_j \geq \underline{d}_j$. We start with the second inequality. Since d^+ and $\underline{d}$ are one the same diagonal line, they are comparable. Furthermore, if one had $d^+ < \underline{d}$ by contradiction, then the induced rectangle $R_{d,\underline{d}}$ would not be flat since $d \leq d^+ < \underline{d}$, which would contradict Remark 2.20. As a consequence, $d^+ \geq \underline{d}$. Taking the j-th coordinate yields $d_j = d^+_j \geq \underline{d}_j$. The first inequality holds using the same arguments.

This proof applies straightforwardly to birthpoints by symmetry. □

Using Lemma 2.22, one can generalize Lemma 3.8 above to the case where some lines in $L_{x,\delta}$ have an empty intersection with $\mathrm{supp}(I)$, and then define a common location for all endpoints that belong to the convex hull of the same L-surrounding set, as we do in the following proposition.

Proposition 3.9. *Let I be an n-interval module. Let $\delta > 0$, K be a compact set of $\mathbb{R}^n$ and L be a δ-grid of K. Let $l \in L$ such that $|L_l| = 2^{n-1}$. Then, there exists a set B_l (resp. D_l) such that for any $x \in \mathrm{conv}(L_l) \cap L[I]$ (resp. $\mathrm{conv}(L_l) \cap U[I]$), one has either $x \in B_l$ (resp. D_l) or $\|b^I_x - d^I_x\|_\infty \leq \delta$, where B_l (resp. D_l) is a rectangular set in $\mathbb{R}^n$ that can be constructed from the birthpoints $(b^I_{l'})_{l' \in L_l}$ (resp. deathpoints $(d^I_{l'})_{l' \in L_l}$). Moreover, one has*

1. *$\sup\{t \geq 0 : x + t \cdot \mathbf{1} \in B_l\} \leq \delta$ (resp. $\sup\{t \geq 0 : x + t \cdot \mathbf{1} \in D_l\} \leq \delta$), and*
2. *B_l (resp. D_l) is included in a ball of radius δ: there exists x_l such that B_l (resp. D_l) $\subseteq \{y \in \mathbb{R}^n : \|y - x_l\|_\infty \leq \delta\}$.*

Proof. We first construct B_l and D_l, and then we will show items (1) and (2).

Definition of B_l, D_l. Let first assume that x is in the interior of $\mathrm{conv}(L_l)$, that we denote with $\mathrm{conv}(L_l)^\circ$. Note that if there is a line l_0 that is δ-comparable to l_x, and such that $\mathcal{B}(I\big|_{l_0}) = \emptyset$, then by Lemma 2.22, one immediately has $\|b^I_x - d^I_x\|_\infty \leq \delta$. Hence, we now assume that the barcodes along any line that is δ-comparable to l_x is not empty, which means that the hypotheses of Lemma 3.8 are satisfied for x. Now, remark that since L is a grid, if one is able to find a line l' in L whose intersections with hyperplanes associated to the canonical axes of $\mathbb{R}^n$ are δ-close to x, then, since x is in the interior of an L-surrounding set L_l, l' must belong to that surrounding set L_l as well. More formally, one has that, for any line $l' \in L$,

$$d_\infty(x, l' \cap H_i) \leq \delta \implies l' \in L_l, \quad \text{where } H_i = \{y \in \mathbb{R}^n : y_i = x_i\}.$$

This ensures that $L_{x,\delta}$ (see Lemma 3.8) is included in L_l for any $x \in \mathrm{conv}(L_l)^\circ$, and thus that we can safely define

$$D_l := \overline{\bigcup_{x \in \mathrm{conv}(L_l)^\circ} \mathrm{recthull}[D^I_{x,\delta}]} \quad \text{and} \quad B_l := \overline{\bigcup_{x \in \mathrm{conv}(L_l)^\circ} \mathrm{recthull}[B^I_{x,\delta}]}.$$

Note that B_l and D_l depend only on the endpoints of the lines in L_l (since $L_{x,\delta} \subseteq L_l$ for all $x \in \mathrm{conv}(L_l)^\circ$), and that $d^I_x \in D_l$ and $b^I_x \in B_l$ for any $x \in \mathrm{conv}(L_l)^\circ$ by Lemma 3.8. Furthermore, if x is in the closure of $\mathrm{conv}(L_l)$, the previous statements still hold since D_l and B_l are closed sets. We now show that B_l and D_l satisfy items (1) and (2).

Proof of (1). By applying Lemma 3.8 and its proof for dimension $j = n$ to all $x \in \mathrm{conv}(L_l)$, there exist deathpoints $\overline{d}^n$ and $\underline{d}_n$ that satisfy $(\overline{d}_n)_n = \sup_{d \in D_l} d_n$ and $(\underline{d}_n)_n = \inf_{d \in D_l} d_n$ and $(\underline{d}_n)_n \leq x_n \leq (\overline{d}_n)_n$ for all $x \in \mathrm{conv}(L_l)$. Moreover, these points are located on the lines l and $l + \sum_{1 \leq j < n} \delta e_j$, which are are δ-consecutive. Thus, applying Lemma 2.22 on this pair of line, we end up with D_l having a diagonal smaller than δ. The same goes for birthpoints.

Proof of (2). Note first that

$$D_l \subseteq \left\{ y \in \mathbb{R}^n : \forall i \in [\![1, n]\!], \min_{l' \in L_l} (d^I_{l'})_i \leq y_i \leq \max_{l' \in L_l} (d^I_{l'})_i \right\},$$

and that, for any pair of lines $l_1, l_2 \in L_l$, there is a vector $\overrightarrow{v}$ such that $l_2 = l_1 + \overrightarrow{v}$ with $\left\|\overrightarrow{v}\right\|_\infty \leq \delta$. Thus, one has $\overrightarrow{u} := \overrightarrow{v} + \boldsymbol{\delta} \geq 0$ and $\left\|\overrightarrow{u}\right\|_\infty \leq 2\delta$. Moreover, l_2 can also be written as $l_2 = l_1 + \overrightarrow{u}$; thus any two lines l_1 and l_2 in L_l are 2δ-consecutive. Now, for an arbitrary dimension $i \in [\![1, n]\!]$, by applying Lemma 2.22 on the pair of lines $l_1, l_2 \in L_l$ such that $(d^I_{l_1})_i = \min_{l' \in L_l}(d^I_{l'})_i$ and $(d^I_{l_2})_i = \max_{l' \in L_l}(d^I_{l'})_i$, one has that the difference of the i-th coordinates between any two points in D_l is upper bounded by 2δ. Since this is true for any i, item (2) is true. The same goes for birthpoints. □

Remark 3.10. These bounds are sharp in dimension $n \geq 3$:

1. (1) Let $\delta > 0$ and I be the interval with support $\mathrm{supp}(I) = \{x \in \mathbb{R}^n : \langle x, \mathbf{1}\rangle \geq \delta\}$. Let l be the diagonal line passing through 0. Then, one has

$$b^I_l = \left(\frac{\delta}{n}, \ldots, \frac{\delta}{n}\right), \quad b^I_{l+\delta e_i} = (0, \ldots, \delta, \ldots, 0), \quad \ldots, \quad b^I_{l+\delta e_1 + \cdots + \delta e_{n-1}} = \left(\frac{2}{n}\delta, \ldots, \frac{2}{n}\delta, -\frac{(n-2)}{n}\delta\right).$$

   In particular, one has $\left\|b^I_{l+\delta e_1} - b^I_{l+\delta e_2}\right\|_\infty = \delta$ using the lines $l + \delta e_1$ and $l + \delta e_2$ that both belong to L_l.

2. (2) Let I be an interval whose support has a facet F of codirection different than n, and let l be a diagonal line such that $\left\{b^I_{l'} : l' \in L_l\right\} \subseteq F$. Then the radius of the ball containing B_l is exactly 2δ, as illustrated with the red and blue facets in Figure 5.

4. Exact reconstruction

In this section, we show that, under some assumptions on the family of lines that are used and on the underlying multipersistence module M, our approximation $\tilde{M}$ computed by Algorithm 1 cannot be separated from M by both the interleaving and bottleneck distances (see Proposition 4.4 and Corollary 4.6.1). Roughly speaking, when M is interval decomposable, we need assumptions that ensure that all the facets of the summands of M can be identified with associated labels by Algorithm 3. This means that the facets have to be large enough with respect to the spacing between the lines in order to make sure that lines in surrounding sets can reach the same common facets. In addition to this, one also has to ensure that taking the minimum (resp. maximum) of birthpoints (resp. deathpoints) in surrounding sets, as prescribed by Algorithm 4, induces a corner that belongs indeed to the support of the multipersistence module. This means that the support of the module cannot contain holes of small size, such that a line could go through the hole and avoid the support, while all surrounding lines would intersect the support, which would lead to a fake corner.

We now characterize those interval modules that satisfy the aforementioned informal assumptions. Given a size parameter $\delta > 0$, these interval modules form a subclass of the family of discretely presented interval modules, that we call the *δ-discretely presented interval modules*.

Definition 4.1 (*δ*-discretely presented interval module)**.** Let $K \subseteq \mathbb{R}^n$ be a compact rectangle of $\mathbb{R}^n$, and let I be a discretely presented interval module. Given $\delta > 0$, we say that I is *δ-discretely presented* in K if:

1. (Large facets) for each point $x \in L[I]$ (resp. $U[I]$) there exists, for each facet F containing x, an $(n-1)$-hypercube Q^x_F of side length 2δ such that $x \in Q^x_F$ and $Q^x_F \subseteq F$;
2. (Large holes) if there exists a diagonal line l such that $l \cap \mathrm{supp}(I) = \emptyset$, then there exists an n-hypercube R of side length δ containing $\mathbf{0}$ such that for any line l' in $l + R$, one has $l' \cap \mathrm{supp}(I) = \emptyset$;
3. (Locally small complexity) any ∞-ball of radius δ, i.e., any set $B_\delta(x) := \{y \in \mathbb{R}^n : d_\infty(x, y) \leq \delta\}$ for some $x \in \mathbb{R}^n$, intersects at most one facet in $L[I]$ (resp. $U[I]$) of any fixed codirection;
4. (Compact description) each facet of I has a non-empty intersection with K.

Assumptions 1 and 2 correspond to the assumptions mentioned at the beginning of the section, while Assumptions 3 and 4 ensure that surrounding sets of lines can detect at most one facet associated to a given codirection at a time, and that critical points of I are all included in a rectangle respectively.

Remark 4.2. One might wonder whether Assumption 2 and Assumption 3 are redundant with Assumption 1. In other words, one might wonder whether it is actually possible to define an interval module with large facets and small holes, or with large facets that can share the same codirection and lie close to each other at the same time. Even though this seems to be impossible when $n = 2$ (indicating that Assumption 2 and Assumption 3 might indeed be redundant with Assumption 1), it can definitely happen in dimension $n \geq 3$, as Figure 8 shows.

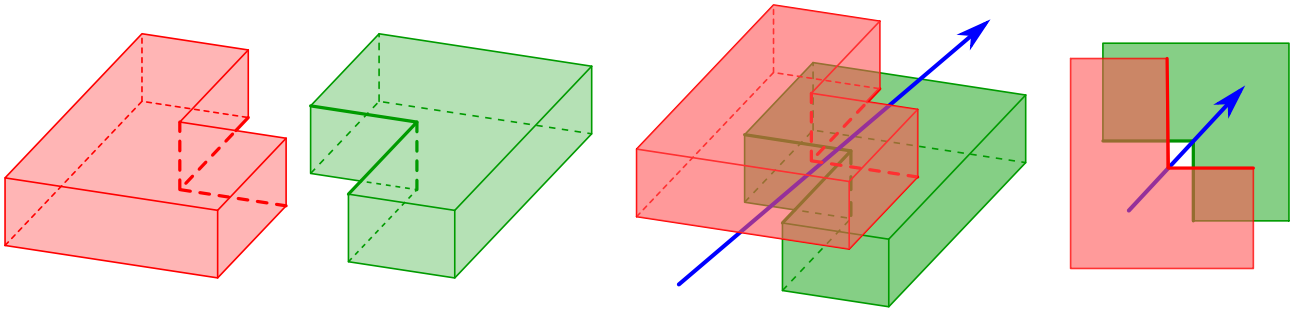

Figure 8: Example of interval module in dimension $n = 3$ with large facets, small holes and some facets with the same codirection close to each other. The support of the module can be constructed by taking the (closed) red and (open) green L-shaped sets on **(Left)**, and glue them together as shown in **(Middle)**. While arbitrarily large facets can be created using this construction, the resulting interval always contains a small hole and large facets of same codirection that are close to each other. Because of this, it is possible to find a (blue) diagonal line that goes through the support without intersecting it, while lines in its surrounding set will detect some facets. **(Right)** View of the interval from the top showing the hole and the spatially close facets (showed in bold font). This is an example where Assumptions 1 and 4 of Definition 4.1 are satisfied, while Assumptions 2 and 3 are not.

The main advantage of δ-grids and δ-discretely presented modules is that they ensure that Algorithm 3 can identify every single facet with a corresponding label.

Lemma 4.3. *Let $\delta > 0$ and K be a compact rectangle of $\mathbb{R}^n$. Let I be a δ-discretely presented interval module in K, and let L be a δ-grid of $K^{2\delta}$. Then, there is a bijection between the facets of I and the labels identified by Algorithm 3.*

Proof. We first prove the result for birthpoints and facets of $L[I]$.

Let F be a facet of $L[I]$. Let $l_F \in L$ be a diagonal line intersecting F, and $b_F \in \mathbb{R}^n$ be the associated birthpoint. By Definition 4.1, item (1), there exists an $(n-1)$-hypercube $Q_F^{b_F} \subseteq F$ of side length 2δ such that $b_F \in Q_F^{b_F}$. This ensures that for any dimension i that is not in the codirection: $i \in [\![1, n]\!] \backslash \mathrm{codir}(F)$, one has either $b_F + \delta e_i \in Q_F^{b_F}$ or $b_F - \delta e_i \in Q_F^{b_F}$. Since L is a δ-grid of $K^{2\delta}$, and since $Q_F^{b_F}$ is an $(n-1)$-hypercube, there exists a line $l_0 \in L$ such that l_F belongs to the surrounding set L_{l_0}, and such that the birthpoints corresponding to the lines in L_{l_0} are all in $Q_F^{b_F}$. This means that $\mathrm{codir}(F)$ is detected as a label of b_F by Algorithm 3.

Reciprocally, assume there exists a line $l_0 \in L$ such that all birthpoints associated to the lines in the surrounding set L_{l_0} share a coordinate along dimension $i \in [\![1, n]\!]$, so that i is a label detected by Algorithm 3. Then, the set of birthpoints $B_{L_{l_0}}$ has a minimal element, and thus its convex hull $\mathrm{conv}(B_{L_{l_0}})$ is in $L[I]$. Since $\mathrm{conv}(B_{L_{l_0}})$ is an $(n-1)$-hypercube of codirection i, it must be associated to a facet of $L[I]$ of codirection i as well.

The proof extends straightforwardly for deathpoints. □

Now that we have proved that all facets can be detected with δ-grids and δ-discretely presented modules, we can state our first main result, which claims that it is possible to *exactly* recover the underlying module under the same assumptions.

Proposition 4.4 (Exact recovery)**.** *Let $\delta > 0$ and $K = R_{\alpha,\beta}$ be a compact rectangle of $\mathbb{R}^n$, where $\alpha \leq \beta$. Let I be a δ-discretely presented interval module in K, and let L be a δ-grid of $K^{2\delta}$. Let $C_B^L(I)$ and $C_D^L(I)$ be the L-birth and death corners of I computed by Algorithm 4, and let $\tilde{I} = \mathbf{Ind}\left(\bigcup_{c \in C_B^L(I)} \bigcup_{c' \in C_D^L(I)} R_{c,c'}\right)$ be the approximation computed by Algorithm 2. Then, one has*

$$d_I(I, \tilde{I}) = d_b(I, \tilde{I}) = 0. \tag{5}$$

Proof. As interval modules are characterized by their support, it is enough to show that $\overline{\mathrm{supp}(I)} = \overline{\mathrm{supp}(\tilde{I})}$. In the following, we thus assume that $\mathrm{supp}(I)$ is closed in $\overline{\mathbb{R}}^n$.

We first show the inclusion $\mathrm{supp}(\tilde{I}) \subseteq \mathrm{supp}(I)$**.** More specifically, we have to prove that the (finite, pseudo and infinite) L-corners computed by Algorithm 4 all belong to $\mathrm{supp}(I)$. A key argument that we will use several times comes from the following lemma, which allows for a local control of the boundary of $\mathrm{supp}(I)$ using the hyperplanes associated to specific L-corners.

Lemma 4.5. *Let b be a birthpoint (resp. deathpoint) of I in K^δ, and $l_0 \in L$ be the line such that $b \in \mathrm{conv}(L_{l_0})$ (this line exists since L fills $K^{2\delta}$). Then, one has the following:*

1. *for any facet F of $L[I]$ (resp. $U[I]$) containing b, there exists a line $l_F \in L_{l_0}$ such that $b^I_{l_F} \in F$ (resp. $d^I_{l_F} \in F$).*

2. *for any dimension i, there exists at most one facet of codirection i intersecting the set of birthpoints (resp. deathpoints)* $\left\{b^I_l : l \in L_{l_0}\right\}$ *(resp.* $\left\{d^I_l : l \in L_{l_0}\right\}$*.*

3. *let* $b'_{L_{l_0}}$ *(resp.* $d'_{L_{l_0}}$*) be the the pseudo or finite L-corner generated by* L_{l_0}*. Then, one has:*

$$\operatorname{conv}(L_{l_0}) \cap L[I] \cap K^{2\delta} \subseteq \bigcup_{i \in \operatorname{codir}(b')} \left\{x \in \mathbb{R}^n : x_i = b'_i\right\}$$

$$\textit{(resp. } \operatorname{conv}(L_{l_0}) \cap U[I] \cap K^{2\delta} \subseteq \bigcup_{i \in \operatorname{codir}(d')} \left\{x \in \mathbb{R}^n : x_i = d'_i\right\} \textit{).}$$

Proof. We only show the result for birthpoints since the arguments for deathpoints are the same. Let $b \in L[I]$ be a birthpoint in K^δ.

Proof of (1). Let F be a facet containing b. According to Definition 4.1, item (1), there exists an $(n-1)$-hypercube Q^b_F of side length 2δ such that $Q^b_F \subseteq F$ and $b \in Q^b_F$. Since L is a grid, there exists a line $l \in L$ with $d_\infty(b, l) < \delta$ intersecting Q^b_F. Now, since $b \in \operatorname{conv}(L_{l_0})$, one has $d_\infty(l \cap H_F, L_{l_0} \cap H_F) < \delta$, where H_F is the hyperplane containing F; thus, $l \in L_{l_0}$ (the argument is the same than in the proof of Proposition 3.9, first paragraph).

Proof of (2). By Proposition 3.9, item (2), the birthpoints associated to lines of L_{l_0} are all contained in a ball of radius δ. Thus, the unicity of the facets with given codirection comes straightforwardly from Definition 4.1, item (3).

Proof of (3). Note that the birthpoint b is obviously included in the facets of $L[I]$ that contain it, which is a subset of the facets associated to the birthpoints of the lines in L_{l_0}. Now, as Lemma 4.3 ensures that the birthpoints associated to lines in L_{l_0} are correctly labelled, the pseudo or finite L-corner generated by L_{l_0} must be on the intersection of the facets containing b. This ensures that

$$b \in \bigcup_{i \in \operatorname{codir}(b')} \left\{x \in \mathbb{R}^n : x_i = b'_i\right\}.$$

Since these arguments do not depend on $b \in \operatorname{conv}(L_{l_0})$, the result follows. □

Now that we have Lemma 4.5, we can prove that finite, pseudo and infinite L-corners belong to $\operatorname{supp}(I)$. We will prove the results for birth corners, but the arguments for death corners are exactly the same.

Finite and pseudo corners. Let b be a finite or pseudo L-birth corner, associated to a set of consecutive lines L_{l_0} for some line $l_0 \in L$. By assumption, each birthpoint b^I_l, for $l \in L_{l_0}$, is nontrivial; and thus any birthpoint in $\operatorname{conv}(L_{l_0})$ is nontrivial as well, using Definition 4.1, item (2). Let $l \in \operatorname{conv}(L_{l_0})$ be the diagonal line passing through b.
Using Lemma 4.5, one has:

$$b^I_l \in \operatorname{conv}(L_{l_0}) \cap L[I] \cap K^{2\delta} \subseteq \left(\bigcup_{i \in \operatorname{codir}(b)} \{x \in \mathbb{R}^n : x_i = b_i\} \right) \cap l = \{b\}.$$

Thus $b = b^I_l$ and $b \in \operatorname{supp}(I)$.

Infinite corners. Let b be an infinite L-birth corner, and let b' be the corresponding minimal pseudo L-birth corner, associated to a set of consecutive lines L_{l_0} for some line $l_0 \in L$. We will show that, if j is a free coordinate of b', i.e., if $j \in \operatorname{dir}(b')$, then $b'_j < \alpha_j$ (recall that K is the rectangle $R_{\alpha,\beta}$). The reason we want to prove such inequalities is that they directly lead to the result. Indeed, if $b'_j < \alpha_j$ for any $j \in \operatorname{dir}(b')$, then $b' - t\sum_{j \in \operatorname{dir}(b')} e_j$ belongs to $L[I]$ for any $t > 0$, since otherwise the line $\{b' - t\sum_{j \in \operatorname{dir}(b')} e_j : t > 0\}$ would have to intersect a facet $F \subseteq L[I]$ of codirection j for some $j \in \operatorname{dir}(b')$, which would not intersect K, contradicting Definition 4.1, item (4).

Let $j \in \operatorname{dir}(b')$ be a free coordinate. By contradiction, assume that $b'_j \geq \alpha_j$, and let b^j denote the pseudo L-corner generated by $L_{l_0 - \delta e_j}$. In particular, this means that, for any $l \in L_{l_0}$, $l - \delta e_j \in L$ and $L_{l-\delta e_j} \subseteq L$ since L fills $K^{2\delta}$. Now, if for every line $l \in L_{l_0}$ such that $l = l_0 + \overrightarrow{v}$ with $\overrightarrow{v}_j = 0$, one has that b^I_l and $b^I_{l-\delta e_j}$ are on the same facets, then one has $b^I_{l-\delta e_j} = b^I_l - \delta e_j$, and the pseudo corner b^j is equal to $b' - \delta e_j$ by construction, as per Algorithm 4. Moreover, one has $b^j = b' - \delta e_j \leq b'$, contradicting the fact that b' is minimal. Hence, there is at least one line $l \in L_{l_0}$, $l = l_0 + \overrightarrow{v}$ with $\overrightarrow{v}_j = 0$, such that b^I_l and $b^I_{l-\delta e_j}$ are not on the same facets, in other words, there exists a facet F_j of $L[I]$ of codirection j that intersects the (half-open) segment $[b^I_l - \delta e_j, b^I_l)$. In order to locate that facet more precisely, we will prove the following lemma:

Lemma 4.6. *For any* $i \in [\![1, n]\!]$ *and* $s, t \in \mathbb{R}$ *such that* $s < t$*, one has* $(b^I_{l-te_i})_i \leq (b^I_{l-se_i})_i$.

Proof. Without loss of generality, assume $s = 0$. Since $b_l^I - te_i \in l - te_i$, it follows that $b_l^I - te_i$ and $b_{l-te_i}^I$ are comparable. Moreover, one must have $b_l^I - te_i \leq b_{l-te_i}^I$, otherwise one would have $b_l^I > b_l^I - te_i > b_{l-te_i}^I$, contradicting Remark 2.20. If the points are equal, i.e., $b_l^I - te_i = b_{l-te_i}^I$, then one has $(b_l^I)_i \geq (b_{l-te_i}^I)_i$. Otherwise, if $b_l^I - te_i < b_{l-te_i}^I$, then

$$\forall k \neq i,\ (b_{l-te_i}^I)_k > (b_l^I)_k.$$

Moreover, since b_l^I and $b_{l-te_i}^I$ cannot be comparable as per Remark 2.20 one must have $(b_{l-te_i}^I)_i \leq (b_l^I)_i$.

□

Let $H_j = \left\{x \in \mathbb{R}^n : x_j = c_j\right\}$ be the hyperplane associated to F_j. Then, by Lemma 4.6, one has

$$(b_{l-\delta e_j}^I)_j \leq c_j < (b_l^I)_j.$$

Since the lines l and $l - \delta e_j$ both belong to the surrounding set $L_{l_0-\delta e_j}$, it follows from Lemmas 4.3, and 4.5, item (3), that $\mathrm{codir}(b^j) \supseteq \mathrm{codir}(b') \cup \{j\}$. Moreover, since the facets of $L[I]$ associated to $\mathrm{codir}(b^j)$ are unique in a δ-ball around b^j, as per Definition 4.1, item (3), they all have a unique associated value c_i (corresponding to their associated hyperplanes).

Finally, we will show that $b^j \leq b'$. Let $i \in [\![1, n]\!]$ be an arbitrary dimension.

- If $i \in \mathrm{codir}(b')$, then $b_i^j = b'_i$.
- If $i \in \mathrm{codir}(b^j) \backslash \mathrm{codir}(b')$, then $b_i^j \in \left\{c_i, \min_{l \in L_{l_0-\delta e_j}} (b_l^I)_i\right\} \leq \min_{l \in L_{l_0}} (b_l^I)_i = b'_i$, with a strict inequality for $i = j$.
- If $i \in \mathrm{dir}(b^j) \subseteq \mathrm{dir}(b')$, then $b_i^j = \min_{l \in L_{l_0-\delta e_j}} (b_l^I)_i \leq \min_{l \in L_{l_0}} (b_l^I)_i = b'_i$.

Hence, one always has $b_i^j \leq b'_i$, and thus $b^j < b'$, which contradicts the fact that b' is minimal. Thus, one must have $b'_j < \alpha_j$.

We now show that $\mathrm{supp}(I) \subseteq \mathrm{supp}(\tilde{I})$**.** Let $x \in \mathrm{supp}(I)$. We will show that there exists an L-birth corner c such that $c \leq x$. Let $\mathcal{H}$ be the family of hyperplanes associated to the facets of $L[I]$. The corner c will be defined as the limit of a sequence of points $\{x^{(k)}\}_{k \in \mathbb{N}^*}$ in $\overline{\mathbb{R}}^n$, defined by induction with:

1. $x^{(1)} = \inf \{x - t \cdot \mathbf{1} : t \geq 0\} \cap \mathrm{supp}(I)$. Then, one has the two following possibilities:
   - either $x^{(1)} = -\infty$, and we let $c := x^{(1)}$.
   - or there exists a maximal subset of hyperplanes $\mathcal{H}^1 \subset \mathcal{H}$, $\mathcal{H}^1 \neq \emptyset$, such that $x^{(1)} \in \cap_{H \in \mathcal{H}^1} H =: H_1$. Let $\mathcal{J}^1 \subseteq [\![1, n]\!]$ be the set of free coordinates in H_1, i.e., those dimensions such that $j \in \mathcal{J}^1 \iff x^{(1)} - e_j \in H_1$.
2. $x^{(2)} = \inf \left\{x^{(1)} - t \cdot \sum_{j \in \mathcal{J}^1} e_j : t \geq 0\right\} \cap \mathrm{supp}(I)$. Then, one has the two following possibilities:
   - either $x^{(2)}$ is at infinity in H_1, i.e., $x_j^{(2)} = -\infty$ if $j \in \mathcal{J}^1$ and $x_j^{(2)} = x_j^{(1)}$ otherwise, and we let $c := x^{(2)}$.
   - or there exists a maximal subset of hyperplanes $\mathcal{H}^2 \supsetneq \mathcal{H}^1$ such that $x^{(2)} \in \cap_{H \in \mathcal{H}^2} H =: H_2$. Let $\mathcal{J}^2 \subseteq [\![1, n]\!]$ be the set of free coordinates in H_2, i.e., those dimensions such that $j \in \mathcal{J}^2 \iff x^{(2)} - e_j \in H_2$.
3. For $k \geq 3$, $x^{(k+1)} = \inf \left\{x^{(k)} - t \cdot \sum_{j \in \mathcal{J}^k} e_j : t \geq 0\right\} \cap \mathrm{supp}(I)$. Then, one has the two following possibilities:
   - either $x^{(k+1)}$ is at infinity in H_k, i.e., $x_j^{(k+1)} = -\infty$ if $j \in \mathcal{J}^k$ and $x_j^{(k+1)} = x_j^{(k)}$ otherwise, and we let $c := x^{(k+1)}$.
   - or there exists a maximal subset of hyperplanes $\mathcal{H}^{k+1} \supsetneq \mathcal{H}^k$ such that $x^{(k+1)} \in \cap_{H \in \mathcal{H}^{k+1}} H =: H_{k+1}$. Let $\mathcal{J}^{k+1} \subseteq [\![1, n]\!]$ be the set of free coordinates in H_{k+1}, i.e., those dimensions such that $j \in \mathcal{J}^{k+1} \iff x^{(k+1)} - e_j \in H_{k+1}$.

If this sequence stops at step one, i.e., $c = x^{(1)} = -\infty$, then every birthpoint of I is at $-\infty$, the only birth corner is $c = -\infty$, and one trivially has $c \leq x$. Hence, we assume in the following that c is obtained after at least one iteration of the sequence. Note that this sequence of points has length at most n. Let c^- and c be the penultimate and last elements of the sequence respectively, and let $\mathcal{J}^-$ be the set of free coordinates associated to c^-. By construction, one has:

$$c \leq c^- \leq \cdots \leq x^{(2)} \leq x^{(1)} \leq x.$$

We now show that c is indeed a birth corner. If c is finite, then it must belong to the intersection of n hyperplanes, and it is thus a finite birth corner. Hence, we assume now that c is not finite. We will construct a minimal pseudo birth corner from c^-, and show that c is its associated infinite birth corner. We will consider two different cases, depending on whether c^- is close to $K = R_{\alpha,\beta}$ or not. If $c^- \in K^\delta$, the filling property of L and the size of the facets of $L[I]$ ensure that c^- is itself a minimal pseudo birth corner, associated to c, which is thus an infinite birth corner. If $c^- \notin K^\delta$, then let $\overrightarrow{v} \in \mathbb{R}^n$ be a vector that pushes back c^- into K^δ, i.e., such that, for any dimension $i \in \mathcal{J}^-$, one has

$$\alpha_i - \delta \leq (c^- + \overrightarrow{v})_i < \alpha_i,$$

and $\overrightarrow{v}_i = 0$ if $i \notin \mathcal{J}^-$. Let S be the segment $[c^-, c^- + \overrightarrow{v}]$. We have the two following cases:

1. Assume $S \subseteq L[I]$. Then $c^- + \overrightarrow{v} \in \mathrm{supp}(I) \cap K^\delta$, and there exists a line $l \in L$ such that $c^- + \overrightarrow{v} \in \mathrm{conv}(L_l)$. Let c^l be the pseudo birth corner associated to L_l. Since one has $c^l_j < \alpha_j$ for any dimension $j \in \mathcal{J}^-$, it follows that $\mathcal{J}^- \subseteq \mathrm{dir}(c^l)$. Furthermore, since $c^- + \overrightarrow{v}$ belongs to the same facets than c and c^-, and since $c^- + \overrightarrow{v} \in \mathrm{conv}(L_l)$ one has $\mathrm{codir}(c^l) \supseteq \mathrm{codir}(c)$ and $\mathrm{dir}(c) = \mathcal{J}^-$. Thus, c is an infinite birth corner associated to the minimal pseudo birth corner c^l.

2. Assume $S \nsubseteq L[I]$. In that case, there must be a facet of codirection j, for some $j \in \mathcal{J}^-$, that intersects S. Since one has $c^-_j \leq (c^- + \overrightarrow{v})_j < \alpha_j$ for any $j \in \mathcal{J}^-$, this means that the facet would not intersect K, which yields to a contradiction as per Definition 4.1, item (4).

This concludes that $\mathrm{supp}(I) \subseteq \mathrm{supp}(\tilde{I})$, and the equality between these supports holds. □

Proposition 4.4 extends to the following corollary, whose proof is immediate from the definition of exact matchings (see Definition 3.5 above).

Corollary 4.6.1. *Let M be an interval decomposable multipersistence module, whose interval summands all satisfy the assumptions of Proposition 4.4. Let $\tilde{M}$ be the multipersistence module computed by Algorithm 1. Then, one has*

$$d_I(M, \tilde{M}) = d_b(M, \tilde{M}) = 0.$$

5. Multipersistence module approximation

In this section, we propose an approximation result, which states that the bottleneck and interleaving distances between an interval decomposable multipersistence module M and its approximation $\tilde{M}$ computed with Algorithm 1 can be upper bounded under weaker assumptions than the ones in Proposition 4.4 In order to do this, we first characterize a family of approximation modules, that we call *candidates* in Section 5.1, and whose distance to a target module can be controlled. Then, we show in Section 5.2 that the module approximation computed by Algorithm 1 belongs indeed to this family.

5.1. Candidates and approximation error

In this section, we define a family of "good" candidate multipersistence modules (see Definition 5.1) for approximating an interval decomposable multipersistence module M, in the sense that $d_I(M, \tilde{M})$ and $d_b(M, \tilde{M})$ are upper bounded for any module $\tilde{M}$ in this family.

Support assumption. In order to simplify proofs, we assume in this section that $\mathrm{supp}(M) \subseteq K$, where K is a compact set in $\mathbb{R}^n$. This assumption is used in practice, for instance in [CB20, CFK+19, Vip20b], where multipersistence modules are either finite or intersected with a compact set in order to generate descriptors.

Candidates. We first define *candidate modules*, which are, roughly speaking, modules with the same fibered barcodes than M on a regular set of lines, paired with a *candidate pairing* that commutes with the exact matching induced by M.

Definition 5.1 (Candidate)**.** Let K be a compact set of $\mathbb{R}^n$, $\delta > 0$ and L be a δ-grid of $K^{2\delta}$. Let $M = \bigoplus_{i \in \mathcal{I}} I_i$ be an interval decomposable multipersistence module, with L-fibered barcode $\mathcal{FB}(M)_L = \left\{\mathcal{B}(M|_l) \ : \ l \in L\right\}$. Let σ_M be its associated exact matching. An interval decomposable multipersistence module $\tilde{M} = \bigoplus_{i \in \tilde{\mathcal{I}}} \tilde{I}_i$ is called an *L-candidate* of M if:

1. $\mathcal{B}(M|_l) = \mathcal{B}(\tilde{M}|_l)$ for any $l \in L$, i.e., their L-fibered barcodes are the same, and

2. there exists a surjection $v \colon \mathcal{I} \to \tilde{\mathcal{I}}$ such that $i \notin \mathrm{coim}(v) \Rightarrow d_I(I_i, \mathbf{0}) \leq \delta$, and such that, for any two consecutive lines $l, l' \in L$, the following diagram commutes:

$$\begin{array}{ccc} \mathcal{B}(M|_l) & \xrightarrow{\sigma_M} & \mathcal{B}(M|_{l'}) \\ \downarrow{\scriptstyle v_l} & & \downarrow{\scriptstyle v_{l'}} \\ \mathcal{B}(\tilde{M}|_l) & \xrightarrow{\sigma_{\tilde{M}}} & \mathcal{B}(\tilde{M}|_{l'}) \end{array}$$

   where $v_l \colon I_i|_l \in \mathcal{B}(M|_l) \mapsto \tilde{I}_{v(i)}|_l \in \mathcal{B}(\tilde{M}|_l)$. In other words, M and $\tilde{M}$ have the same matched barcodes along L, up to interval reordering. We call σ the *candidate interval pairing*.

We now claim that multipersistence modules that are L-candidates of a given interval decomposable multipersistence module M are δ close to M, as stated in the following approximation result.

Proposition 5.2 (Approximation result)**.** *Let K be a compact set of $\mathbb{R}^n$, $\delta > 0$ and L be a δ-grid of $K^{2\delta}$. Let M be an interval decomposable multipersistence module. Then, any L-candidate $\tilde{M}$ of $M|_K$ satisfies $d_I(\tilde{M}, M|_K) \leq d_b(\tilde{M}, M|_K) \leq \delta$.*

Proof. Let $M = \bigoplus_{i\in\mathcal{I}} I_i$ and $\tilde{M} = \bigoplus_{i\in\tilde{\mathcal{I}}} \tilde{I}_i$ be the interval decompositions of M and $\tilde{M}$, and σ be the associated candidate interval pairing. Without loss of generality, assume that the support of M is included in K, i.e., $M = M\big|_K$. In order to upper bound the bottleneck distance $d_b(M, \tilde{M})$, one can upper bound the interleaving distance $d_I(I_i, \tilde{I}_{\nu(i)})$ for any index $i \in \mathcal{I}$. Let I and $\tilde{I}$ be two such intervals (we drop the index i to alleviate notations). Since I and $\tilde{I}$ are interval modules, and thus indicator modules, the morphisms $\varphi^{(\delta)}_{I\to\tilde{I}} \colon I \to \tilde{I}[\boldsymbol{\delta}]$ and $\varphi^{(\delta)}_{\tilde{I}\to I} \colon \tilde{I} \to I[\boldsymbol{\delta}]$ are well-defined, as per Definition 2.15). We thus need to show that these morphisms commute, i.e., that they induce a δ-interleaving. Hence, we first show that

$$\left(\varphi^{(\delta)}_{\tilde{I}\to I}\right)_{x+\boldsymbol{\delta}} \circ \left(\varphi^{(\delta)}_{I\to\tilde{I}}\right)_x = \left(\varphi^{(2\delta)}_I\right)_x, \tag{6}$$

for any $x \in \mathbb{R}^n$.

Let $x \in K$. If $x \in l$ for some line $l \in L$, Equation (6) is satisfied from $\mathrm{supp}(I) \cap l = \mathrm{supp}(\tilde{I}) \cap l$, which itself comes from the fact that $\tilde{M}$ is an L-candidate of M. Hence, we assume in the following that $x \notin \cup_{l\in L} l$. Furthermore, if $x \notin \mathrm{supp}(I)$ or $x + 2\boldsymbol{\delta} \notin \mathrm{supp}(I)$, then Equation (6) is trivially satisfied. Hence, we also assume $x, x + 2\boldsymbol{\delta} \in \mathrm{supp}(I) \subseteq K$. This means that b^I_x and d^I_x are well-defined, and that $\left(\varphi^{(2\delta)}_I\right)_x \cong \mathrm{id}_{k\to k}$. Thus we only have to show that $\tilde{I}_{x+\boldsymbol{\delta}} \cong k$, i.e., $x + \boldsymbol{\delta} \in \mathrm{supp}(\tilde{I})$.

As L is a δ-grid, let $l \in L$ be a line such that $x \in \mathrm{conv}(L_l)$ and let $l_x \subseteq \mathrm{conv}(L_l)$ be the diagonal line passing through x. Now, as $R_{x,x+\boldsymbol{\delta}} \subseteq \mathrm{supp}(I)$, Lemma 2.22 ensures that $\mathcal{B}(I\big|_l) \neq \emptyset$ for any line $l \in L$ that is δ-comparable to l_x; and the same holds for $\tilde{I}$ since $\mathcal{FB}(I)_L = \mathcal{FB}(\tilde{I})_L$. Using Proposition 3.9 on both I and $\tilde{I}$, there exist two sets B_l and D_l such that $d^I_x, d^{\tilde{I}}_x \in D_l$ and $b^I_x, b^{\tilde{I}}_x \in B_l$, with the segments $l_x \cap B_l$ and $l_x \cap D_l$ having length at most δ. Since one also has

$$b^I_x \leq x \leq x + 2\boldsymbol{\delta} \leq d^I_x,$$

and $\left\|d^I_x - d^{\tilde{I}}_x\right\|_\infty, \left\|b^I_x - b^{\tilde{I}}_x\right\|_\infty \leq \delta$, one finally has $b^{\tilde{I}}_x \leq x + \boldsymbol{\delta} \leq d^{\tilde{I}}_x$, which concludes that $x + \boldsymbol{\delta} \in \mathrm{supp}(\tilde{I})$. □

Remark 5.3. This bound is sharp up to a $\frac{1}{2}$ factor, as illustrated by Figure 9.

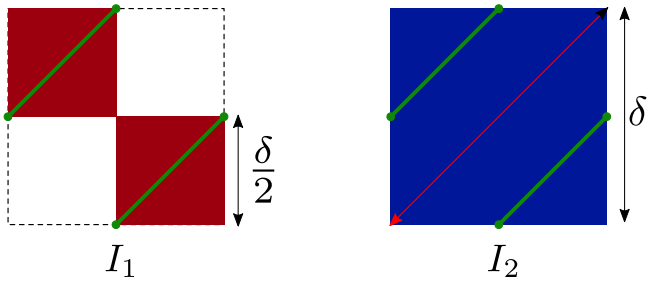

Figure 9: Two interval modules, one with support colored in red **(Left)** and the other in blue **(Right)**. These modules have the same barcodes (green bars) along two δ-consecutive lines; and $d_b(I_1, I_2) = d_I(I_1, I_2) = \delta/2$. This construction can easily be generalized in $\mathbb{R}^n$ with $n > 2$ by setting I_1 as the union of two hypercubes of side length $\delta/2$ located on the anti-diagonal, and I_2 as the standard hypercube with side length δ.

5.2. Algorithm 1 provides a candidate

In this section, we first show, given an interval module I (with support included in a compact K), that the approximation $\tilde{I}$ computed by Algorithm 2 is an L-candidate of I (see Proposition 5.4). This will in turn allow us to state our final approximation bound with Algorithm 1, that is valid for any interval decomposable multipersistence module M (see Proposition 5.5).

Proposition 5.4. *Let I be an interval module with support in a compact set $K \subseteq \mathbb{R}^n$, $\delta > 0$, and L be a δ-grid of $K^{2\delta}$. Let $\tilde{I}$ be the interval module computed with Algorithm 2. Then, $\tilde{I}$ is an L-candidate of I.*

Proof. Let $C^L_B(I)$ and $C^L_D(I)$ be the birth and death corners computed by Algorithm 4, i.e., one has

$$\tilde{I} = \mathbf{Ind}\left(\bigcup_{c\in C^L_B(I)} \bigcup_{c'\in C^L_D(I)} R_{c,c'}\right), \tag{7}$$

In order to show that $\tilde{I}$ is an L-candidate of I, we need to show that the L-fibered barcodes of $\tilde{I}$ and I are the same, i.e., $\mathcal{FB}(I)_L = \mathcal{FB}(\tilde{I})_L$. Equivalently, we need to show that $\mathrm{supp}(I) \cap l = \mathrm{supp}(\tilde{I}) \cap l$ for any line $l \in L$. We first show that

they share the same birthpoints, i.e., that $L[I] \cap L = L[\tilde{I}] \cap L$. Let $l \in L$. Note that b_l^I and $b_l^{\tilde{I}}$ are comparable since they belong to the same diagonal line l.

Strategy. In order to show $b_l^{\tilde{I}} = b_l^I$, we are going to show that 1. $b_l^I \leq b_l^{\tilde{I}}$ and 2. $b_l^{\tilde{I}} \leq b_l^I$.

1. In order to show $b_l^I \leq b_l^{\tilde{I}}$, we are going to show that $c \not< b_l^I$ for any corner $c \in C_B^L(I)$. Indeed, if one assumes $b_l^I > b_l^{\tilde{I}}$, and since there always exists a birth corner $c \in C_B^L(I)$ such that $c \leq b_l^{\tilde{I}}$ by construction of $\tilde{I}$, one has $c \leq b_l^{\tilde{I}} < b_l^I$.
2. In order to show $b_l^{\tilde{I}} \leq b_l^I$, we are going to show that there exists a corner $c \in C_B^L(I)$ such that $c \leq b_l^I$. Indeed, if there is such a birth corner, and if $b_l^{\tilde{I}} > b_l^I$ by contradiction, then $c \leq b_l^I < b_l^{\tilde{I}}$, and $R_{c,b_l^{\tilde{I}}}$ is not flat, contradicting Remark 2.20.

Proof of (2). By construction of $\tilde{I}$ with Algorithm 4, if b_l^I is labelled, then there exists a line l' and a corner $c^{l'} \in C_B^L(I)$ that is smaller than b_l^I so we can take $c := c^{l'}$. If b_l^I is not labelled, it belongs itself to $C_B^L(I)$, and we can take $c := b_l^I$.

Proof of (1). Let $c \in C_B^L(I)$ be a birth corner, and let L_{l_0} be the associated surrounding set of lines for some $l_0 \in L$. Let $[c]_l := \min\left[(c + (\mathbb{R}_+)^n) \cap l\right]$ be the smallest element in the intersection between the positive cone on c and l. Assume $[c]_l \geq b_l^I$ and $c < b_l^I$. Then $R_{c,[c]_l}$ is not flat, contradicting the fact that $[c]_l$ is the smallest element. Thus, we only have to show $[c]_l \geq b_l^I$. There are two cases.

- Either some birthpoints of L_{l_0} are not labelled by Algorithm 3, and c is equal to $b_{l'}^I$ for some $l' \in L_{l_0}$. Now, assume $[c]_l < b_l^I$ by contradiction. Then $b_{l'}^I = c \leq [c]_l < b_l^I$. Thus $b_{l'}^I < b_l^I$ and $R_{b_{l'}^I, b_l^I}$ is not flat, contradicting Remark 2.20. Hence $[c]_l \geq b_l^I$.
- Or all the birthpoints of L_{l_0} are labelled by Algorithm 3. Again, we study two separate cases. See Figure 10 for an illustration.
  - Either $l \in L_{l_0}$. Then, $\exists i \in [\![1, n]\!]$ such that $(b_l^I)_i = c_i$. This yields $(b_l^I)_i = c_i \leq ([c]_l)_i$, and thus $[c]_l \geq b_l^I$ since they both belong to the same diagonal line l.
  - Or the line l does not belong to L_{l_0}. Since $[c]_l$ is on the boundary of the positive cone based on c, there exists $i \in [\![1, n]\!]$ such that $([c]_l)_i = c_i$. Assume again by contradiction that $b_l^I > [c]_l$, and write

$$[c]_l = c + \sum_{j \neq i} (\delta \alpha_j) e_j =: c + \overrightarrow{v} < b_l^I$$

    with $\alpha_j \geq 0$ for $j \in [\![1, n]\!] \setminus \{i\}$. Since $l \notin L_{l_0}$, there exists some j_0 such that $\alpha_{j_0} > 1$. Let $\overrightarrow{u} := ((\overrightarrow{v}_j \bmod \delta)_{j \in [\![1,n]\!]}) = (([c]_l - c)_j \bmod \delta)_{j \in [\![1,n]\!]} \in [0, \delta)^n \leq \overrightarrow{v}$. Let $l' := l_{c+\overrightarrow{u}}$ be the diagonal line passing through $c + \overrightarrow{u}$. Now, recall that the lines of L are drawn on a grid (see Remark 3.3), so $l' \in L$ since $l' = l + \overrightarrow{u} - \overrightarrow{v}$. Moreover, one has by definition, $c \in \mathrm{conv}(L_{l_0})$. Since the lines of L are on a grid, one has

$$\forall l_1, l_2 \in L, \|l_1 \cap H_n, \mathrm{conv}(L_{l_2}) \cap H_n)\|_\infty < \delta \Longrightarrow l_1 \in L_{l_2} \quad \text{where } H_n = \{x \in \mathbb{R}^n : x_n = c_n\}.$$

    Now, note that $c + \overrightarrow{u}$ and $c + \overrightarrow{u} - \overrightarrow{u}_n \cdot \mathbf{1}$ both belong to l', and that $c + \overrightarrow{u} - \overrightarrow{u}_n \cdot \mathbf{1} \in H_n$. Moreover, since

$$\left\|(c + (\overrightarrow{u} - \overrightarrow{u}_n \cdot \mathbf{1})) - c\right\|_\infty = \left\|\overrightarrow{u} - \overrightarrow{u}_n \cdot \mathbf{1}\right\|_\infty < \delta,$$

    one has $l' \in L_{l_0}$. Thus, there exists $i' \in [\![1, n]\!]$ such that $(b_{l'}^I)_{i'} = c_{i'} \leq (c + \overrightarrow{u})_{i'}$ and thus $b_{l'}^I \leq (c + \overrightarrow{u})$ since $b_{l'}^I$ and $c + \overrightarrow{u}$ are comparable on the diagonal line l'. Finally, $b_{l'}^I \leq c + \overrightarrow{u} \leq c + \overrightarrow{v} < b_l^I$, and $R_{b_{l'}^I, b_l^I}$ is not flat, contradicting Remark 2.20. Hence, $b_l^I \leq [c]_l$.

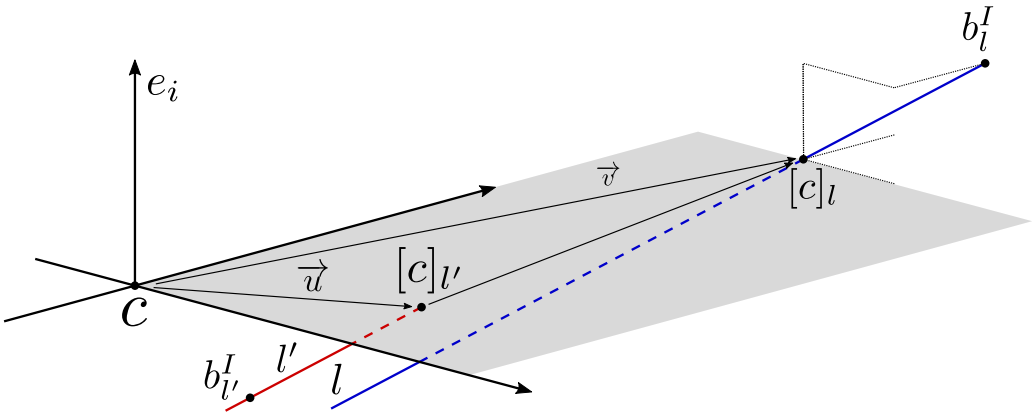

Figure 10: Illustration of $l, l', c, [c]_l, [c]_{l'}, \overrightarrow{u}, \overrightarrow{v}, b_l^I, b_{l'}^I$ when one assumes that $[c]_l < b_l^I$.

The proof applies straightforwardly to deathpoints by symmetry. □

Proposition 5.5 (Conclusion)**.** *Let $M = \bigoplus_{i\in\mathcal{I}} I_i$ be an interval decomposable multipersistence module. Let K be a compact rectangle in $\mathbb{R}^n$, $\delta > 0$, and let L be a δ-grid of $K^{2\delta}$. Let $\tilde{M} = \bigoplus_{i\in\tilde{\mathcal{I}}} \tilde{I}_i$ be the multipersistence module computed with Algorithm 1. Note that $\tilde{\mathcal{I}} \subseteq \mathcal{I}$ by construction. Then,*

1. *if a summand I_i of M is 2δ-discretely presented, then $d_I(I_i, \tilde{I}_i) = 0$.*
2. *if a summand I_i of M has a support included in K, then $d_I(I_i, \tilde{I}_i) < \delta$.*
3. *if a summand I_i of M is δ-trivial (i.e., $d_I(I_i, 0) \leq \delta$), then either $\mathrm{supp}(I_i) \cap L \neq \varnothing$ and thus $i \in \tilde{\mathcal{I}}$ and $\tilde{I}_i$ is also δ-trivial, or $\mathrm{supp}(I_i) \cap L = \varnothing$ and $\tilde{M}$ has no summand matched to I_i.*

In particular, if $K \subseteq \mathbb{R}^n$ contains the supports of those summands of M whose support is precompact, and if the remaining summands are all δ-discretely presented in K, then one has $d_I(M, \tilde{M}) \leq d_b(M, \tilde{M}) \leq \delta$.

Proof. Item (1) comes from Proposition 4.4. Item (2) is a direct consequence of Propositions 5.2 and 5.4. Item (3) comes from the construction of $\tilde{I}_i$. □

6. Exact matching

The algorithms and theoretical results presented in the previous sections were all obtained using exact matchings (see Definition 3.5). In this section, we seek to understand conditions under which a given matching function is exact. We first present assumptions that allow for finding exact matching functions in Section 6.1). Then, we discuss these assumptions in Section 6.2, and we finally show that the vineyard matching [CSEM06] is exact in Section 6.3.

6.1. A naive approach to exact matching

In order to understand which matching functions are exact, we first define a notion of *compatibility* between bars.

Definition 6.1 (Compatible bars)**.** Let I be an interval module, and let $l_1, l_2 \subseteq \mathbb{R}^n$ be two δ-consecutive diagonal lines. Assume $\mathrm{supp}(I) \neq \varnothing$ and $\mathrm{supp}(\tilde{I}) \neq \varnothing$, and let $[b_{l_1}^I, d_{l_1}^I]$ and $[b_{l_2}^I, d_{l_2}^I]$ be the corresponding bars. These bars are *compatible* if the rectangles $R_{b_{l_1}^I, b_{l_2}^I}$, $R_{b_{l_2}^I, b_{l_1}^I}$, $R_{d_{l_1}^I, d_{l_2}^I}$ and $R_{d_{l_2}^I, d_{l_1}^I}$ are flat. Moreover, we say that $[b_{l_1}^I, d_{l_1}^I]$ is *compatible with the empty set* in l_2 if $\left\| b_{l_1}^I - d_{l_1}^I \right\|_\infty \leq 2\delta$.

Remark 6.2. It follows from Lemma 2.9 that bars along consecutive lines that correspond to the same indicator summand of a multipersistence module are always compatible.

Compatible bars enjoy some useful properties, that we state in the following proposition.

Lemma 6.3. *Let l_1 and l_2 be two δ-consecutive lines, and let $[b_1, d_1] := \mathcal{B}(I|_{l_1})$ be the bar of an indicator module along l_1. Let $[b_2, d_2]$ be a bar along l_2 that is compatible with $[b_1, d_1]$. Then, d_2 (resp. b_2) is included in a segment of size δ in l_2 that is independent of d_2 (resp. b_2).*

Proof. Applying Lemma 2.22, one has

$$d_2 \in C := \left[B_\delta(d_1) \cap l_2\right] \backslash \left[\{z \in \mathbb{R}^n : z > d_1\} \cup \{z \in \mathbb{R}^n : z < d_1\} \right]$$

Since C is a nonempty, totally ordered set, we can define $y := \min C$. By construction, we know that there exists a dimension i such that $y_i \geq (d_1)_i$, and thus C must be included in the segment $[y, y + \delta \cdot \mathbf{1}]$ along l_2.

The proof applies straightforwardly to b_2 by symmetry. □

Since bars that are matched under an exact matching function are always compatible, one way to construct an exact matching between two barcodes is therefore to isolate, among all possible matching functions, the ones such that matched bars are compatible. If this family contains a single element, it must be the exact matching we are looking for. This typically happens for interval decomposable multipersistence module whose summands are sufficiently separared, as we show in the proposition below.

Proposition 6.4. *Let $M = \bigoplus_{I \in \mathcal{I}} I$ be an interval decomposable multipersistence module. Let $\delta > 0$, and I, I' be two interval summands in the decomposition of M. Assume that the two following properties are satisfied:*

1. *Let $l \subset \mathbb{R}^n$ be a diagonal line such that $\mathrm{supp}(I) \cap l \neq \emptyset$ and $\mathrm{supp}(I') \cap l \neq \emptyset$. Then, one has either $\left\|b_l^I - b_l^{I'}\right\|_\infty > \delta$ or $\left\|d_l^I - d_l^{I'}\right\|_\infty > \delta$. In other words, the endpoints of the bar in $\mathcal{B}(I|_l)$ and of the bar in $\mathcal{B}(I'|_l)$ are at distance at least δ.*
2. *The bars of length at most 2δ in I and I' are at distance at least δ, i.e., if we let $S^I := \left\{l : l \cap \mathrm{supp}(I) \neq \emptyset, \left\|b_l^I - d_l^I\right\|_\infty \leq 2\delta\right\}$ (and similarly for I'), one has*
$$d_\infty(S^I, S^{I'}) > \delta/2.$$
In other words, a small bar in I cannot be too close to a small bar in I'.

Let $K \subseteq \mathbb{R}^n$ be a compact set and L be a δ-grid of K. Then, the matching function m_{comp}, induced by matching bars that are compatible together, is well-defined and exact.

See Figure 11 for an illustration of assumptions (1) and (2).

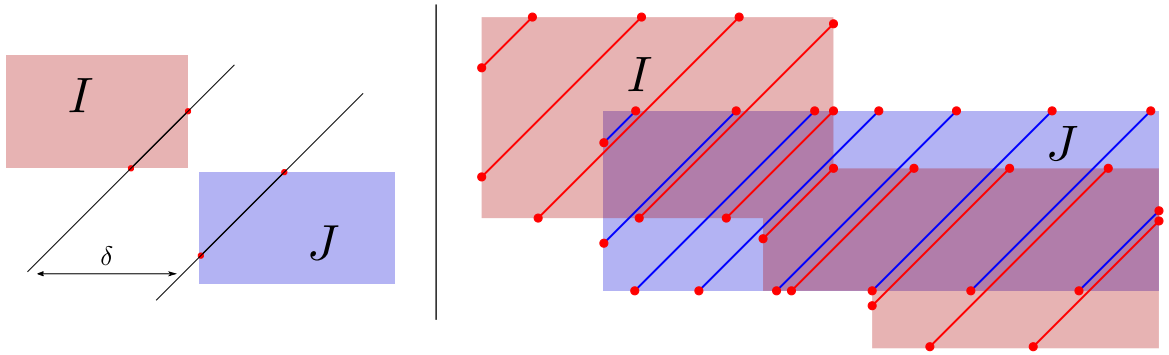

Figure 11: **(Left)** Example of module whose interval summands do not satisfy assumption (2). **(Right)** Example of module whose interval summands do satisfy assumptions (1) and (2). Bars corresponding to consecutive lines can only be matched if they are compatible, which, in this figure, means that they have the same color, i.e., that they are associated to the same interval summand.

Proof. Let I and I' be two interval summands in the decomposition of M. Let l_1 and l_2 be two δ-consecutive lines of L, and let $b := \mathcal{B}(I|_{l_1})$ be the bar corresponding to I along l_1. We will show that m_{comp} must match b to either $b' := \mathcal{B}(I|_{l_2})$ if $\mathrm{supp}(I) \cap l_2 \neq \emptyset$, or the empty set if $\mathrm{supp}(I) \cap l_2 = \emptyset$.

- If $\mathrm{supp}(I) \cap l_2 = \emptyset$, then by Lemma 2.22, the length of b is at most δ, i.e., $\left\|b_{l_1}^I - d_{l_1}^I\right\|_\infty \leq \delta$. It is thus compatible with the empty set. Now, since $d_\infty(l_1, l_2) = \delta/2$ and since $l_1 \in S^I$, assumption (2) ensures that the bar $b'' := \mathcal{B}(I'|_{l_2})$ (if it exists) must be of length at least 2δ. In particular, it is not compatible with b, hence m_{comp} cannot match b to b'', and must match b to the empty set.
- If $\mathrm{supp}(I) \cap l_2 \neq \emptyset$, then the bar $b' = [b_{l_2}^I, d_{l_2}^I]$ in $\mathcal{B}(I|_{l_2})$ is compatible with b, as per Remark 6.2. According to Lemma 6.3, it follows that the birthpoint and deathpoint of any bar along l_2 that is compatible to l_1 must belong to segments s_b, s_d of length δ that contain $b_{l_2}^I$ and $d_{l_2}^I$ respectively. Let $b'' := [b_{l_2}^{I'}, d_{l_2}^{I'}]$ be the bar in $\mathcal{B}(I'|_{l_2})$ (if it exists). According to assumption (1), we either have $\left\|b_{l_2}^I - b_{l_2}^{I'}\right\|_\infty > \delta$ or $\left\|d_{l_2}^{I'} - d_{l_2}^{I'}\right\|_\infty > \delta$. In particular this means that either $b_{l_2}^{I'} \notin s_b$ or $d_{l_2}^{I'} \notin s_d$. Hence b'' is not compatible with b, and m_{comp} must match b to b'.

In both cases, m_{comp} is well-defined and exact. □

Note that, since the number of lines of L, and thus their spacing δ, is controlled by the user in Algorithm 1, one can ensure that the assumptions of Proposition 6.4 are satisfied by asking L to have a sufficient number of lines in Algorithm 1 (which obviously increases the complexity as well, unfortunately). Note also that bars matched under the vineyard matching [CSEM06] are always compatible (see Section 6.3), which ensures that the vineyard matching is exact when L contains a sufficient number of lines.

6.2. Limitations

One might wonder whether the usual distances between barcodes, such as the bottleneck or Wasserstein distances, could be used to define exact matching functions, instead of having to look for matching functions that only match bars that are compatible. Indeed, a major advantage of, e.g., Wasserstein distances, is that their associated matching functions is usually unique. However, when the spacing δ between two lines is too large, Wasserstein distances can still fail to match bars exactly, if assumptions (1) or (2) are not satisfied, as shown in Figure 12.

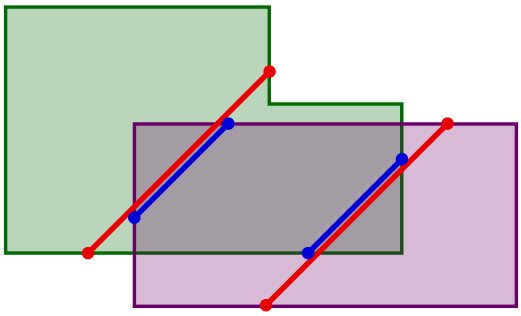

Figure 12: Example interval decomposable multipersistence module with two interval summands (green and purple), and its barcodes along two lines (here the two couples of red-blue bars). Any matching function induced by, e.g., Wasserstein distances between the barcodes, will match the red bar with the red bar and the blue bar with the blue bar; however, this matching is not exact.

Moreover, even when the spacing δ is small, it is easy to build examples where assumptions (1) and (2) are not satisfied. The toy example in Figure 13 shows that finding exact matching functions from bar compatibility alone can lead to poor results in general.

Figure 13: The interval decomposable multipersistence modules on the **(Left)** and on the **(Right)** both have a yellow and a brown summand, and have the same fibered barcodes. Since bars corresponding to lines in the middle have multiplicity 2, matching functions identified using bar compatibility can match them arbitrarily.

Furthermore, even a single mistake in the matching between consecutive barcodes can lead to arbitrary different decompositions, as illustrated in Figure 14.

Figure 14: The modules on the **(Left)** and on the **(Right)** are both decomposable into two interval summands (yellow and brown). These modules, which are at a large bottleneck distance from each other, can be obtained from a single matching exchange in the middle of the small square.

One way to handle these issues is to use the representative chains associated to the bars of the fibered barcode in order to find a matching. Indeed, given two bars in consecutive barcodes, one can compare representatives of their generator chains in order to check if they correspond to the same underlying interval summand, by assessing whether one chain can be obtained from the other through the addition of another positive chain that appeared earlier than the current chain (in its corresponding filtration). Note that computing barcodes and matching their bars through representative chains in two separate steps is not efficient: for simplicial complexes, the cost of computing the barcode on a given line is $O(N^3)$ (where N is the number of simplices), and checking if two generator chains are associated to the same summand takes $O(N^2)$ operations (which is the cost of applying Gaussian elimination on a column of the boundary matrix). Fortunately, the so-called *vineyard* algorithm [CSEM06] can perform both operations at the same time, i.e., it can

reduce the boundary matrices of different lines, and retrieve matching functions between the barcodes of consecutive lines as a byproduct. We detail these statements in the next section.

6.3. Vineyard matching

In this section, we prove that the matching induced by the vineyard algorithm [CSEM06] is an exact matching for multipersistence modules computed from simplicial complexes (although this seems to be common knowledge, we could not find a proof of this result in the literature). We first recall the basic notions of persistent homology from simplicial complexes in Section 6.3.1, and then provide an analysis of the vineyard algorithm in Section 6.3.2.

6.3.1. Persistent homology of simplicial complexes. We assume in the following that the reader is familiar with simplicial complexes, boundary operators and homology groups, and we refer the interested reader to [Mun84, Chapter 1] for a thorough treatment of these notions. The first important definition is *filtered simplicial chain complexes*.

Definition 6.5. Let S be a simplicial complex, and $f\colon S \to \mathbb{R}$ be a *filtration function*, i.e., f satisfies $f(\sigma) \leq f(\tau)$ when $\sigma \subseteq \tau$. Then, the *filtered simplicial chain complex* (S, f) is defined as $(S, f) = ((C_t)_{t\in\mathbb{R}}, \iota)$, where

- $C_t = \langle \sigma_0, \dots, \sigma_i \rangle$ is the vector space over a field k whose basis elements are the simplices that have filtration values smaller than t, i.e., $\{\sigma_0, \dots, \sigma_i\} = \{\sigma \in S : f(\sigma) \leq t\}$, and
- for any $s \leq t$, the map $\iota = \iota_s^t\colon C_s \hookrightarrow C_t$ is the canonical injection.

Note that f can be used to define an order on the simplices of $S = \{\sigma_i\}_{i=0}^N$, by using the ordering induced by the filtration values. In other words, we assume in the following that $f(\sigma_0) \leq f(\sigma_1) \leq \cdots \leq f(\sigma_N)$. We also slightly abuse notations and define $C_i := \langle \sigma_0, \dots, \sigma_i \rangle$ for any $i \in [\![0, N]\!]$ and

$$(S, f) = \left(C_0 \xhookrightarrow{\iota_0} C_1 \xhookrightarrow{\iota_1} \dots \xhookrightarrow{\iota_{N-1}} C_N = \langle S \rangle \right). \tag{8}$$

Then, applying the homology functor H_* on this filtered simplicial chain complex yields the following one-dimensional persistence module

$$H_*(S, f) = 0 \to H_*(C_0) \to H_*(C_1) \to \cdots \to H_*(C_N).$$

An important theorem of (one-dimensional) persistent homology states that, up to a change of basis, it is possible to pair some chains together in order to define the so-called *one-dimensional persistence barcode* associated to the filtered simplicial chain complex.

Theorem 6.6 (Persistence pairing, [dMV11, Theorem 2.6])**.** *Given a filtered simplicial chain complex $(S, f) = C_1 \hookrightarrow C_2 \hookrightarrow \cdots \hookrightarrow C_N$ and associated persistence module $H_*(S, f)$, there exists a partition $[\![1, N]\!] = E \sqcup B \sqcup D$, a bijective map* $\mathrm{Low} : D \to B$*, and a new basis $\hat{\sigma}_1, \dots, \hat{\sigma}_N$ of C, called* reduced basis*, such that:*

- $C_i = \langle \hat{\sigma}_1, \dots, \hat{\sigma}_i \rangle$,
- $\partial \hat{\sigma}_e = 0$ *for any* $e \in E$,
- *for any $d \in D$, one has $\partial \hat{\sigma}_{\mathrm{Low}(d)} = 0$, and $\partial \hat{\sigma}_d$ is equal to $\hat{\sigma}_{\mathrm{Low}(d)}$* up to simplification*, i.e., there exists a set of indices* $\mathrm{bd}(d)$ *such that (i) $j < \mathrm{Low}(d) \leq d$ for any $j \in \mathrm{bd}(d)$, and (ii) $\partial \hat{\sigma}_d = \hat{\sigma}_{\mathrm{Low}(d)} + \sum_{j\in\mathrm{bd}(d)} \hat{\sigma}_j$.*

In particular, the chains $\{\hat{\sigma}_j : j \in E \cap [\![1, i]\!]\} \cup \{\hat{\sigma}_j : j \in B \cap [\![1, i]\!]$ and $\exists d > i$ s.t. $\mathrm{Low}(d) = j\}$ form a basis of the simplicial homology groups $H_*(C_i)$. Moreover, the chains $\{\hat{\sigma}_j : j \in B \sqcup E\}$ are called generator chains *while the chains $\{\hat{\sigma}_j : j \in D\}$ are called* relation chains*.*

The multiset of bars $\mathcal{B} := \{[f(\sigma_b), f(\sigma_d)] : b = \mathrm{Low}(d)\} \cup \{[f(\sigma_e), +\infty) : e \in E\}$ is called the persistence barcode *of the filtered simplicial chain complex (S, f) and of the single-parameter persistence module $H_*(S, f)$.*

Note that while the reduced basis $\{\hat{\sigma}_1, \dots, \hat{\sigma}_N\}$ does not need to be unique, the pairing map Low is actually independent of that reduced basis (see [EH10, VII.1, Pairing Lemma]).

6.3.2. Vineyard algorithm and matching. The vineyard algorithm is a method that allows to find reduced chain bases for filtered simplicial complexes whose simplex orderings only differ by a single transposition of consecutive simplices, that we denote by $(i\ i+1)$.

Proposition 6.7 ([CSEM06])**.** *Let $S = \{\sigma_1, \dots, \sigma_N\}$ be a (filtered) simplicial complex (with filtration function f), and let $B = \{\hat{\sigma}_1, \dots, \hat{\sigma}_N\}$ be a corresponding reduced chain basis. Let $\tilde{f} : S \to \mathbb{R}$ be a filtration function that swaps the simplices at positions i and $i+1$, i.e., that induces a new filtered simplicial complex $\tilde{S} = \{\sigma_1, \dots, \sigma_{i+1}, \sigma_i, \dots, \sigma_N\}$. Note that the swapped basis $\mathrm{sw}_i^S(B) := \{\hat{\sigma}_1, \dots, \hat{\sigma}_{i+1}, \hat{\sigma}_i, \dots, \hat{\sigma}_N\}$ might not be a reduced basis for $(\tilde{S}, \tilde{f})$, since the pairing map* Low *might not be well-defined anymore. Fortunately, there exists a change of basis, called* vineyard update*, that turns $\mathrm{sw}_i^S(B)$ into a reduced basis $\tilde{B}$ of $(\tilde{S}, \tilde{f})$ in $O(N)$ time, and that comes with a bijective map* vine $: B \to \tilde{B}$*, called* vineyard matching.

Note that the vineyard matching can be straightforwardly extended to filtration functions whose induced ordering differs by a whole permutation from the one of the initial filtration function by simply decomposing the permutation into elementary transpositions (using, e.g., Coxeter decompositions). However, one might wonder whether the resulting vineyard matching depends on the decomposition of the permutation or not. While the vineyard matching seems to be independent from the decomposition that we used in our experiments, we leave this question as a conjecture for future work.

Conjecture 6.8. *Let $\tau \in \mathfrak{S}_N$ be a simplex permutation of a (filtered) simplicial complex $S = \{\sigma_1, \dots, \sigma_N\}$. Then, the vineyard matching between the reduced bases of S and $\tilde{S} = \{\sigma_{\tau(1)}, \dots, \sigma_{\tau(N)}\}$ does not depend on any sequence $J = \{i_1, \dots, i_k\}$ such that $\tau = \prod_{j=1}^{k} (i_j\ i_j + 1)$.*

6.3.3. Application to multipersistence. When the simplices of S are (partially) ordered from a function $f : S \to \mathbb{R}^n$, i.e., one has $f(\tau) \leq f(\sigma)$ for any $\tau \subseteq \sigma$ (where $\leq$ is the partial order of $\mathbb{R}^n$), then the function f is called a *multi-parameter filtration function* of S. In that case, applying the homology functor also leads to simplicial homology groups $\{H_d(C_i)\}_{i \in I}$, where $I \subseteq [\![1, N]\!]^n$ is the (partial) ordering associated to f, and these groups are connected by morphisms as long as their indices are comparable in $\mathbb{R}^n$. These groups and maps are called the *multi-parameter persistent homology* associated to the multi-filtered simplicial chain complex (S, f). Similarly to the single-parameter case, when k is a field, one can define the *multi-parameter persistence module* associated to (S, f) as the family of vector spaces indexed over $\mathbb{R}^n$ defined with the identifications $M_s := H_d(C)$ where $C = \mathrm{Vect}\{\sigma \in S : f(\sigma) \leq s \in \mathbb{R}^n\}$.

We will now show that using Conjecture 6.8, one can prove that the vineyard algorithm yields an exact matching in the sense of Definition 3.5.

Proposition 6.9. *Let M be an interval decomposable multipersistence module computed from a finite multi-filtered simplicial chain complex (S, f) over $\mathbb{R}^n$, with support included in a compact set K of $\mathbb{R}^n$. Let $\delta > 0$ and L be a δ-grid of $K^{2\delta}$. Assume that, for any two lines $l, l' \in L$, there exists a sequence $l = l_1, \dots, l_k = l'$ such that the simplex orderings induced by l_i and l_{i+1} differ by at most one transposition of two consecutive simplices, for any $i \in [\![1, k-1]\!]$. Then, the vineyard matching is exact.*

Proof. Let l, l' be two diagonal lines in $\mathbb{R}^n$, and let $F := f\big|_l : S \to \mathbb{R}$ and $F' := f\big|_{l'} \to \mathbb{R}$. Up to a reordering of the simplices of $S = \{\sigma_1, \dots, \sigma_N\}$, we assume without loss of generality that $F(\sigma_1) \leq \cdots \leq F(\sigma_N)$. Let $\mathfrak{S}_N^S \subseteq \mathfrak{S}_N$ be the subset of permutations that satisfies $\tau \in \mathfrak{S}_N^S \implies \left(C_k^\tau\right)_{k \in [\![1,N]\!]} = \left(\{\sigma_{\tau(1)}, \dots, \sigma_{\tau(k)}\}\right)_{k \in [\![1,N]\!]}$ is a filtration of S. Finally, let $N = (N_{\tau,k})_{(\tau,k) \in \mathfrak{S}_N^S \times [\![1,N]\!]}$, where $N_{\tau,k} = H_*(C_k^\tau)$ be the multipersistence module indexed over $\mathfrak{S}_N^S \times [\![1, N]\!]$, with arrows induced by inclusion. Note that the one-dimensional persistence module $N_{\mathrm{id}} := (N_{\mathrm{id},k})_{k \in [\![1,N]\!]}$ is isomorphic to $M\big|_l$.

Now, let τ be the simplex permutation that matches the simplex ordering induced by F to the one induced by F'. By definition, one has $\tau \in \mathfrak{S}_N^S$. Let i be the index of the first transposition in the Coxeter decomposition of τ. Then, the matching vine associated to the transposition $(i\ i+1)$ induces a matching between the bars of $\mathcal{B}(M\big|_l)$ and $\mathcal{B}(M\big|_{l'})$, or equivalently, between the bars of $\mathcal{B}(N_{\mathrm{id}})$ and $\mathcal{B}(N_{(i\ i+1)})$, as well as a morphism between N_{id} and $N_{(i\ i+1)}$. Let us now consider the diamond:

$$\begin{array}{ccccc} & & N_{(i\,i+1),i} & & \\ & \nearrow & & \searrow & \\ \cdots \longrightarrow N_{\mathrm{id},i-1} = N_{(i\,i+1),i-1} & & & & N_{\mathrm{id},i+1} = N_{(i\,i+1),i+1} \longrightarrow \cdots \\ & \searrow & & \nearrow & \\ & & N_{\mathrm{id},i} & & \end{array} \tag{9}$$

First, note that all maps in that diamond are either injective of corank 1 or surjective of nullity 1 (since they are all induced by adding a positive or negative chain of the corresponding reduced bases). Moreover, by the Mayer-Vietoris theorem, the following sequence is exact:

$$N_{\mathrm{id},i-1} = N_{(i\,i+1),i-1} = H_*(C_{i-1}^{\mathrm{id}}) \to N_{(i\,i+1),i} \oplus N_{\mathrm{id},i} = H_*(C_i^{(i\,i+1)}) \oplus H_*(C_i^{\mathrm{id}}) \xrightarrow{(x,y) \mapsto y-x} H_*(C_{i+1}^{\mathrm{id}}) = N_{\mathrm{id},i+1} = N_{(i\,i+1),i+1}.$$

Such diamonds are called *transposition diamonds*, and it has been shown in [MO15, Theorem 2.4] that the morphism induced by vine between the lower and upper parts of the diamond matches bars with same representative positive chains together. Hence, bars in $\mathcal{B}(N_{\mathrm{id}})$ and $\mathcal{B}(N_{(i\,i+1)})$ that are matched under vine are associated to the same summand of N. Moreover, by repeating this argumentation with the other transpositions in the decomposition of τ, one has that the same is true for bars in $\mathcal{B}(M\big|_l)$ and $\mathcal{B}(M\big|_{l'})$.

Now, if N is interval decomposable, then, since the transition maps of M can be seen as transition maps of N, it follows that interval summands of M correspond to interval summands of N. In that case, using a dimensionality argument with Theorem 6.6, one can show that two bars of $\mathcal{B}(M\big|_l)$ and $\mathcal{B}(M\big|_{l'})$ that are matched under vine through arrows of N also belong to the same interval summand of M. Unfortunately, even though N and M are constructed from the same

chain complex, N contains much more arrows that M, and we cannot guarantee in the general case that N is interval decomposable.

We will thus conclude with Conjecture 6.8. By assumption, there exists a sequence of diagonal lines between l and l' such that the simplex orderings of two consecutive lines differ by at most a single transposition. The vineyard matching vine associated to these transpositions induce morphisms of M that can be seen as morphisms of N, thus ensuring that it is exact on both M and N between l and l'. Hence, by Conjecture 6.8, the vineyard matching is unique and exact.

□

Remark 6.10. Note that the assumption of Proposition 6.9 is always satisfied when δ becomes smaller than the smallest distance (in filtration function values) between critical points of the module.

7. Experiments

In this section, we showcase the performances of Algorithm 1 on various data sets. More precisely, we evaluate the running times and approximation errors of our approximation scheme on both synthetic and real data sets, and we measure the empirical dependencies on the number of simplices, on the number of lines that are used, and on the dimension n. We also compare our approach to Rivet when $n = 2$. All experiments were done on a laptop with AMD Ryzen 4800 CPU. Our code is publicly available at `https://gitlab.inria.fr/dloiseau/multipers`, and is implemented in C++, with Python interface.

7.1. Simple examples

In this section, we first provide examples of multipersistence module approximation when the underlying multipersistence module is manually crafted and known. In Figures 15 and 16, we provide two examples of pairs of distinct interval decomposable multipersistence modules that have the same pointwise Betti numbers and rank invariants. In both examples, our approximation scheme manages to recover the correct decompositions. In Figure 17, we provide a multipersistence module that is not interval decomposable, and our (fake) candidate decomposition.

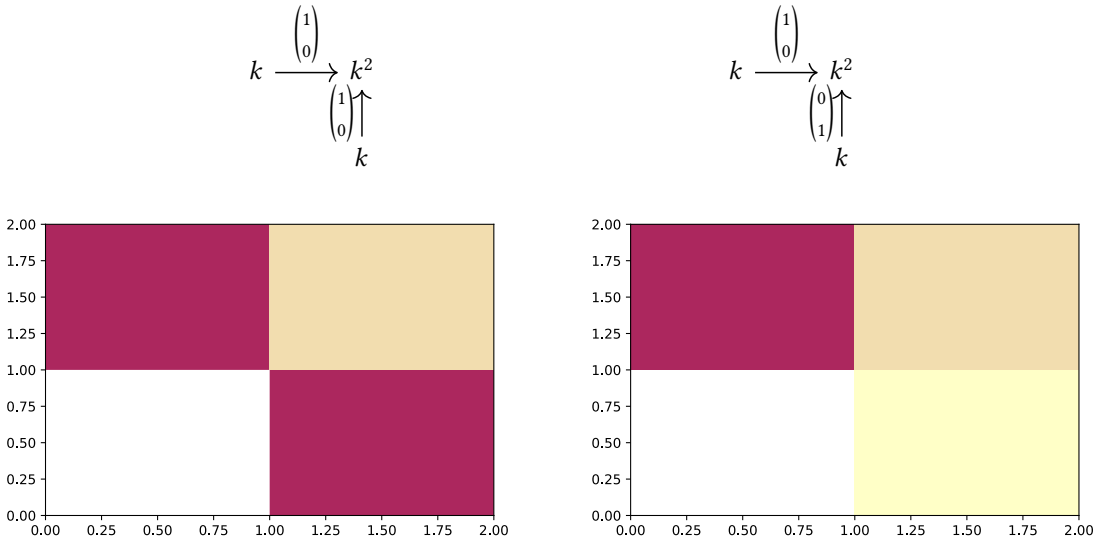

Figure 15: **(Top)** Two distincts interval decomposable modules having the same rank invariant. **(Bottom)** Output of Algorithm 1; each color corresponds to a different summand.

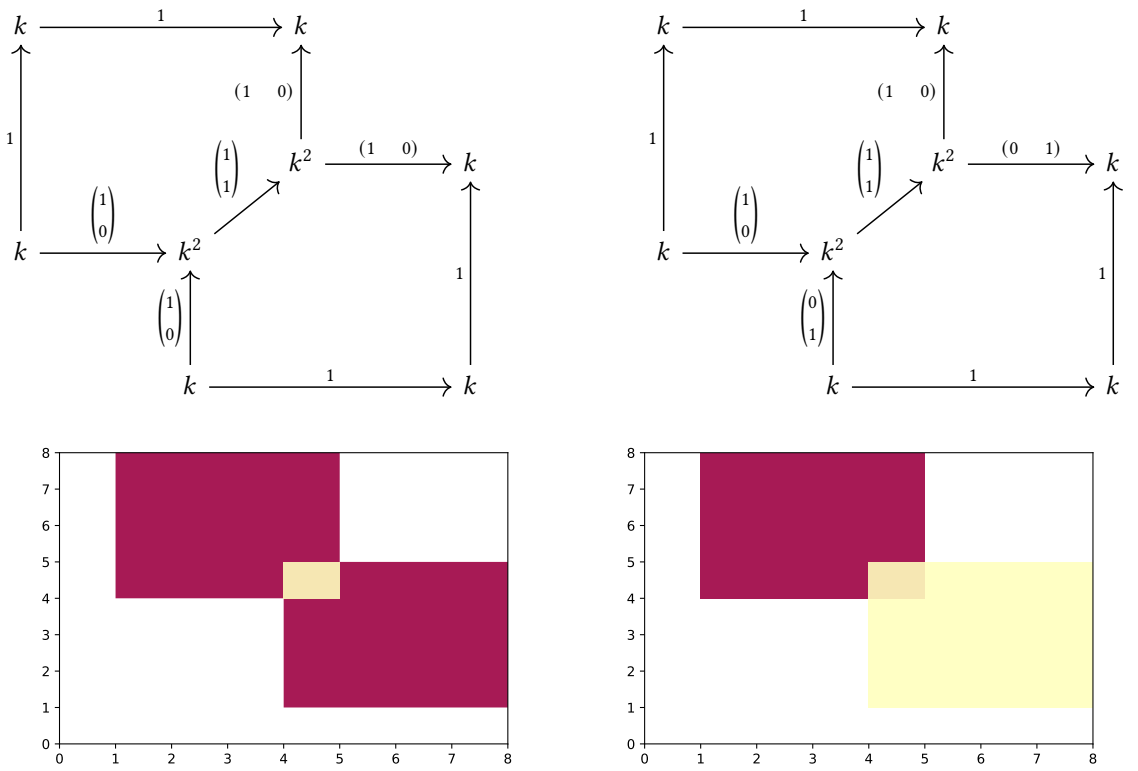

Figure 16: **(Top)** Two distincts interval decomposable modules having the same rank invariant. **(Bottom)** Output of Algorithm 1; each color corresponds to a different summand.

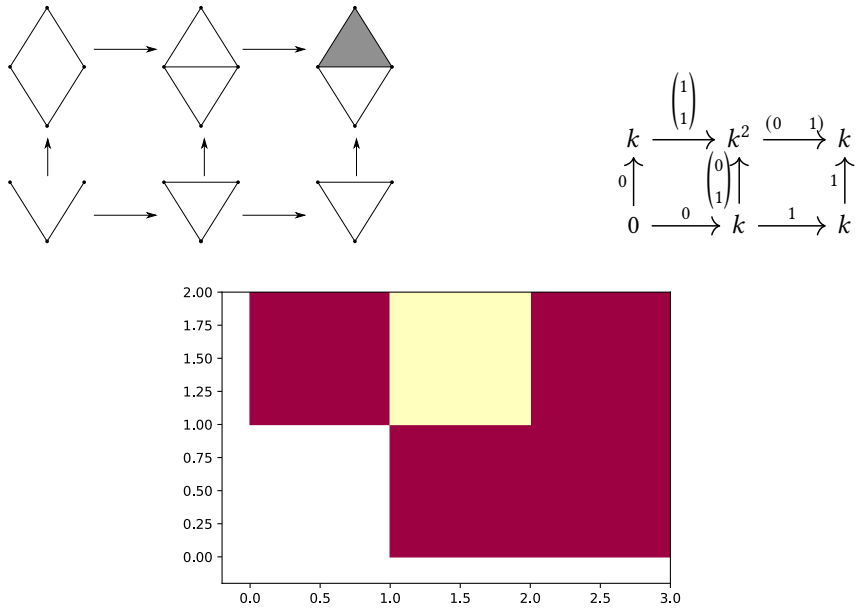

Figure 17: **(Top left)** Filtered simplicial chain complex that leads to the **(Top right)** indecomposable multipersistence module. **(Bottom)** Output of Algorithm 1; each color corresponds to a different summand. The decomposition is not real, although our output still preserves the rank invariant.

7.2. Convergence

In this section, we check the empirical convergence of Algorithm 1 on real data sets. We look at a noisy circle with 1,000 points (60% of those are on the annulus, and the remaining 40% are outliers in the square) in Figure 18, as well as three time series from the UCR archive [Che15] that were embedded in $\mathbb{R}^3$ using time delay embedding in Figures 19, 20 and 21. Our approximation was computed with $n = 2$ filtrations, the Alpha complex filtration and a log-density estimation. Since we do not know the underlying multipersistence module, we use the Euclidean norm between (a) the multipersistence

image [CB20] of our approximation and (b) a limit multipersistence image computed on our approximation with a limit precision (i.e., distance between consecutive lines) of $\delta = 10^{-4}$ as a proxy to measure the distance between our approximation and the true underlying module. In all cases, one can see that the error curve decreases fast and smoothly, as predicted by our approximation result Proposition 5.5.

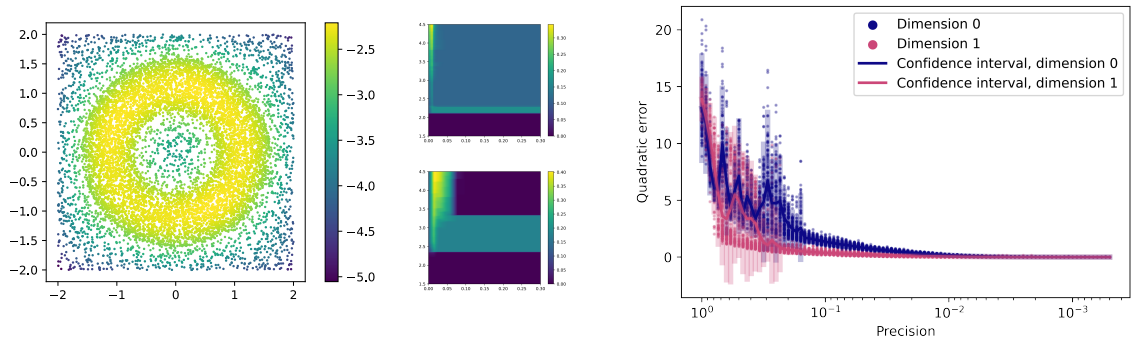

Figure 18: **(Left)** Noisy annulus data set colored by log density. **(Middle)** Multipersistence image in dimensions 0 and 1. **(Right)** Error plot showing convergence.

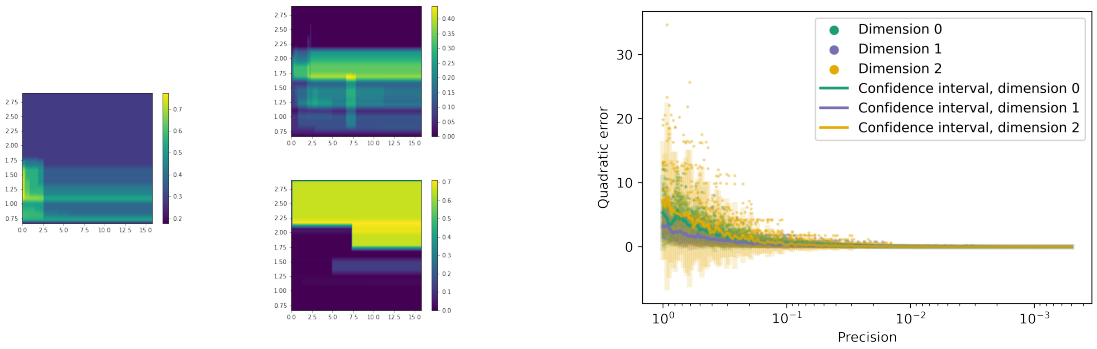

Figure 19: **(Left)** Multipersistence image in dimensions 0 (left), 1 (up right) and 2 (bottom right). **(Right)** Error plot showing convergence for the first time series of the Coffee data set.

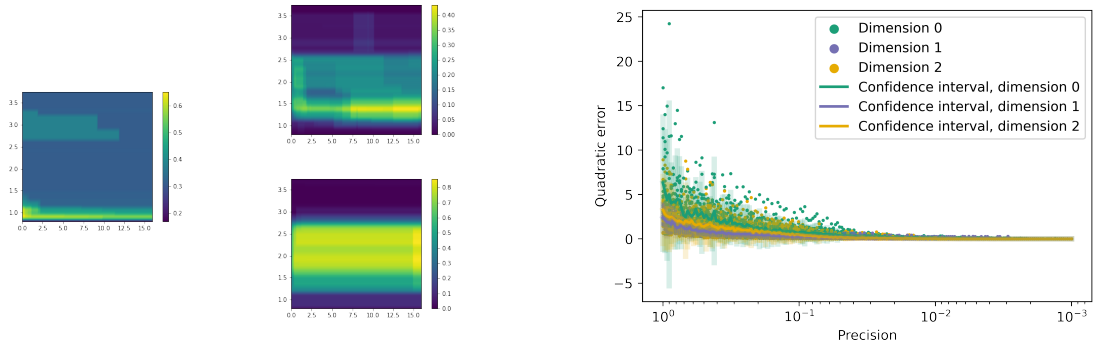

Figure 20: **(Left)** Multipersistence image in dimensions 0 (left), 1 (up right) and 2 (bottom right). **(Right)** Error plot showing convergence for the first time series of the Worms data set.

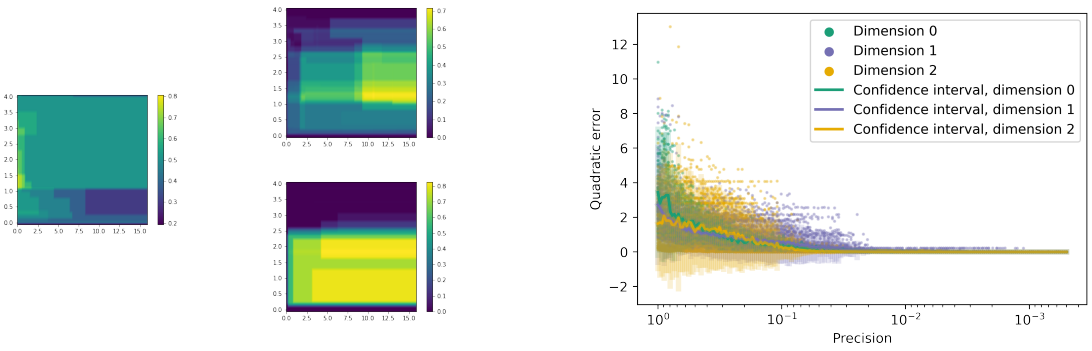

Figure 21: **(Left)** Multipersistence image in dimensions 0 (left), 1 (up right) and 2 (bottom right). **(Right)** Error plot showing convergence for the first time series of the Wine data set.

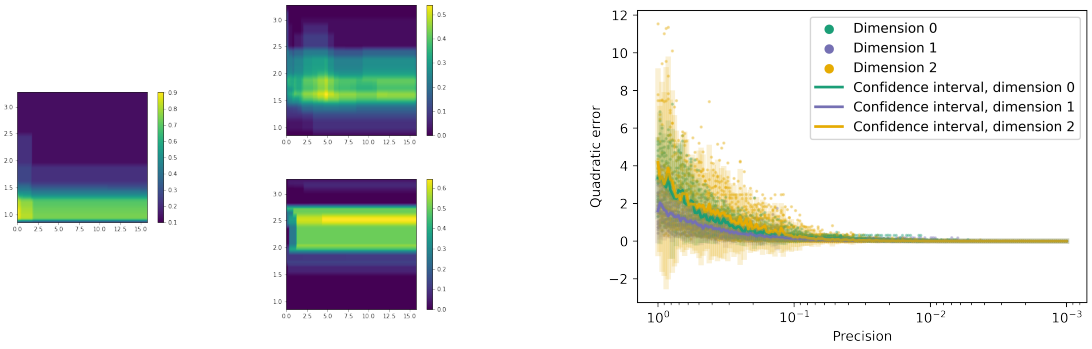

Figure 22: **(Left)** Multipersistence image in dimensions 0 (left), 1 (up right) and 2 (bottom right). **(Right)** Error plot showing convergence for the first time series of the Ham data set.

7.3. Performance

In this section, we empirically check the dependencies between running time and numbers of lines, simplices and dimensions.

7.3.1. Synthetic data with $n = 2$. We first focus on two synthetic data sets: (a) the noisy annulus of Section 7.2, and (b) a random point cloud in the unit square $[0, 1]^2$ with one filtration being the usual Alpha complex filtration, and the other being the lower star filtration of a random function on the points. We show the influence of the number of lines and simplices on the running times for both data sets in Figure 23. As expected, the running time is linear w.r.t. the number of lines. Furthermore, one can see that the linear coefficient depends on the complexity of the data set, as we can see that the random filtration on the points of the square yields longer running times than the noisy annulus. As for the number of simplices, we empirically noted a quadratic dependency with the running times.

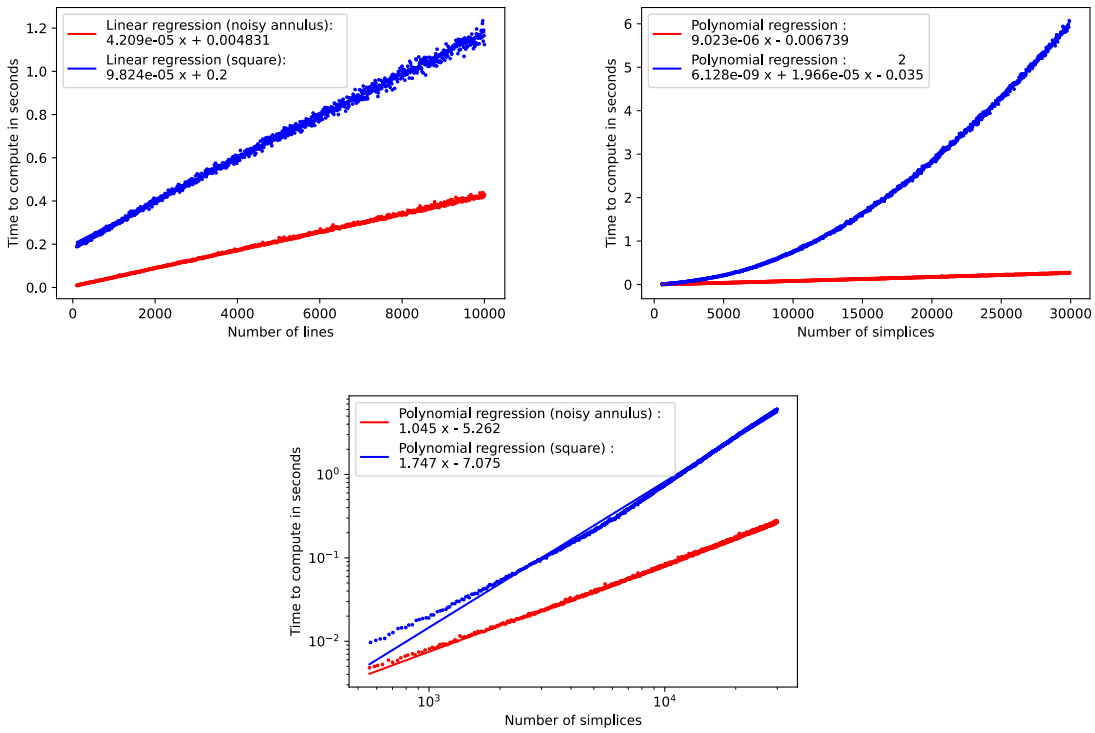

Figure 23: Running times for the noisy annulus (red) and the random point cloud in the unit square (blue). **(Top)** Running times with respect to the number of lines of the two datasets; the number of points is here fixed at 1,000 points. **(Bottom left)** Running times with respect to the number of simplices; the number of lines is here fixed at 1,000 lines. We also show the curve in log-log scale in **(Bottom right)**.

7.3.2. Higher dimension. To illustrate the fact that Algorithm 1 can run with more than two filtrations, we now focus on a synthetic data set. We uniformly sample 300 points in the unit square $[0,1]^2$, and then compute its Alpha complex. Finally, we assign to each vertex a random function value in $[0,1]^n$, and compute the approximate multipersistence module (with $\delta = 0.7$) induced by the lower-star filtration of this random function. We repeated this experiment for several numbers of dimensions n, and show the result in Figure 24. As expected, there is an exponential scaling with respect to the dimension when δ is fixed.

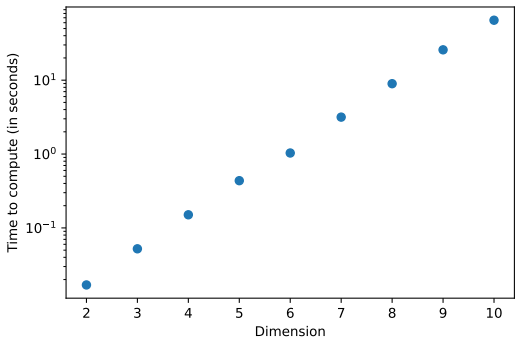

Figure 24: Running time w.r.t. the number of dimensions. Note that since the precision δ is fixed, the number of lines grows exponentially with the number of dimensions.

7.3.3. Comparison with `Rivet` when $n = 2$. As mentioned in the previous sections, `Rivet` [LW15] is a tool for computing minimal presentations of 2-multipersistence modules. We provide performance comparisons between `Rivet` and Algorithm 1 in the tables below.

In this first table, we focus on the noisy annulus of Section 7.2, with 80% of the points uniformly sampled on the annulus, and 20% are outliers in the square; and the same filtrations than in Section 7.2.

n	#simplices	Rivet	Rivet, peak RAM	Alg. 1, $\delta = 0.01$	Alg. 1, $\delta = 0.001$	Alg. 1, peak RAM
100	563	0.02s	9MB	0.004s	0.03s	140MB
1 000	5 943	0.49s	220MB	0.18s	0.69s	150MB
5 000	29 907	22.13s	5.41GB	4.15s	8.6s	180MB
7 000	41 879	59s	10.29GB	8.00s	14.39s	187MB
10 000	59 887	OOM	12.8GB	16s	26s	204MB
20 000	119 831	-	-	71s	85s	237MB

One can see that Algorithm 1 significantly outperforms Rivet both in terms of RAM usage and running time.

In our second table, we focus on the first time series from the data set Coffee, and processed it as in Section 7.2. Then, we used the Vietoris-Rips filtration as the first filtration, and density estimation as the second one.

threshold	#simplices	Rivet	Rivet, peak RAM	Alg. 1, $\delta = 0.01$	Alg. 1, peak RAM
0.1	9 961	0.28s	38MB	0.96s	170MB
0.2	35 620	0.79s	80MB	12s	201MB
0.3	71 230	1.45s	122MB	48s	281MB
0.4	114 144	2.6s	166MB	124s	396MB
0.5	168 513	5.1s	219MB	263s	576MB

One can see that while Rivet works remarkably well on flag complexes such as Vietoris-Rips, Algorithm 1 is still able to run in a reasonable amount of time.

8. Conclusion

In this article, we presented an algorithm for approximating any n-multipersistence module, whose complexity, running time, and approximation error can be controlled by user-defined parameters. We then showcased the performances of our method on synthetic and real data sets, and provided our code in an open-source package available at `https://gitlab.inria.fr/dloiseau/multipers`.

Several questions remain open for future work. While we proved that our candidate has bounded approximation error when approximating interval decomposable modules, can we prove that it is optimal (in some way) among the family of interval decomposable modules when the input is *not* interval decomposable? What are the properties of this output in the general case, and can it be used in practice instead of the original (non interval decomposable) module? Finally, can we use the stability results that are now becoming available for specific multipersistence modules [BL21] to infer confidence regions and convergence rates for our candidate decompositions?

Acknowledgments

The authors would like to thank Hannah Schreiber for her tremendous help in implementing our approximation and running our numerical experiments.

References

[ABE+21] Hideto Asashiba, Mickaël Buchet, Emerson Escolar, Ken Nakashima, and Michio Yoshiwaki. On interval decomposability of 2D persistence modules. *arXiv:1812.05261 [math]*, May 2021.

[AEK+17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[AENY19] Hideto Asashiba, Emerson Escolar, Ken Nakashima, and Michio Yoshiwaki. On approximation of 2D persistence modules by interval-decomposables. *arXiv:1911.01637 [math]*, November 2019.

[Bje20] Håvard Bjerkevik. Stability of higher-dimensional interval decomposable persistence modules. *arXiv:1609.02086 [cs, math]*, January 2020.

[BL18] Magnus Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & Geometric Topology*, 18(6):3133–3204, October 2018.

[BL21] Andrew Blumberg and Michael Lesnick. Stability of 2-parameter persistent homology. *arXiv:2010.09628 [cs, math]*, February 2021.

[BLO20] Magnus Botnan, Vadim Lebovici, and Steve Oudot. Local characterizations for decomposability of 2-parameter persistence modules. *arXiv:2008.02345 [math]*, August 2020.

[BLO22] Magnus Botnan, Vadim Lebovici, and Steve Oudot. On rectangle-decomposable 2-parameter persistence modules. *Discrete & Computational Geometry*, 2022.

[BOO21] Magnus Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions. *arXiv:2107.06800 [cs, math]*, July 2021.

[Bub15] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3):77–102, 2015.

[Car09] Gunnar Carlsson. Topology and data. *Bulletin of The American Mathematical Society*, 46:255–308, 2009.

[CB20] Mathieu Carrière and Andrew Blumberg. Multiparameter persistence image for topological machine learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2020)*, pages 22432–22444. Curran Associates, Inc., 2020.

[CCI+20] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. PersLay: a neural network layer for persistence diagrams and new graph topological signatures. In *23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, pages 2786–2796. PMLR, 2020.

[CCO17] Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced Wasserstein kernel for persistence diagrams. In *34th International Conference on Machine Learning (ICML 2017)*, volume 70, pages 664–673. PMLR, 2017.

[CdGO16] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer International Publishing, 2016.

[CFK+19] René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, and Bei Wang. A kernel for multi-parameter persistent homology. *Computers & Graphics: X*, 2:100005, 2019.

[Che15] Chen, Yanping and Keogh, Eamonn and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo. The UCR time series classification archive, 2015.

[CO19] Jérémy Cochoy and Steve Oudot. Decomposition of exact pfd persistence bimodules. *Discrete & Computational Geometry*, pages 1–39, 2019.

[CSEM06] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *22nd Annual Symposium on Computational Geometry (SoCG 2006)*, pages 119–126. Association for Computing Machinery, 2006.

[dMV11] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, 2011.

[DW22] Tamal Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[DX21] Tamal Dey and Cheng Xin. Rectangular approximation and stability of 2-parameter persistence modules. *arXiv:2108.07429 [cs, math]*, August 2021.

[DX22] Tamal Dey and Cheng Xin. Generalized persistence algorithm for decomposing multiparameter persistence modules. *Journal of Applied and Computational Topology*, 2022.

[EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[KLO19] Michael Kerber, Michael Lesnick, and Steve Oudot. Exact computation of the matching distance on 2-parameter persistence modules. In *35th International Symposium on Computational Geometry (SoCG 2019)*, volume 129, pages 46:1–46:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

[KN20] Michael Kerber and Arnur Nigmetov. Efficient approximation of the matching distance for 2-parameter persistence. *arXiv:1912.05826 [cs]*, March 2020.

[Lan18] Claudia Landi. The rank invariant stability via interleavings. In *Research in Computational Topology*, pages 1–10. Springer, 2018.

[Les15] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.

[LW15] Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules. *arXiv:1512.00180 [cs, math]*, December 2015.

[Mil20] Ezra Miller. Homological algebra of modules over posets. *arXiv:2008.00063 [math]*, August 2020.

[MO15] Clément Maria and Steve Oudot. Zigzag persistence via reflections and transpositions. In *26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, pages 181–199. Society for Industrial and Applied Mathematics, 2015.

[Mun84] James Munkres. *Elements of algebraic topology*. CRC Press, 1984.

[Oud15] Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2015.

[RHBK15] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 4741–4748. IEEE Computer Society, 2015.

[Vip20a] Oliver Vipond. Local equivalence of metrics for multiparameter persistence modules. *arXiv:2004.11926 [math]*, April 2020.

[Vip20b] Oliver Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21(61):1–38, 2020.