Multi-Parameter Topological Data Analysis with MMA

Foundations of Computational Geometry and Topology @ ICERM
May 18th, 2026

Mathieu Carrière
DataShape, Centre Inria d'Université Côte d'Azur
joint work with A. Blumberg, D. Loiseaux, L. Andrianirina

1. **Background: Persistence Theory**
2. **MMA: Multi-parameter Module Approximation**
3. **MMA Application: Machine Learning**
4. **MMA Application: Clustering**

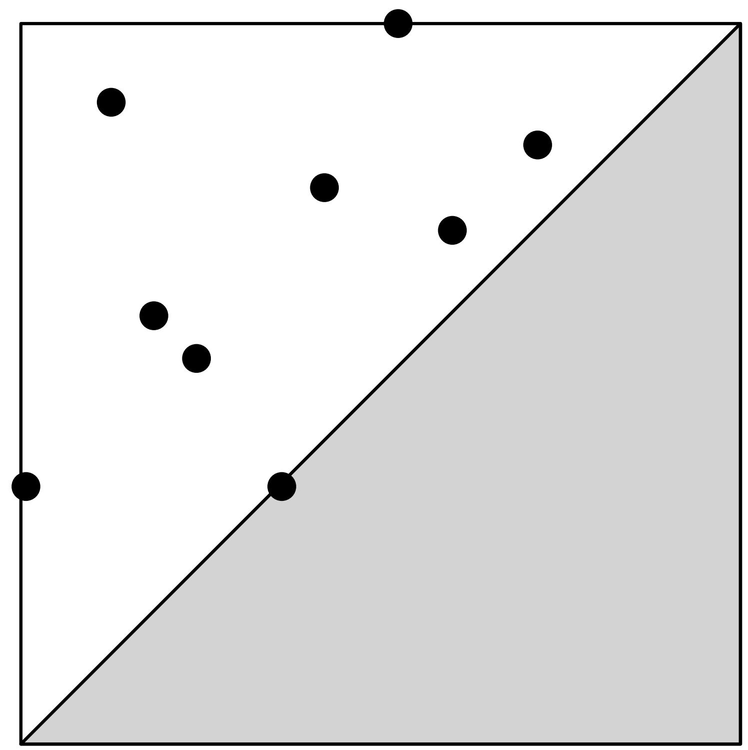
1. **Background: Persistence Theory**
2. MMA: Multi-parameter Module Approximation
3. MMA Application: Machine Learning
4. MMA Application: Clustering

What is Single Parameter Persistence?

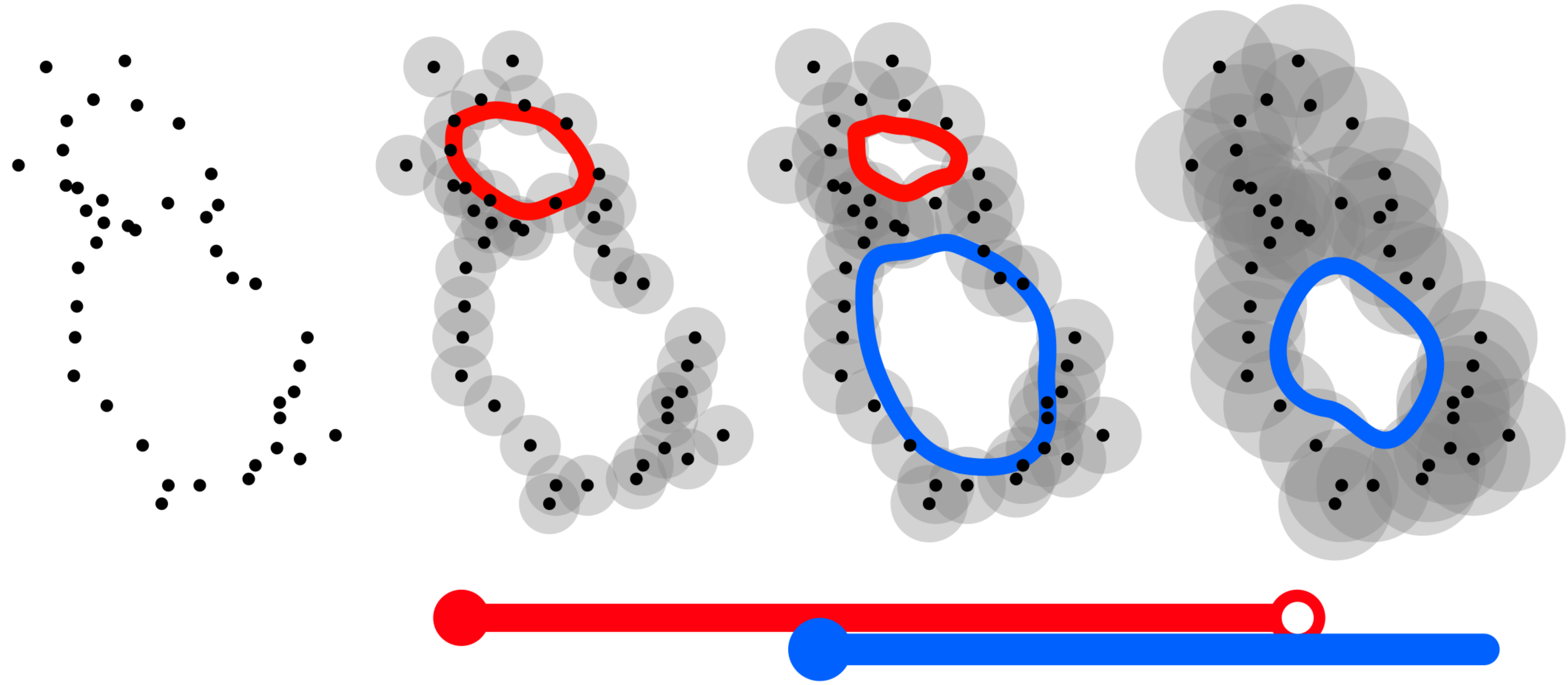
Def: A finite multiset $D(M)$ in $\{(x,y) \in \mathbb{R} \times \mathbb{R} \cup \{+\infty\} \;:\; x \leq y\}$ computed from a *single parameter persistence module* M.

$$M = H_*(X_{\alpha_1}) \to \cdots \to H_*(X_{\alpha_M})$$

Each point represents a topological feature (connected component, loop, cavity, etc.) and the coordinates are proxies for its size.



What is Single Parameter Persistence?



What is Multi-parameter Persistence?

Multiparameter persistence is the study of *multiparameter persistence modules*.

Def: A multiparameter persistence module is a functor from $\mathbb{R}^n$ to $\mathrm{Vect}_{\mathbf{k}}$, the vector spaces over field $\mathbf{k}$.

$$M = \{\{M_x \,:\, x \in \mathbb{R}^n\}, \{\Phi_{x,y} : M_x \to M_y\}_{x,y\in\mathbb{R}^n, x\leq y}$$

$$\text{where } x \leq y \iff x_i \leq y_i \;\forall 1 \leq i \leq n$$

Multiparameter persistence modules can be compared with the *interleaving distance* d_I.

Def: The interleaving distance d_I between multiparameter persistence modules M, M' is defined as:

$$d_I(M, M') = \inf\{\epsilon > 0 \,:\, M, M' \text{ are } \epsilon\text{-interleaved}\}$$

where M, M' are ϵ-interleaved if $\exists A_x : M_x \to M'_{x+\epsilon}, B_x : M'_x \to M_{x+\epsilon}$ s.t.

$$\Phi_{x,x+2\epsilon} = B_{x+\epsilon} \circ A_x \text{ and } \Psi_{x,x+2\epsilon} = A_{x+\epsilon} \circ B_x, \;\forall x \in \mathbb{R}^n$$

What is Multi-parameter Persistence?

Multiparameter persistence modules often come from applying the homology functor to *multiparameter filtrations*.

Def: A multiparameter filtration $\mathcal{F}$ of a topological space X is a family of subsets of X indexed by $\mathbb{R}^n$ such that:

$$\forall x, y \in \mathbb{R}^n, x \leq y \Longrightarrow \mathcal{F}_x \subseteq \mathcal{F}_y$$

Multiparameter filtration

$$\begin{array}{ccccc}
X_{1,3} & \subseteq & X_{2,3} & \subseteq & X_{3,3} \\
\cup\!\mid & & \cup\!\mid & & \cup\!\mid \\
X_{1,2} & \subseteq & X_{2,2} & \subseteq & X_{3,2} \\
\cup\!\mid & & \cup\!\mid & & \cup\!\mid \\
X_{1,1} & \subseteq & X_{2,1} & \subseteq & X_{3,1}
\end{array}$$

Associated multiparameter persistence module

$$\begin{array}{ccccc}
H_*(X_{1,3}) & \longrightarrow & H_*(X_{2,3}) & \longrightarrow & H_*(X_{3,3}) \\
\uparrow & & \uparrow & & \uparrow \\
H_*(X_{1,2}) & \longrightarrow & H_*(X_{2,2}) & \longrightarrow & H_*(X_{3,2}) \\
\uparrow & & \uparrow & & \uparrow \\
H_*(X_{1,1}) & \longrightarrow & H_*(X_{2,1}) & \longrightarrow & H_*(X_{3,1})
\end{array}$$

What is Multi-parameter Persistence?

Multiparameter persistence modules can also be compared with the *rank invariant* and the *fibered barcode*.

Def: The rank invariant of a multiparameter persistence module M is:

$$\mathrm{rk}(M): \begin{cases} \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{N} \\ \quad (x,y) \mapsto \dim(\mathrm{im}(M_x \to M_y)) := \beta^{x,y} \end{cases}$$

$\beta^{x,y}$ is called the *multiparameter persistent Betti number*.

Def: The fibered barcode of a multiparameter persistence module M is:

$$\mathcal{L}(M): \ell \mapsto D(M|_\ell)$$

where ℓ is a line in $\mathbb{R}^n$, and $D(M|_\ell)$ stands for the barcode/persistence diagram obtained by restricting M to ℓ.

What is Multi-parameter Persistence?

Finally, multiparameter persistence modules can be *decomposed* into a direct sum of summands.

$$M \simeq \oplus_i M_i,$$

where M_i cannot be decomposed, i.e., is *indecomposable*.

Contrary to 1D persistence, indecomposable summands in $\mathbb{R}^n$ are **not** necessarily made of *indicator modules*, i.e., modules that contain only single copies of the field $\mathbf{k}$. Among all indecomposable summands, the ones called *intervals* are of particular interest.

Def: An interval module is an indicator module supported on a subset $\mathcal{I} \subseteq \mathbb{R}^n$ which satisfies:

(i) $\mathcal{I} \neq \emptyset$,

(ii) $\forall x, y \in \mathcal{I}$, if $\exists z$ s.t. $x \leq z \leq y$ then $z \in \mathcal{I}$,

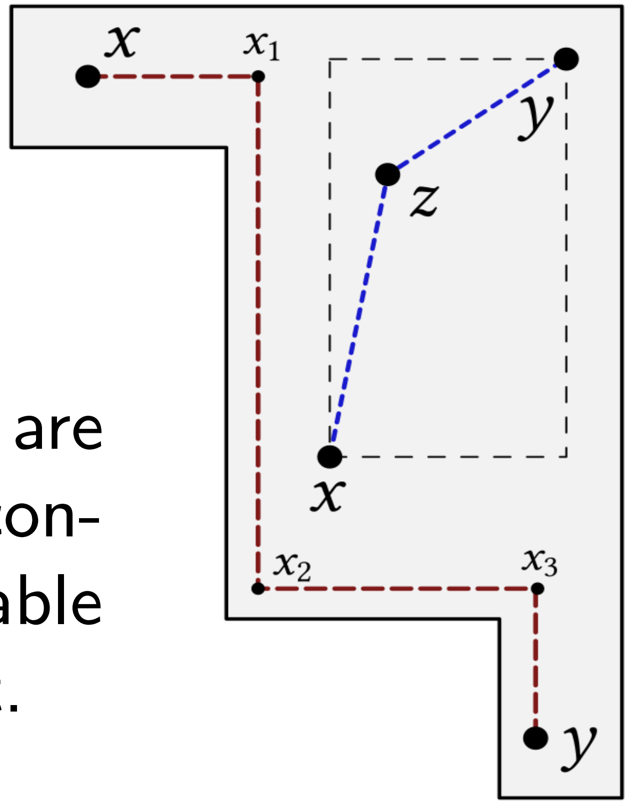
(iii) $\forall x, y \in \mathcal{I}$, $\exists x_0 = x, x_1, \ldots, x_{n-1}, x_n = y$ s.t. $x_i \leq x_{i+1}$ or $x_{i+1} \leq x_i$.

What is Multi-parameter Persistence?

Finally, multiparameter persistence modules can be *decomposed* into a direct sum of summands.

$$M \simeq \oplus_i M_i,$$

where M_i cannot be decomposed, i.e., is *indecomposable*.



Contrary to 1D persistence, indecomposable summands in $\mathbb{R}^n$ are **not** necessarily made of *indicator modules*, i.e., modules that contain only single copies of the field $\mathbf{k}$. Among all indecomposable summands, the ones called *intervals* are of particular interest.

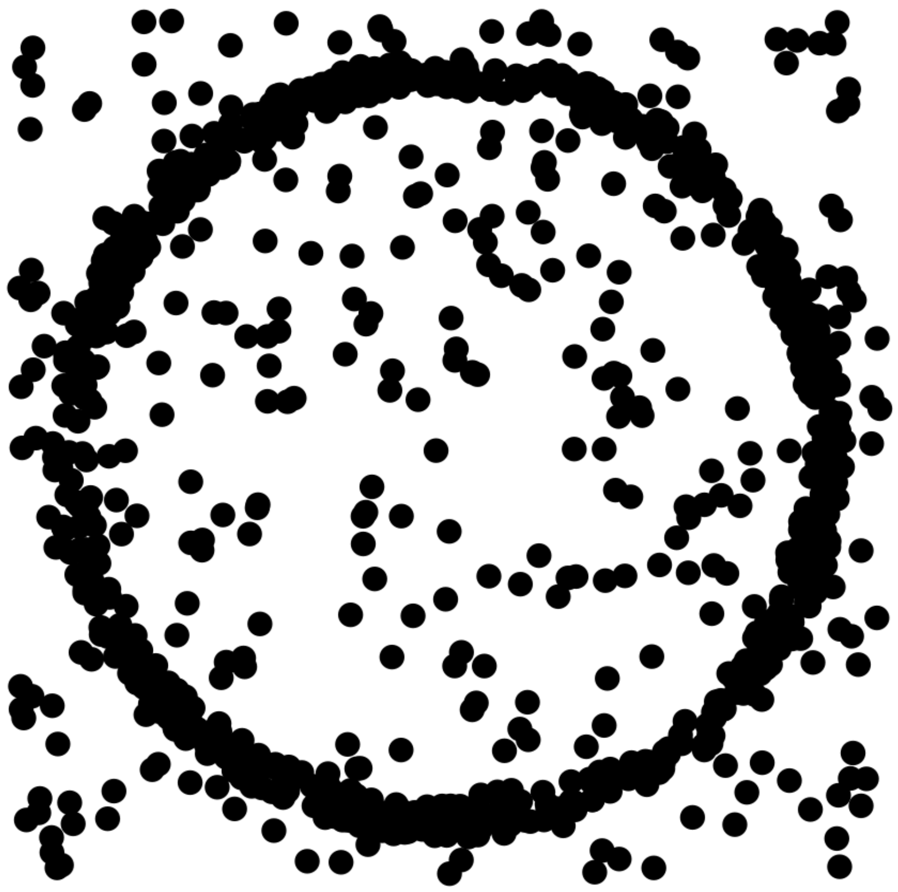
Def: An interval module is an indicator module supported on a subset $\mathcal{I} \subseteq \mathbb{R}^n$ which satisfies:

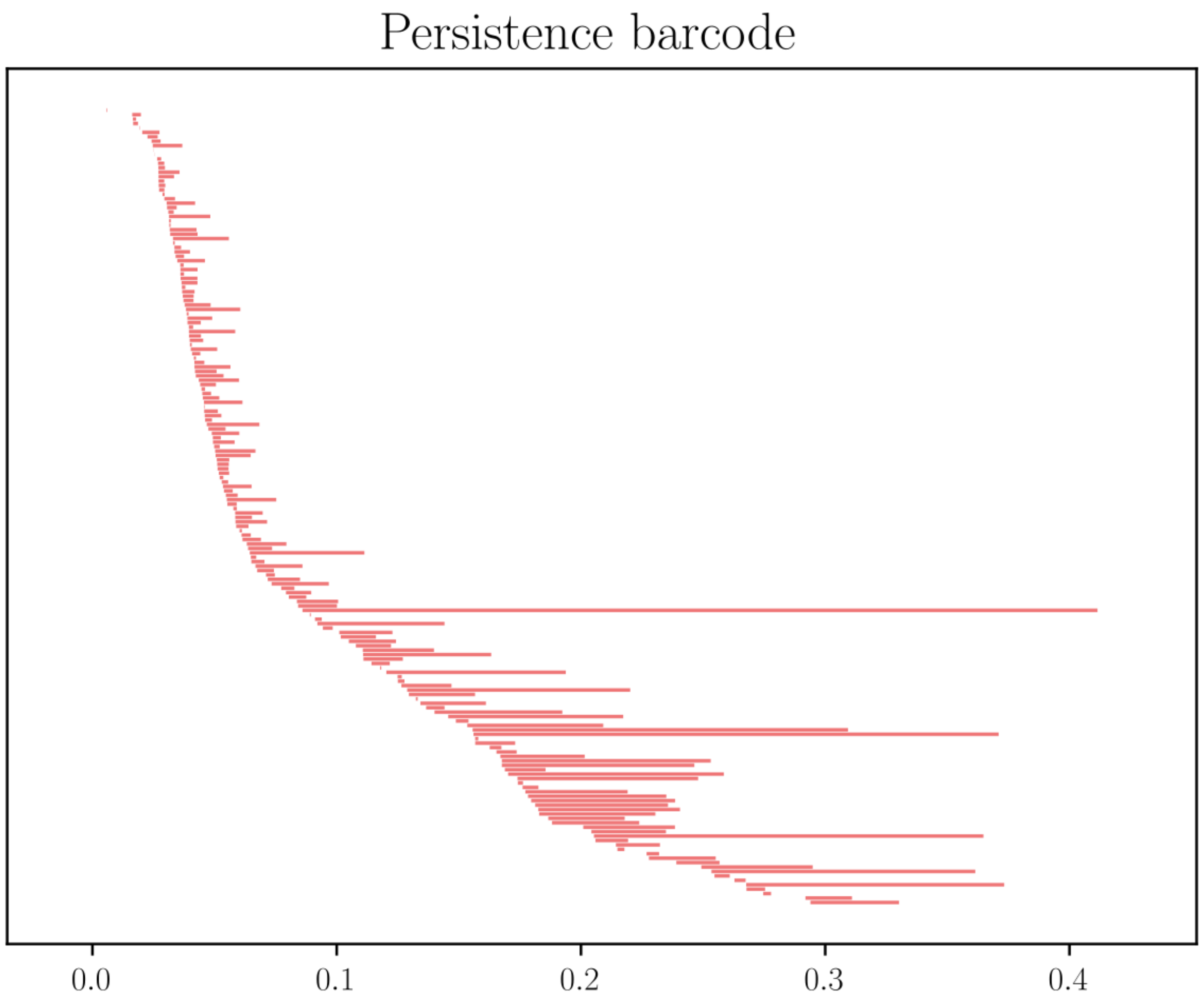
(i) $\mathcal{I} \neq \emptyset$,

(ii) $\forall x, y \in \mathcal{I}$, if $\exists z$ s.t. $x \leq z \leq y$ then $z \in \mathcal{I}$,

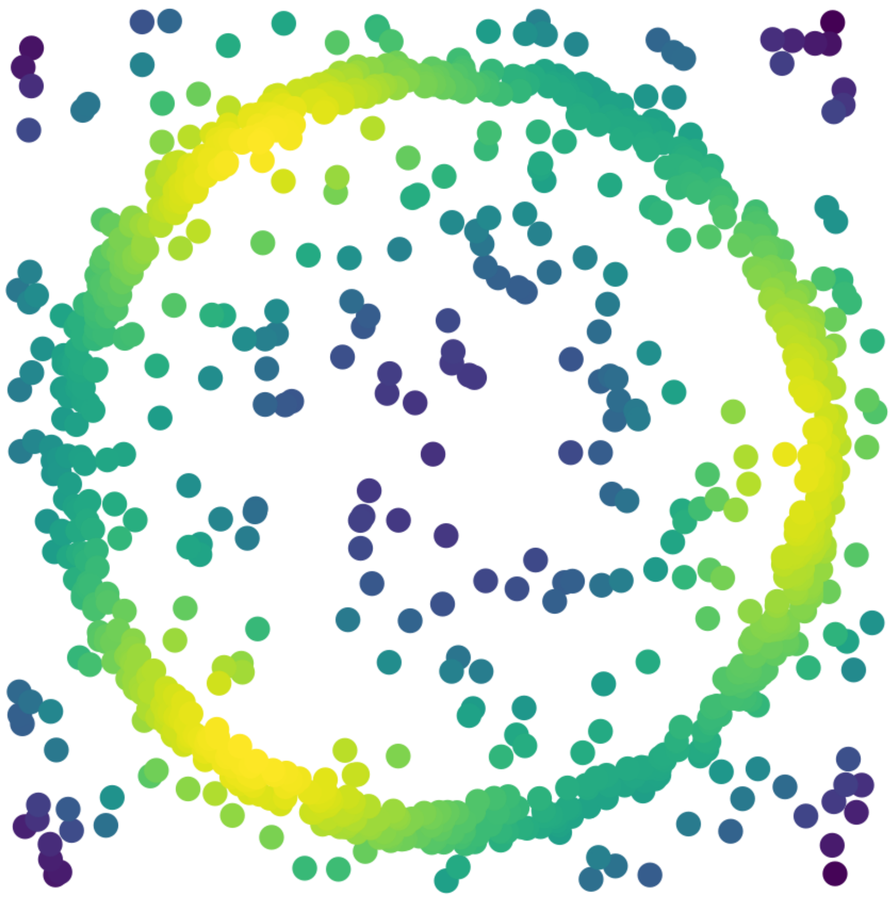
(iii) $\forall x, y \in \mathcal{I}$, $\exists x_0 = x, x_1, \ldots, x_{n-1}, x_n = y$ s.t. $x_i \leq x_{i+1}$ or $x_{i+1} \leq x_i$.

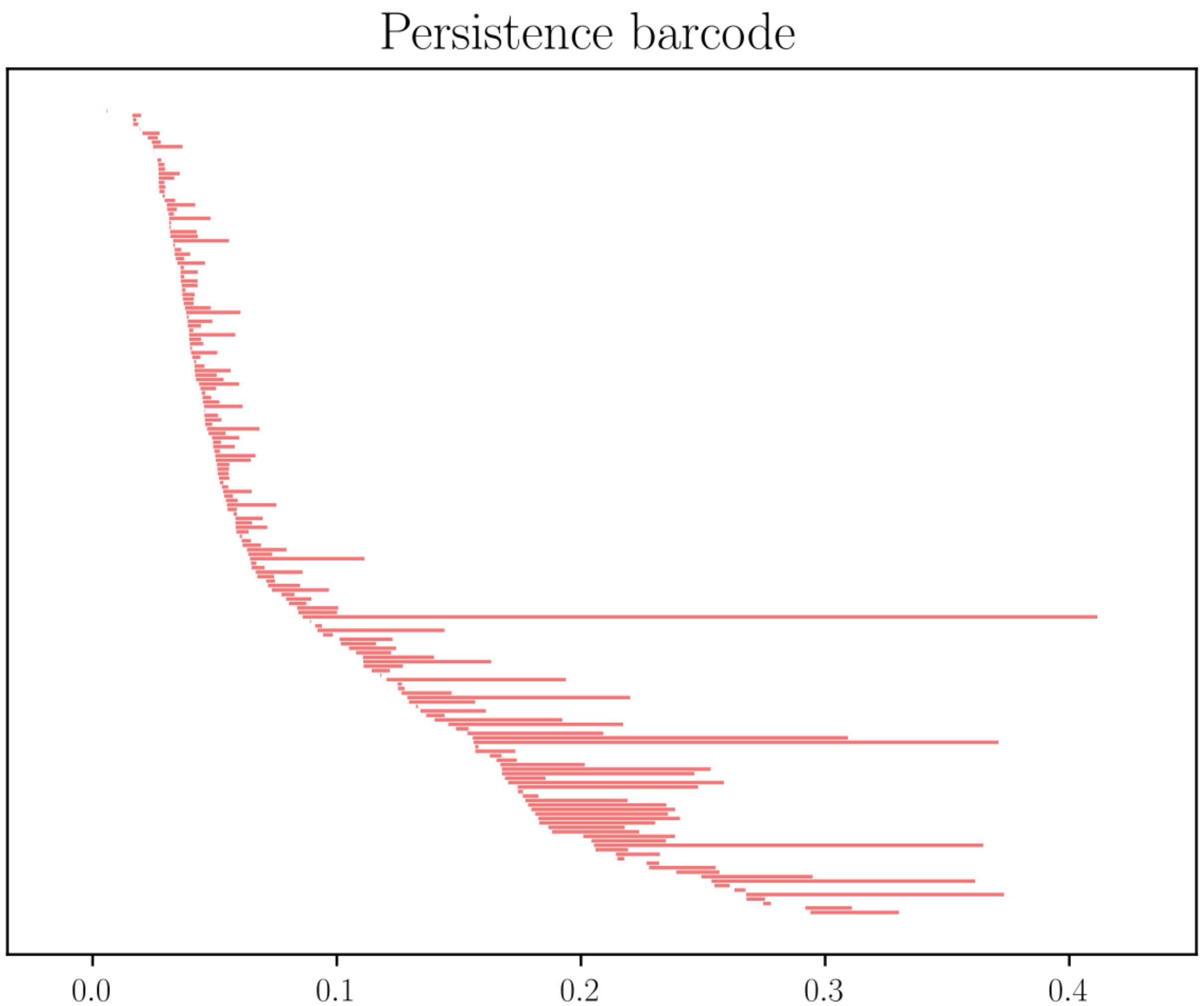
What is Multi-parameter Persistence?



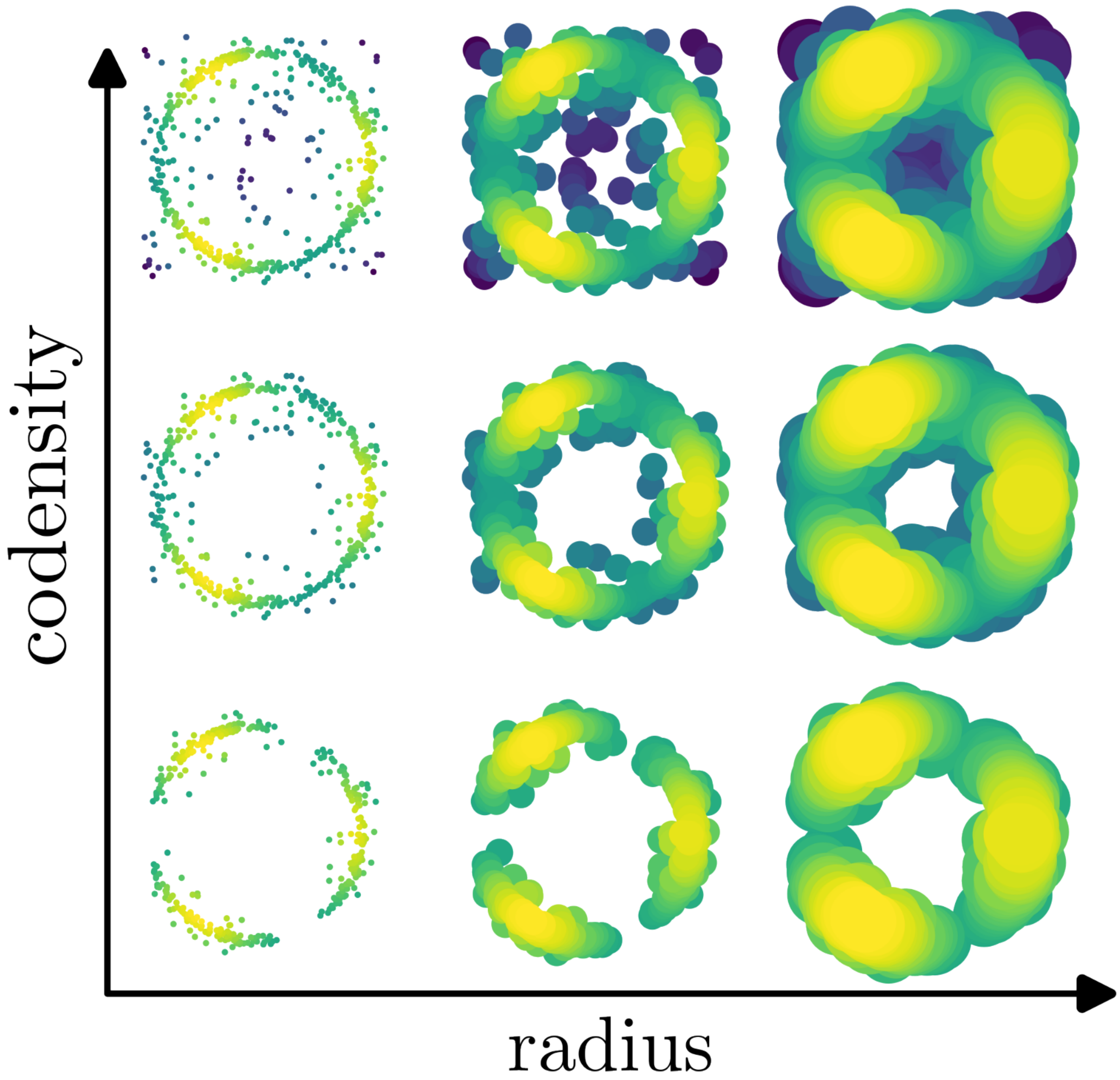


What is Multi-parameter Persistence?





What is Multi-parameter Persistence?



1. Background: Persistence Theory

2. MMA: Multi-parameter Module Approximation

3. MMA Application: Machine Learning

4. MMA Application: Clustering

[Loiseaux, C., Blumberg, **Multi-Parameter Module Approximation: An Efficient and Interpretable Invariant for Multi-Parameter Persistence Modules with Guarantees.** JACT, 2025]

What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

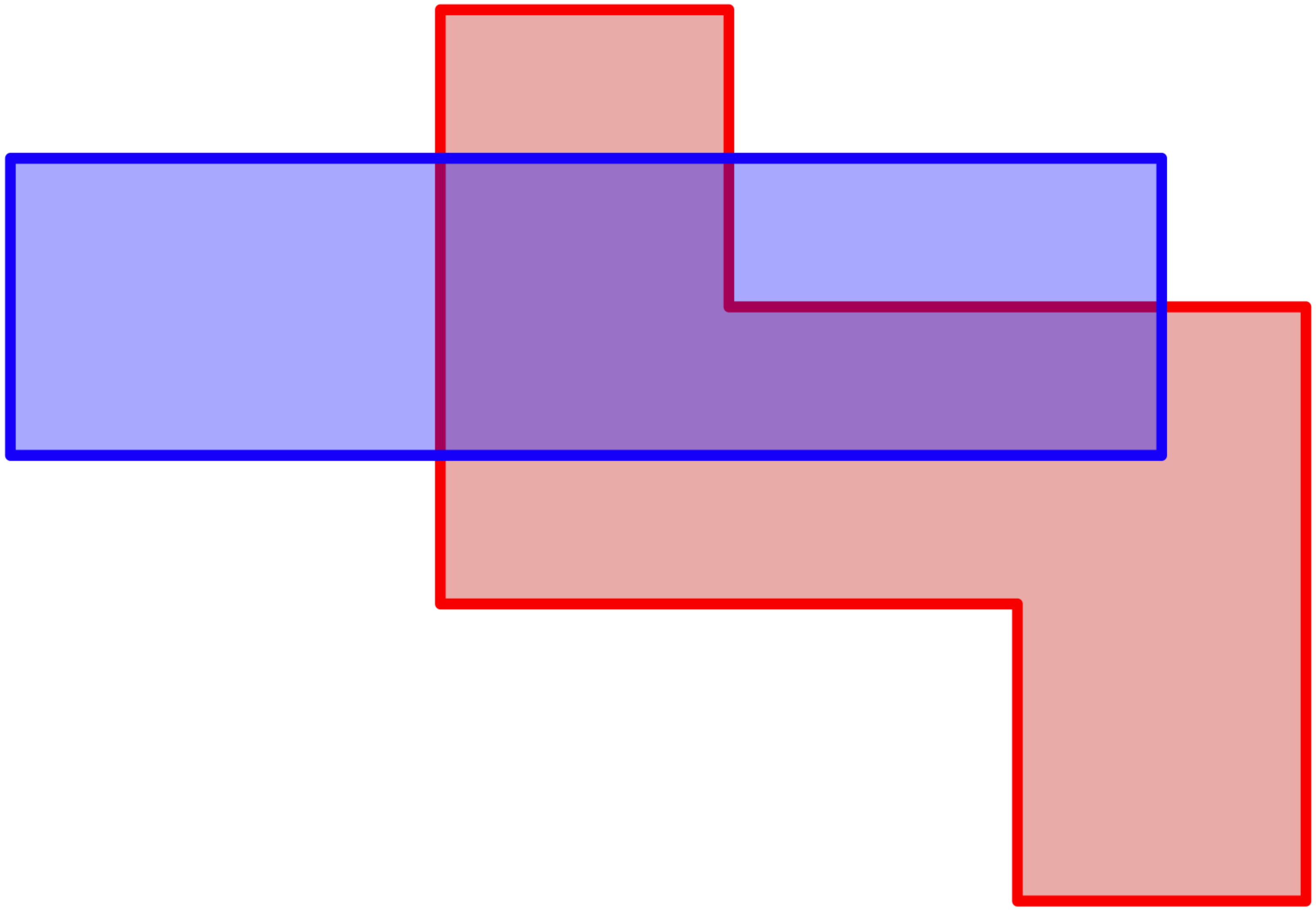
What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

Idea:



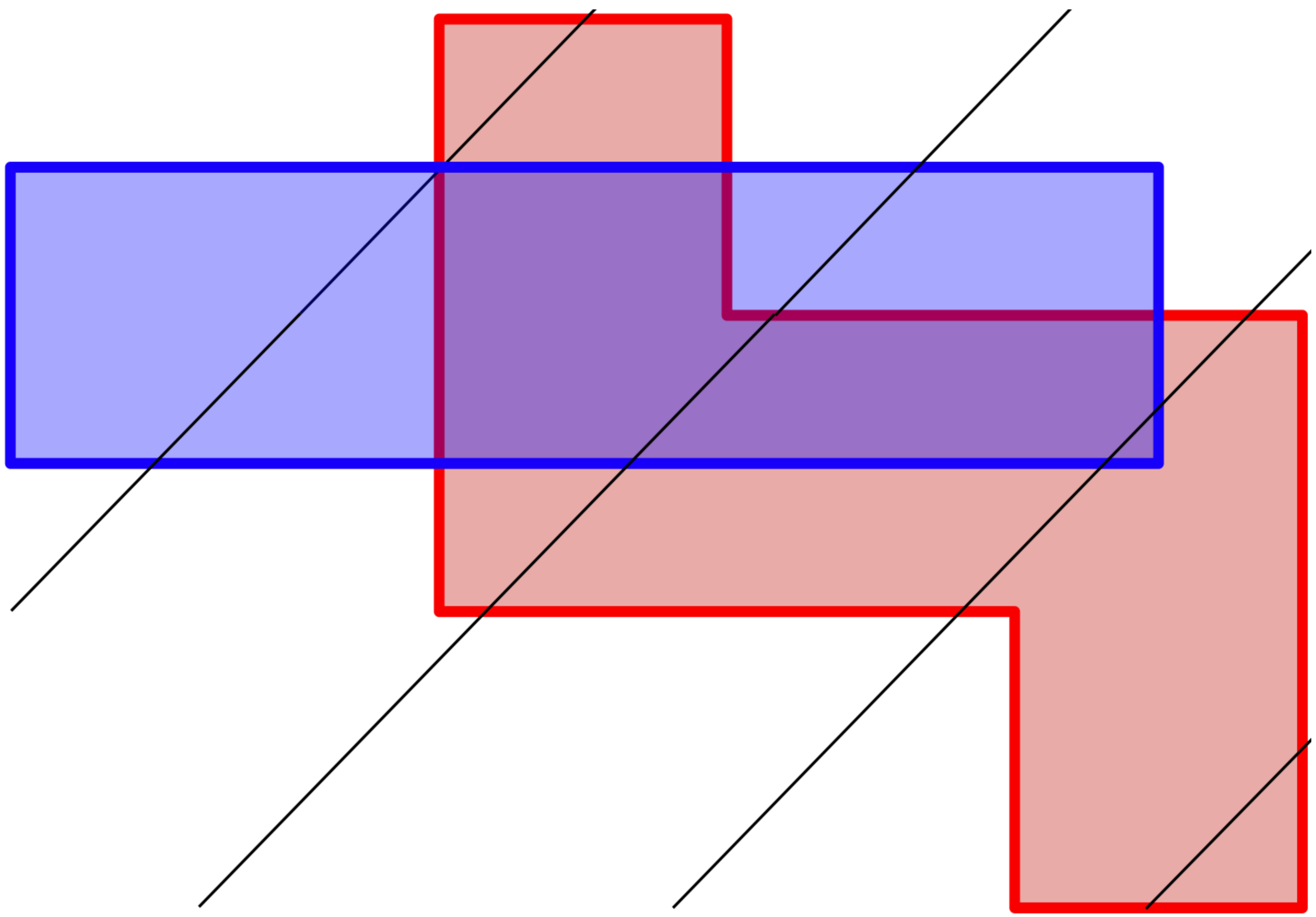
What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

Idea:



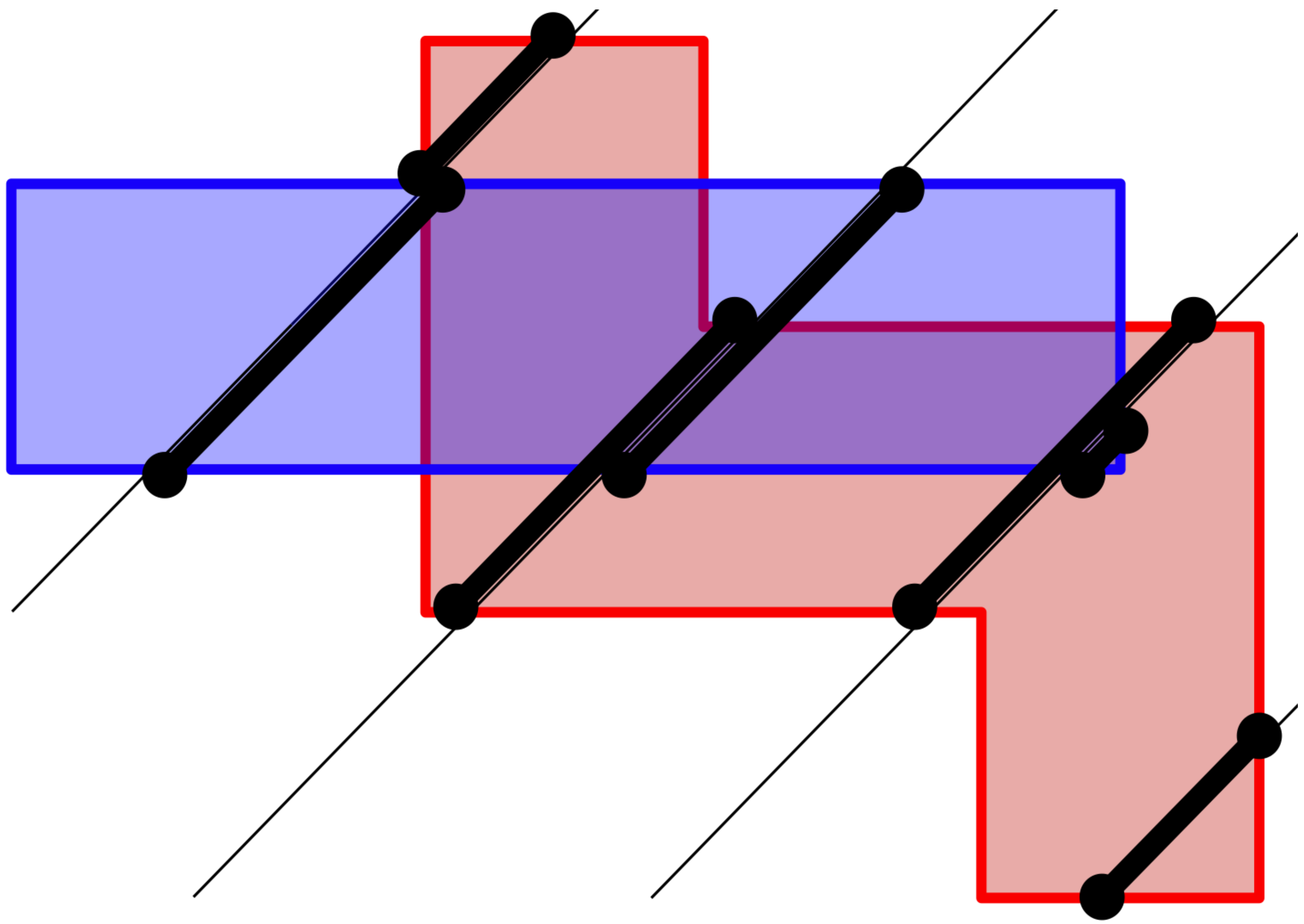
What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

Idea:



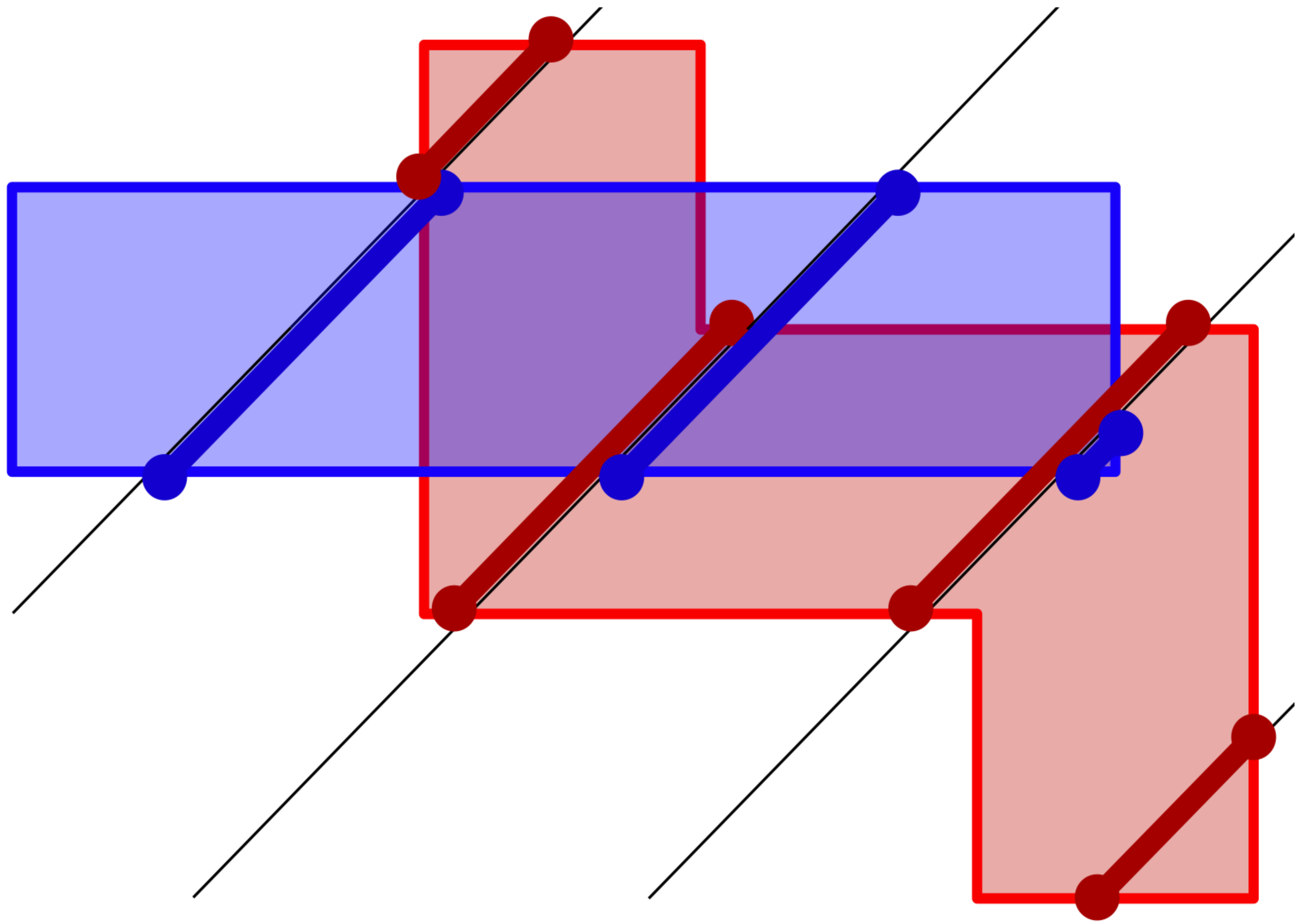
What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

Idea:



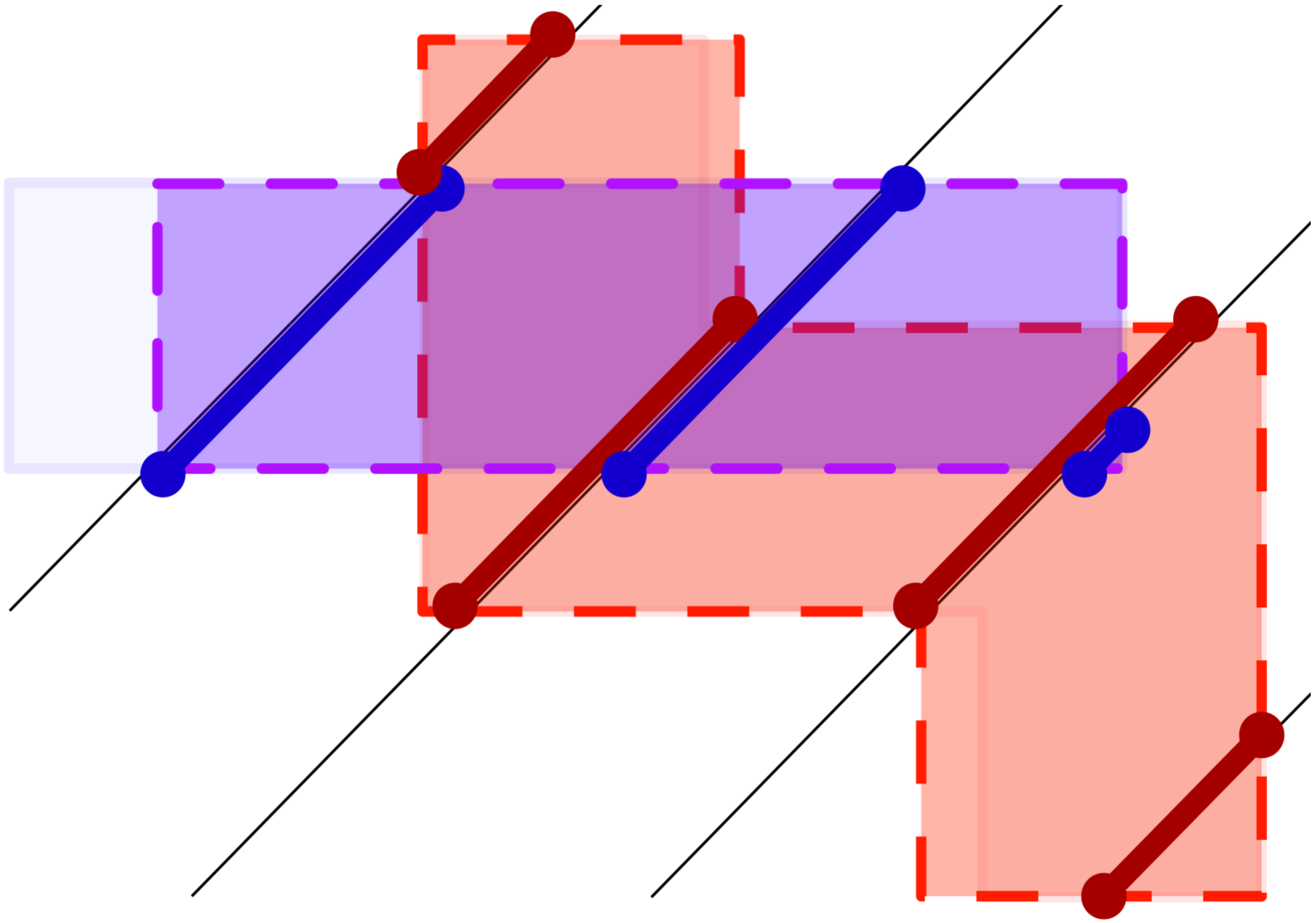
What is MMA?

Goal: Compute an **approximate decomposition** into interval summands s.t.:

each summand is (roughly) the *lifetime* of a cycle,

the fibered barcode is *preserved* (on a given set of lines L).

Idea:



Compatible Matchings and Candidate Decompositions

Compatible Matchings and Candidate Decompositions

Def: A matching is a family of injective maps:

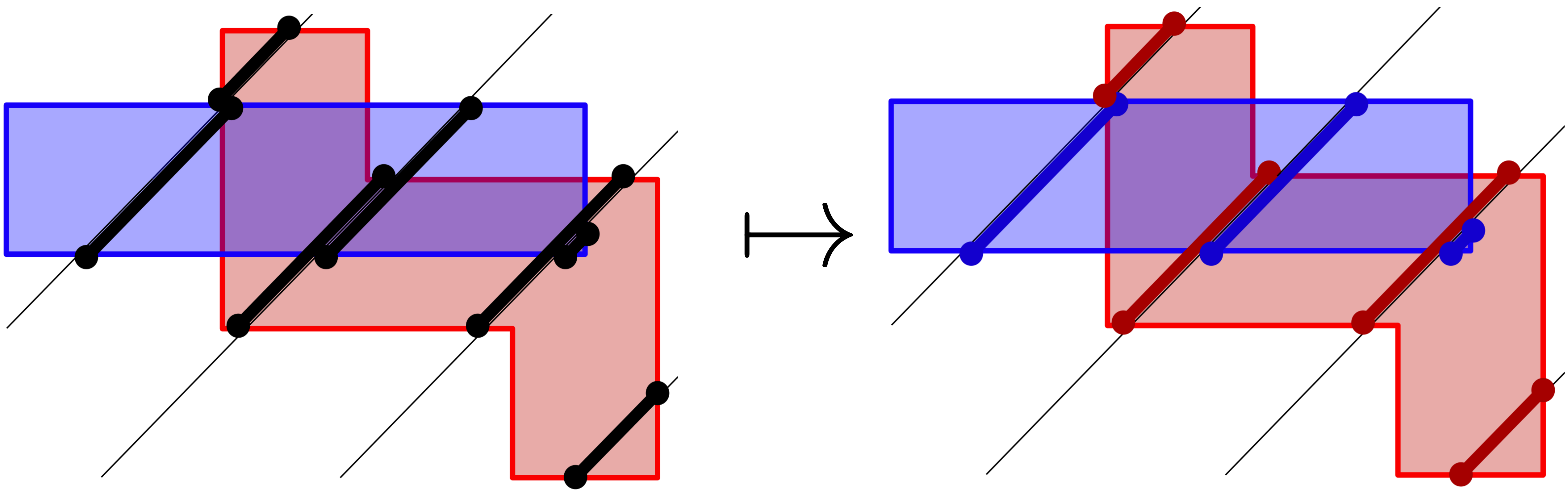
$$\{\mathrm{m}_i : D(M|_{\ell_i}) \to D(M|_{\ell_{i+1}})\}_i.$$

Compatible Matchings and Candidate Decompositions

Def: A matching is a family of injective maps:

$$\{\mathrm{m}_i : D(M|_{\ell_i}) \to D(M|_{\ell_{i+1}})\}_i.$$

Ex: Induced matchings:

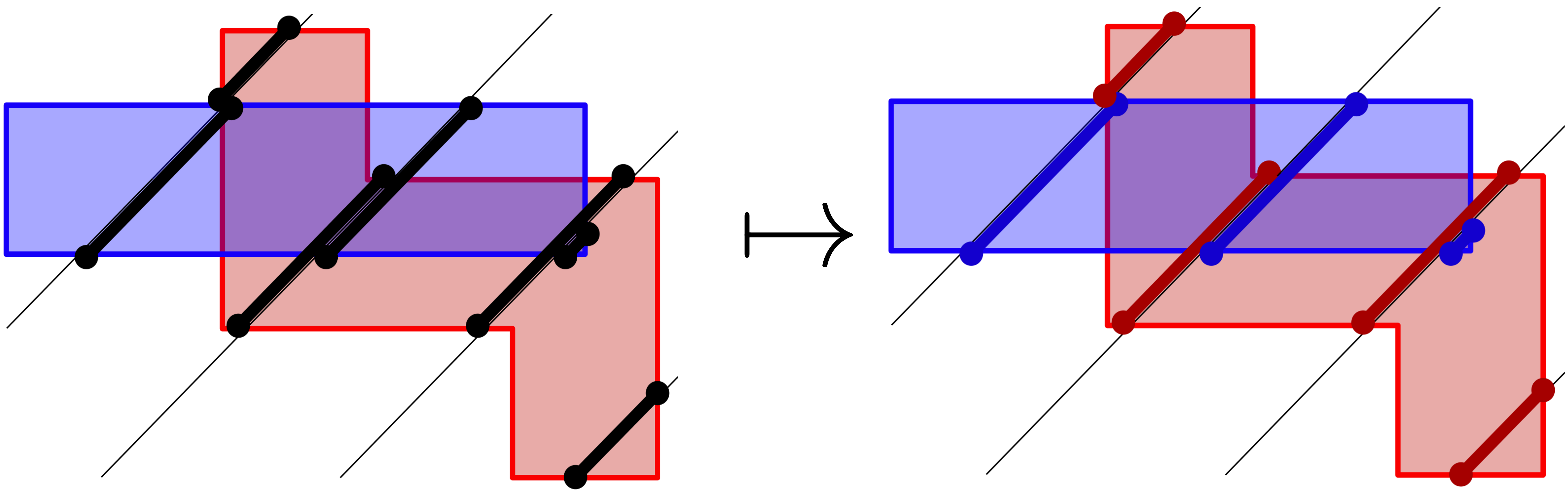


Compatible Matchings and Candidate Decompositions

Def: A matching is a family of injective maps:

$$\{\mathrm{m}_i : D(M|_{\ell_i}) \to D(M|_{\ell_{i+1}})\}_i.$$

Ex: Induced matchings:



In practice: vineyards, Wasserstein distance, etc.

Compatible Matchings and Candidate Decompositions

Def: A matching m of M is *compatible* if there exists N s.t.

(i) N is interval decomposable,

(ii) $D(M|_\ell) = D(N|_\ell), \forall \ell \in L$, and

(iii) m is induced by N.

Moreover, N is called a *candidate decomposition* of M.

Compatible Matchings and Candidate Decompositions

Def: A matching m of M is *compatible* if there exists N s.t.

(i) N is interval decomposable,

(ii) $D(M|_\ell) = D(N|_\ell), \forall \ell \in L$, and

(iii) m is induced by N.

Moreover, N is called a *candidate decomposition* of M.

Th: Let m be a compatible matching of a module M. Let N be its corresponding candidate decomposition. Then:

$$d_b(\mathcal{L}(M), \mathcal{L}(N)) \to 0 \text{ as } |L| \to +\infty.$$

Moreover, if M is interval decomposable, and $|L|$ is large enough:

$$d_I(M, N) = 0.$$

Compatible Matchings and Candidate Decompositions

Def: A matching m of M is *compatible* if there exists N s.t.

(i) N is interval decomposable,

(ii) $D(M|_\ell) = D(N|_\ell), \forall \ell \in L$, and

(iii) m is induced by N.

Moreover, N is called a *candidate decomposition* of M.

Th: Let m be a compatible matching of a module M. Let N be its corresponding candidate decomposition. Then:

$$d_b(\mathcal{L}(M), \mathcal{L}(N)) \to 0 \text{ as } |L| \to +\infty.$$

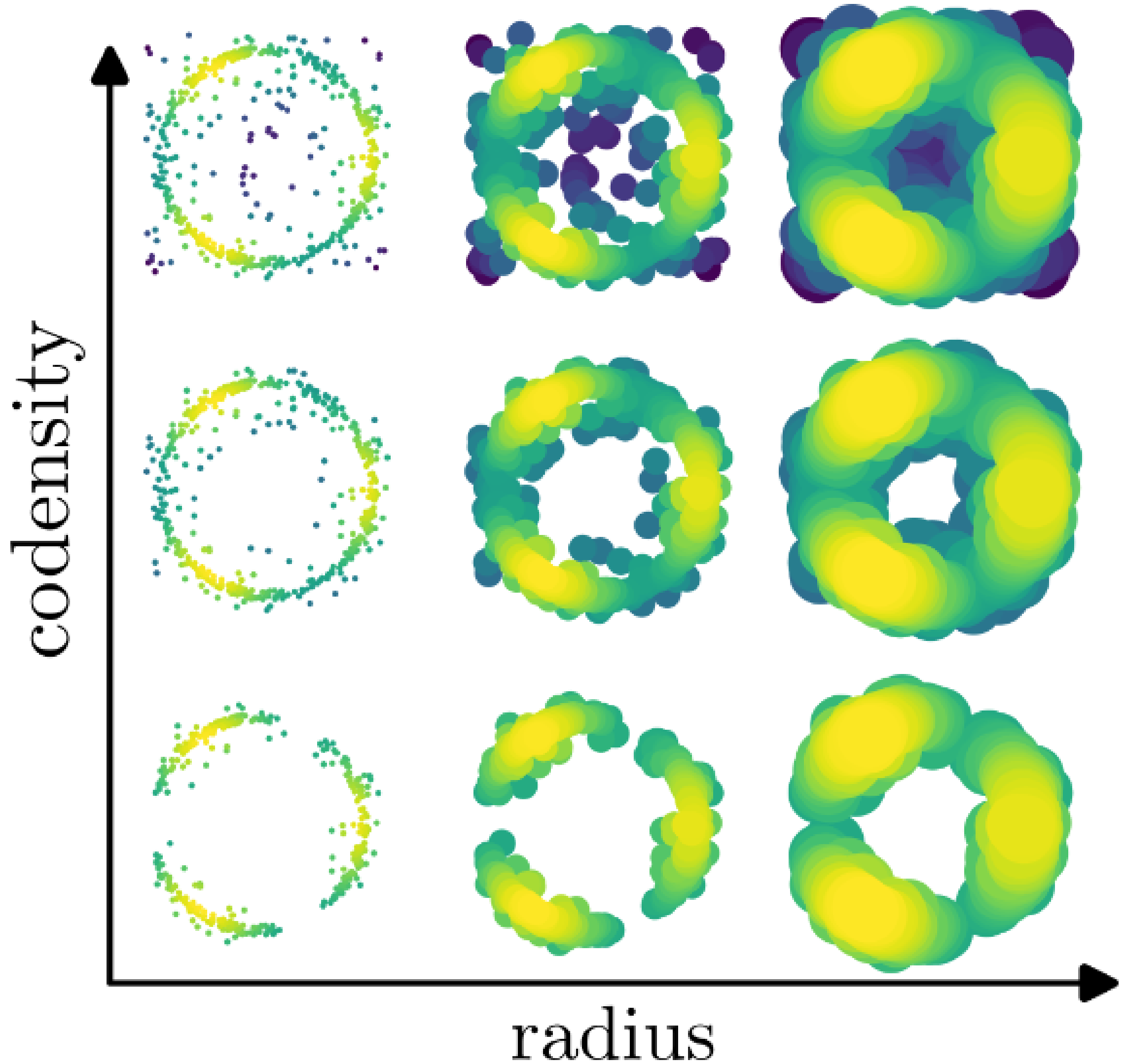
Moreover, if M is interval decomposable, and $|L|$ is large enough:

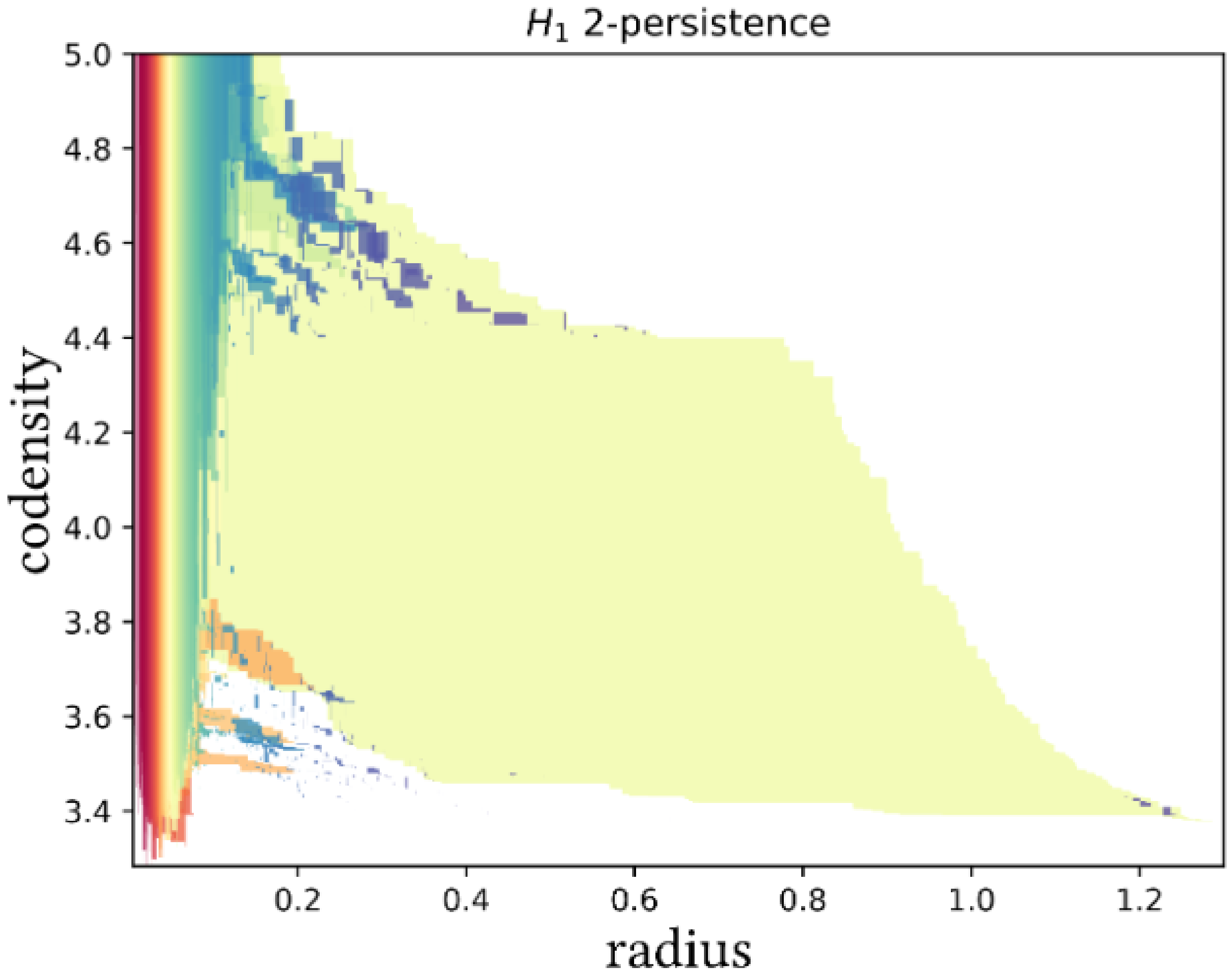
$$d_I(M, N) = 0.$$

In practice: usual matchings are compatible under genericity assumptions.

Algorithms

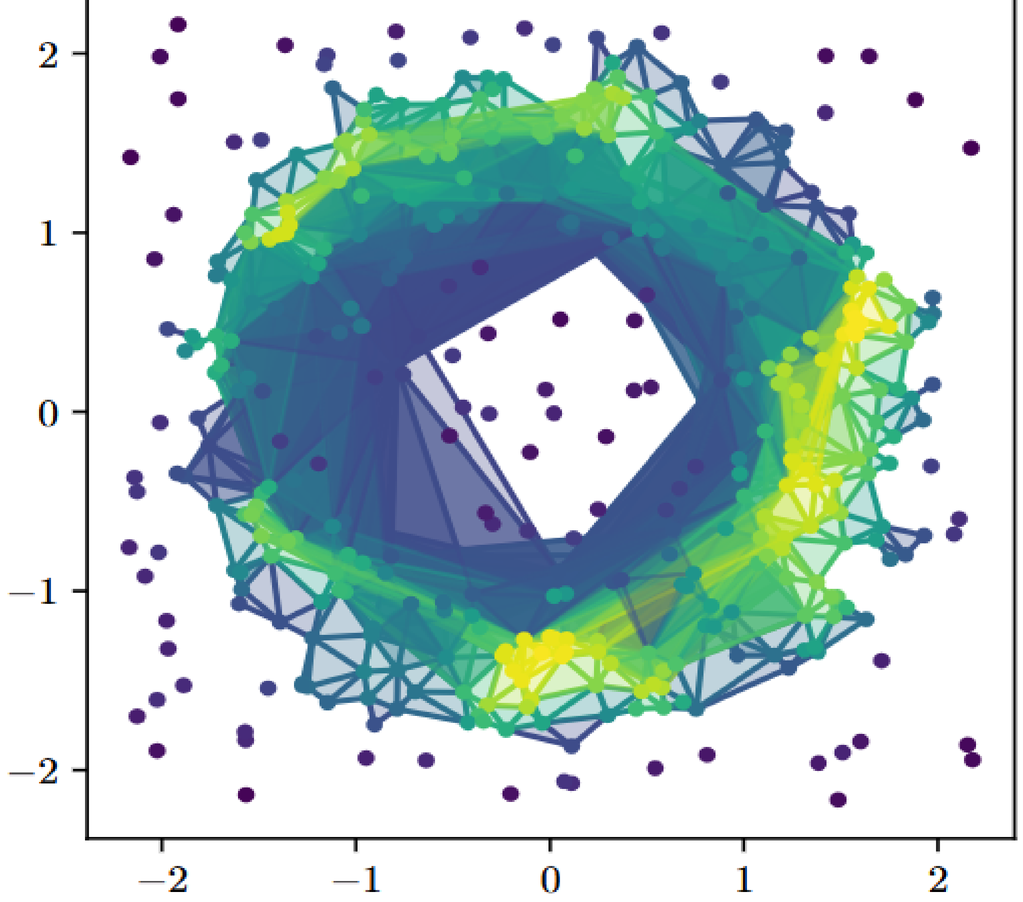
In practice: Candidate decompositions can be computed with `multipers`.

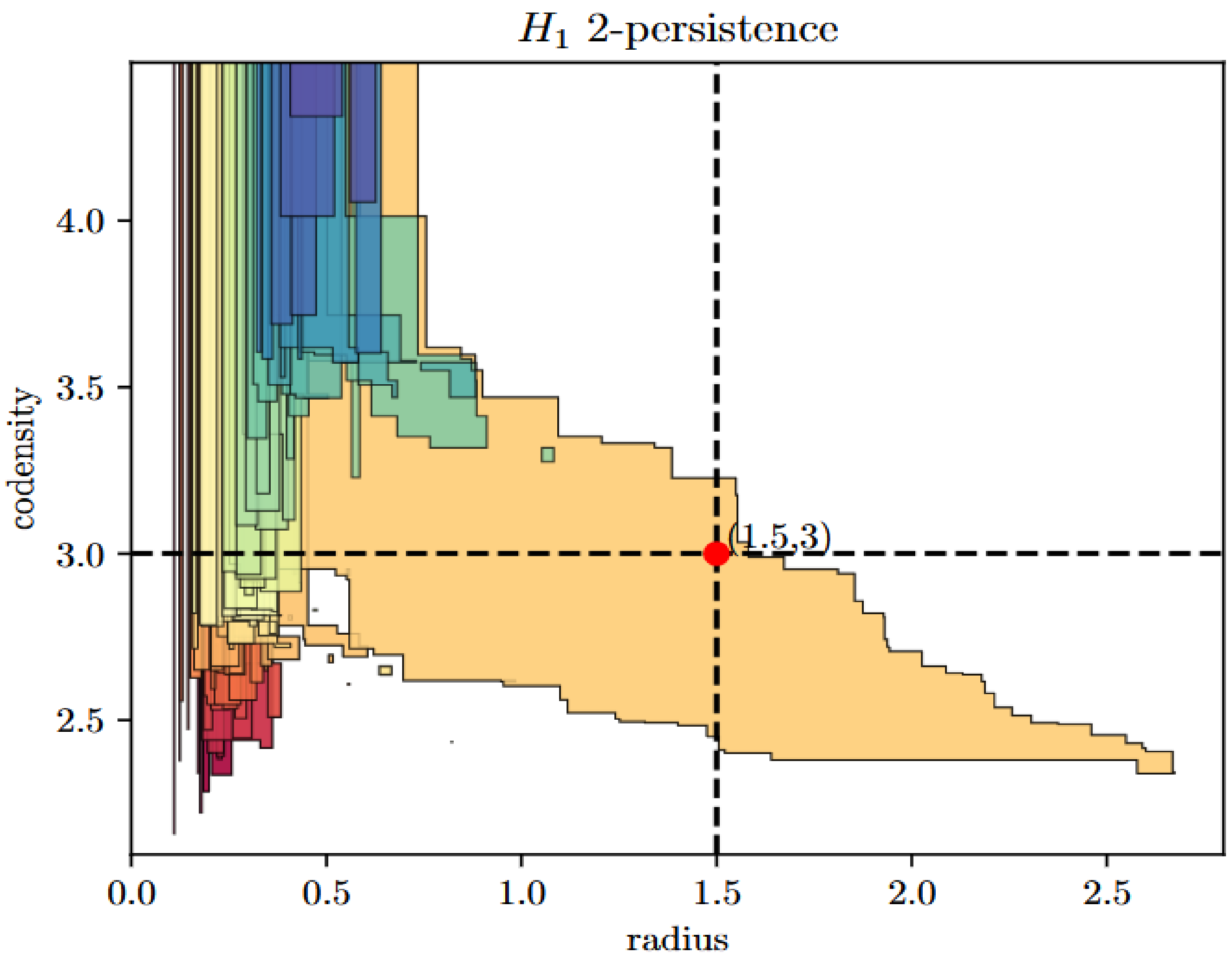




Algorithms

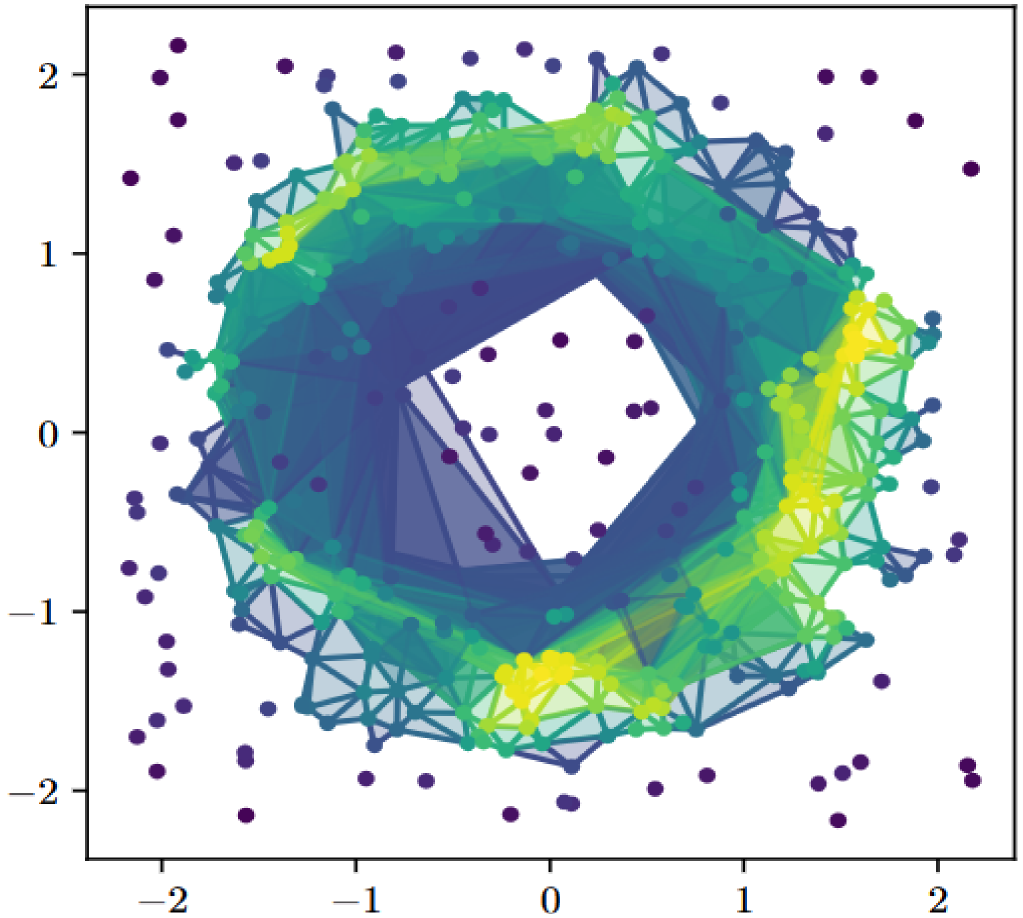
In practice: Candidate decompositions can be computed with `multipers`.

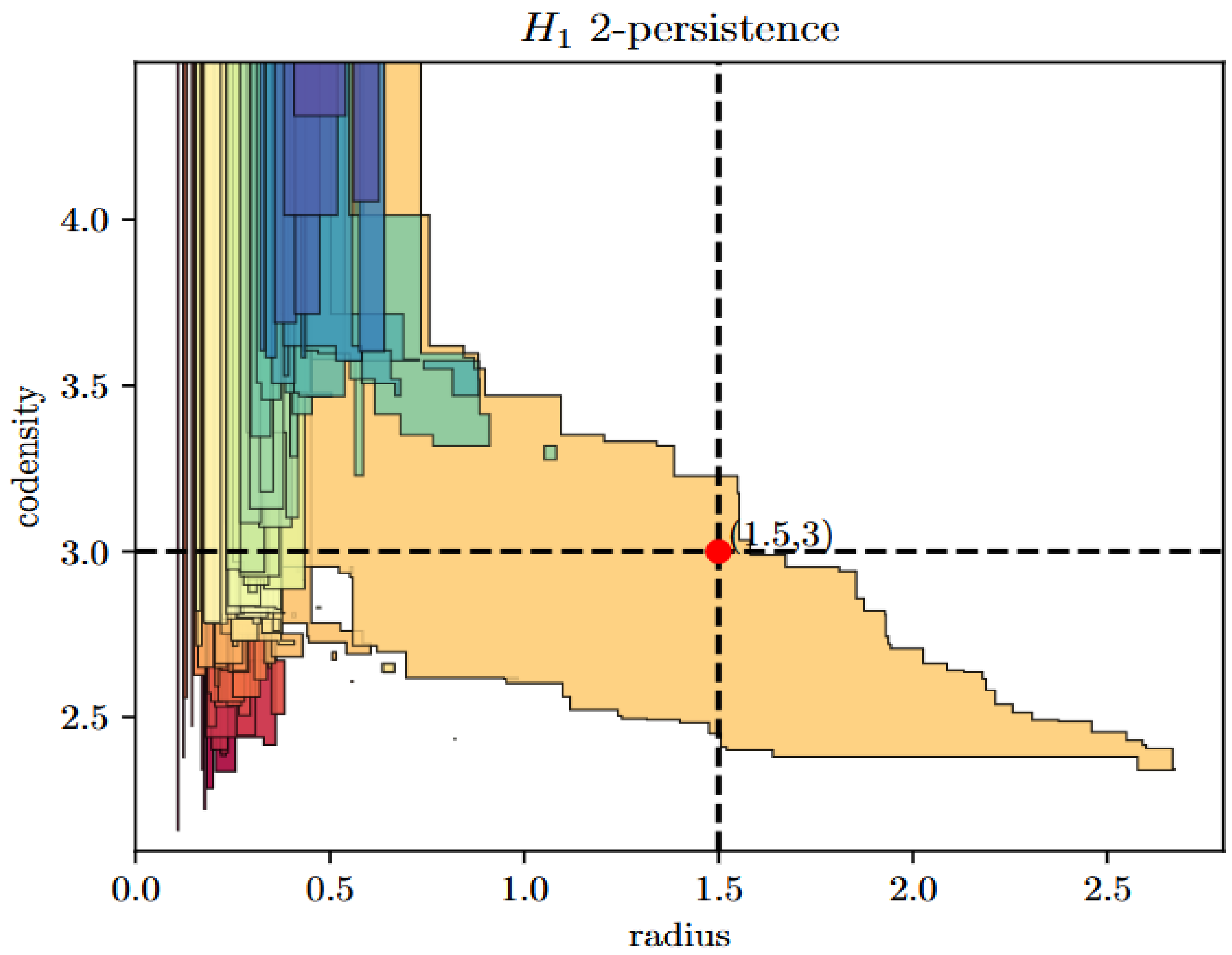




Algorithms

In practice: Candidate decompositions can be computed with `multipers`.





Other candidate decompositions are possible → Havard's talk.

[Loiseaux, C., Blumberg, **A Framework for Fast and Stable Representations of Multiparameter Persistent Homology Decompositions.** NeurIPS, 2023]

Candidate Decompositions $\rightarrow$ Descriptors

Goal: Compute an **embedding** of a candidate decomposition into a Hilbert space $\mathcal{H}$, so as to be able to do statistics and machine learning.

Candidate Decompositions $\to$ Descriptors

Goal: Compute an **embedding** of a candidate decomposition into a Hilbert space $\mathcal{H}$, so as to be able to do statistics and machine learning.

Idea: Use a generalized, "templated" version of persistence images:

Candidate Decompositions $\to$ Descriptors

Goal: Compute an **embedding** of a candidate decomposition into a Hilbert space $\mathcal{H}$, so as to be able to do statistics and machine learning.

Idea: Use a generalized, "templated" version of persistence images:

Def: Let $M = \oplus_i M_i$ be a candidate decomposition. Then,

$$V(M) = \mathrm{op}(\{w(M_i) \cdot \phi(M_i)\}_i)$$

is called a *template candidate decomposition representation* (T-CDR).

Candidate Decompositions $\to$ Descriptors

Goal: Compute an **embedding** of a candidate decomposition into a Hilbert space $\mathcal{H}$, so as to be able to do statistics and machine learning.

Idea: Use a generalized, "templated" version of persistence images:

Def: Let $M = \oplus_i M_i$ be a candidate decomposition. Then,

$$V(M) = \mathrm{op}(\{w(M_i) \cdot \phi(M_i)\}_i)$$

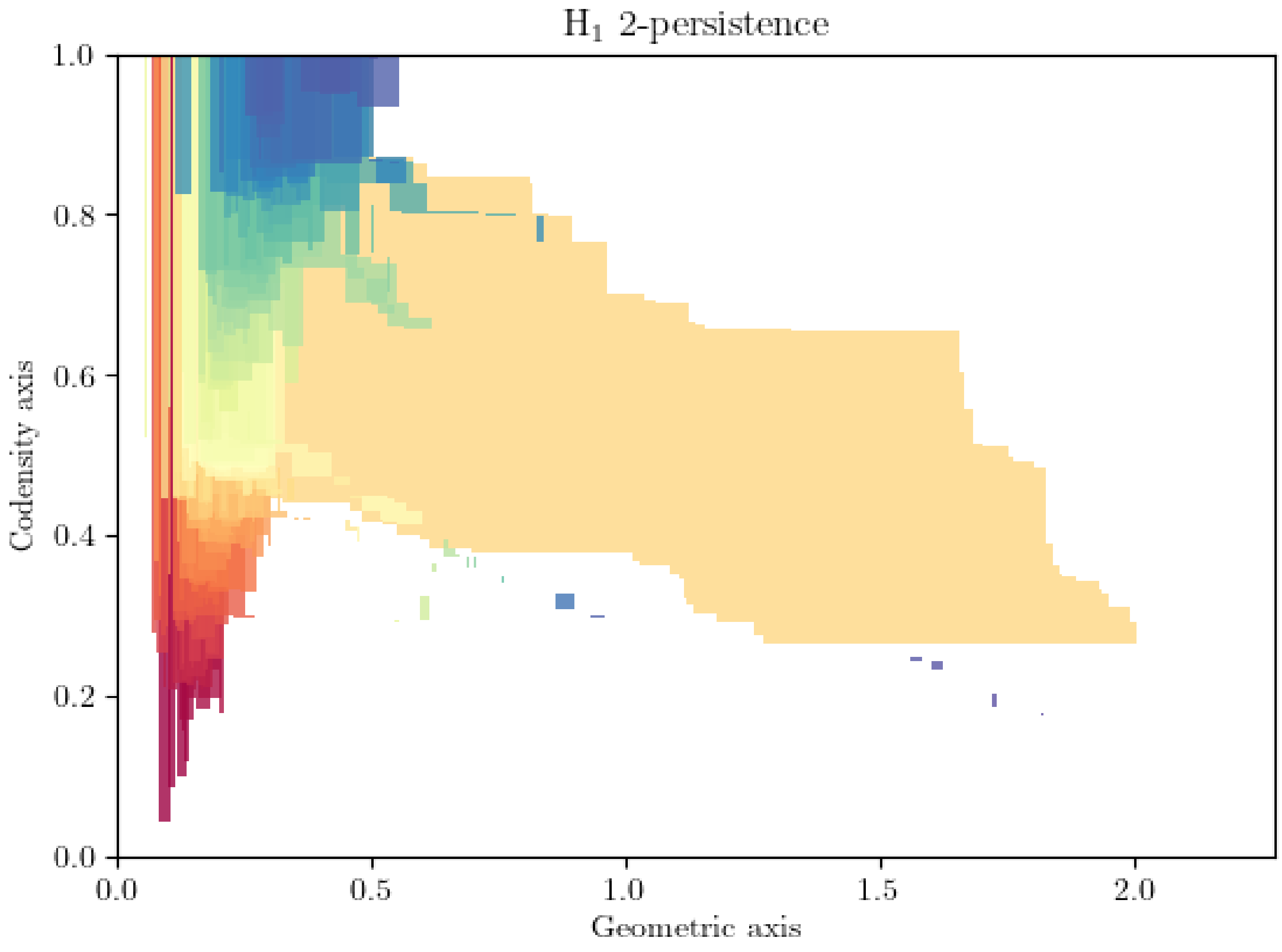
is called a *template candidate decomposition representation* (T-CDR).

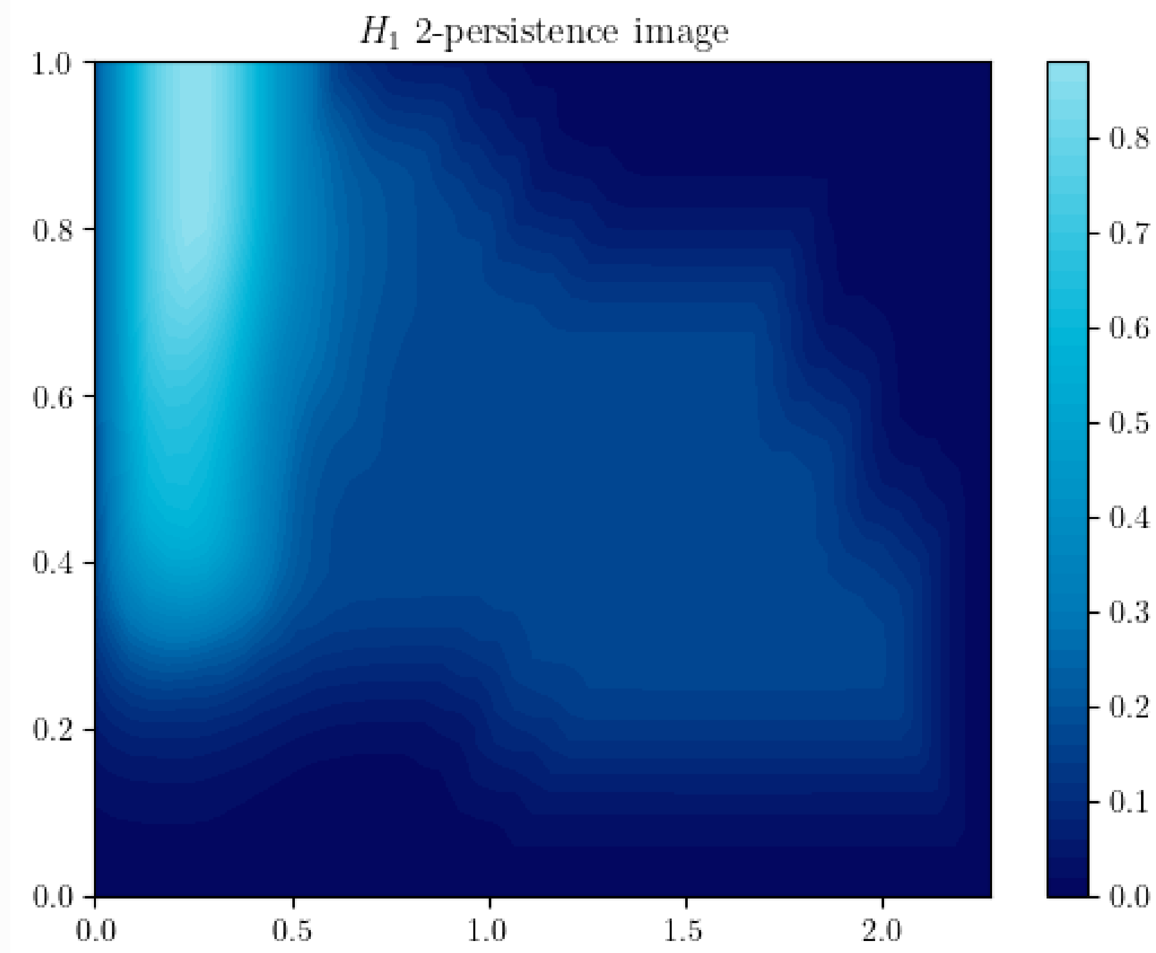
Appropriate choices of op, w and ϕ reproduce the multi-parameter landscape, fibered barcode kernel, etc.

Candidate Decompositions $\to$ Descriptors

Ex: The *stable candidate decomposition representation* (S-CDR) uses:

- $\mathrm{op} = \sum_i$,
- $w =$ longest diagonal in M_i,
- $\phi_\delta : x \mapsto$ volume of $M_i \cap B(x, \delta)$





Stability

Th: Let M, M' be two candidate decompositions. Then, S-CDR satisfy:

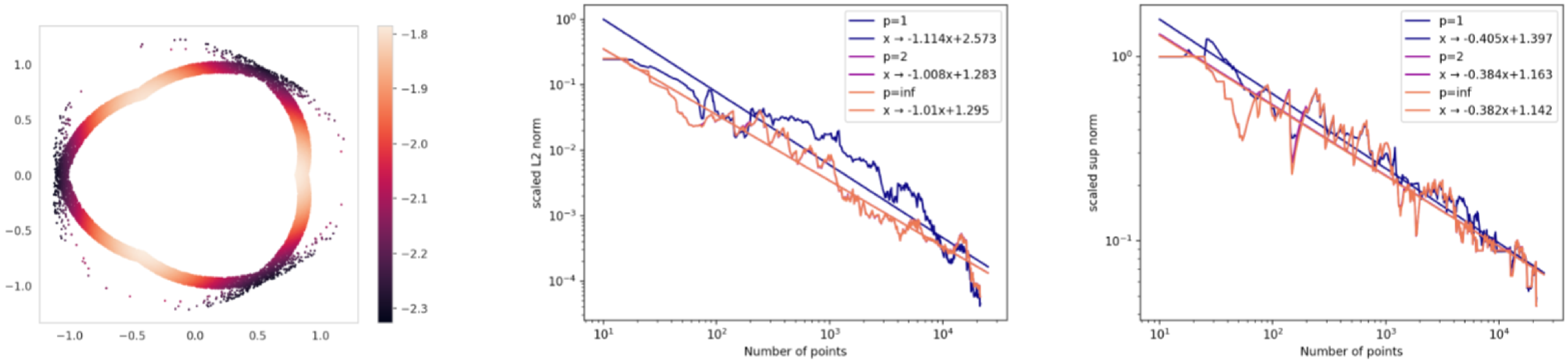
$$\|V(M) - V(M')\|_\infty \leq K d_b(M, M') \leq K d_I(M, M').$$

Stability

Th: Let M, M' be two candidate decompositions. Then, S-CDR satisfy:

$$\|V(M) - V(M')\|_\infty \leq K d_b(M, M') \leq K d_I(M, M').$$

Application: statistical convergence rate:



$$\mathbb{P}(\|V(M) - V(M')\|_\infty \geq \varepsilon) \leq \mathbb{P}(d_I(M, M') \geq \varepsilon / K) \leq \mathbb{P}(\|f - f'\|_\infty \geq \varepsilon / K).$$

Stability

Th: Let M, M' be two candidate decompositions. Then, S-CDR satisfy:

$$\|V(M) - V(M')\|_\infty \leq K d_b(M, M') \leq K d_I(M, M').$$

Application: accuracies in classification tasks:

Dataset	B1	B2	B3	PSS-K	P-I	P-L	MPK	MPL	MPI	S-CDR (Rips + KDE)	S-CDR (Alpha + DTM)
DPOAG	62.6	62.6	**77.0**	76.9	69.8	70.5	67.6	70.5	71.9	71.9	71.9
DPOC	71.7	72.5	71.7	47.5	67.4	66.3	**74.6**	69.6	71.7	73.8	**74.6**
PPOAG	78.5	78.5	80.5	75.9	82.0	78.0	78.0	78.5	81.0	81.9	**84.9**
PPOC	80.8	79.0	78.4	78.4	72.2	72.5	78.7	78.7	81.8	79.4	**83.2**
PPTW	70.7	75.6	75.6	61.4	72.2	73.7	**79.5**	73.2	76.1	75.6	75.1
IPD	**95.5**	**95.5**	95.0	-	64.7	61.1	80.7	78.6	71.9	81.2	77.2
GP	91.3	91.3	90.7	90.6	84.7	80.0	88.7	94.0	90.7	**96.3**	92.7
GPAS	89.9	**96.5**	91.8	-	84.5	87.0	93.0	85.1	90.5	88.0	93.7
GPMVF	97.5	97.5	**99.7**	-	88.3	87.3	96.8	88.3	95.9	95.3	95.9
PC	**93.3**	92.2	87.8	-	83.4	76.7	85.6	84.4	86.7	93.1	90.0

	Ripley	P	MPL	S-CDR
Immuno	67.2(2.3)	60.7(4.2)	65.3(3.0)	**91.4(1.6)**

1. Background: Persistence Theory

2. MMA: Multi-parameter Module Approximation

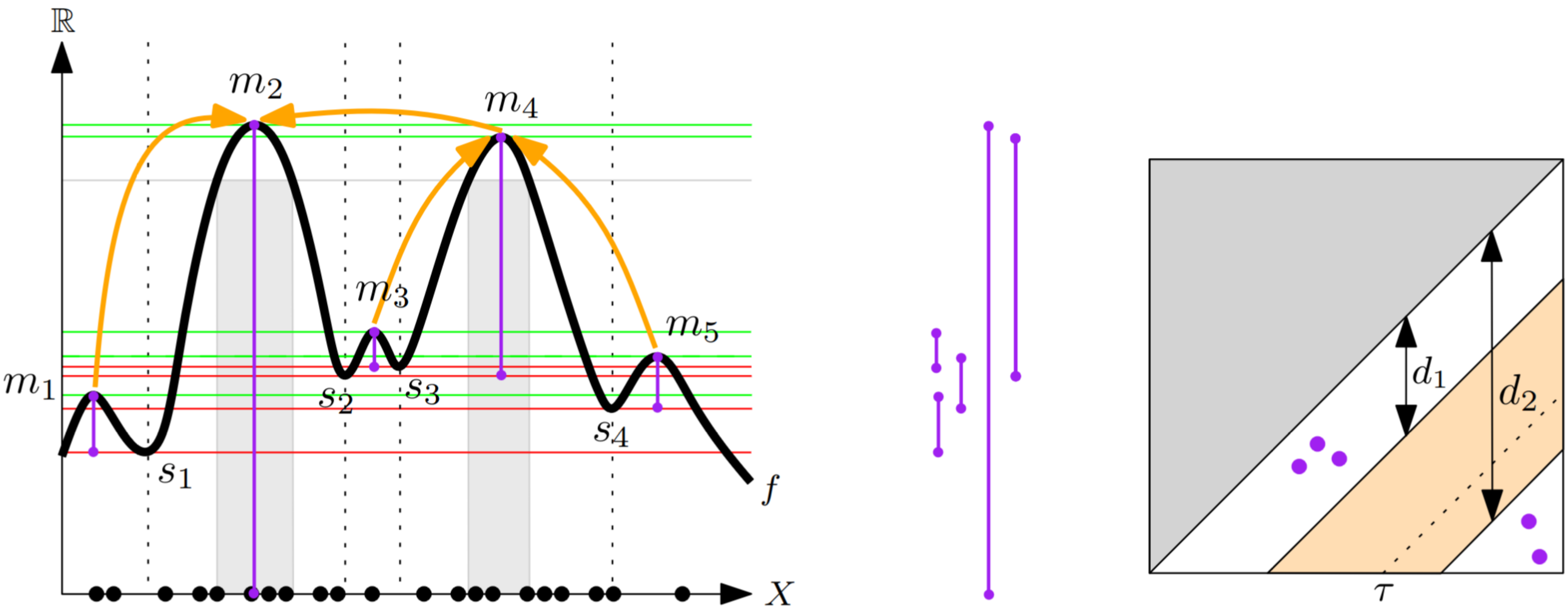
3. MMA Application: Machine Learning

4. MMA Application: Clustering

[Andrianirina, C., **ToMAToMP: Robust and Multi-parameter Topological Clustering.** Preprint, 2026]

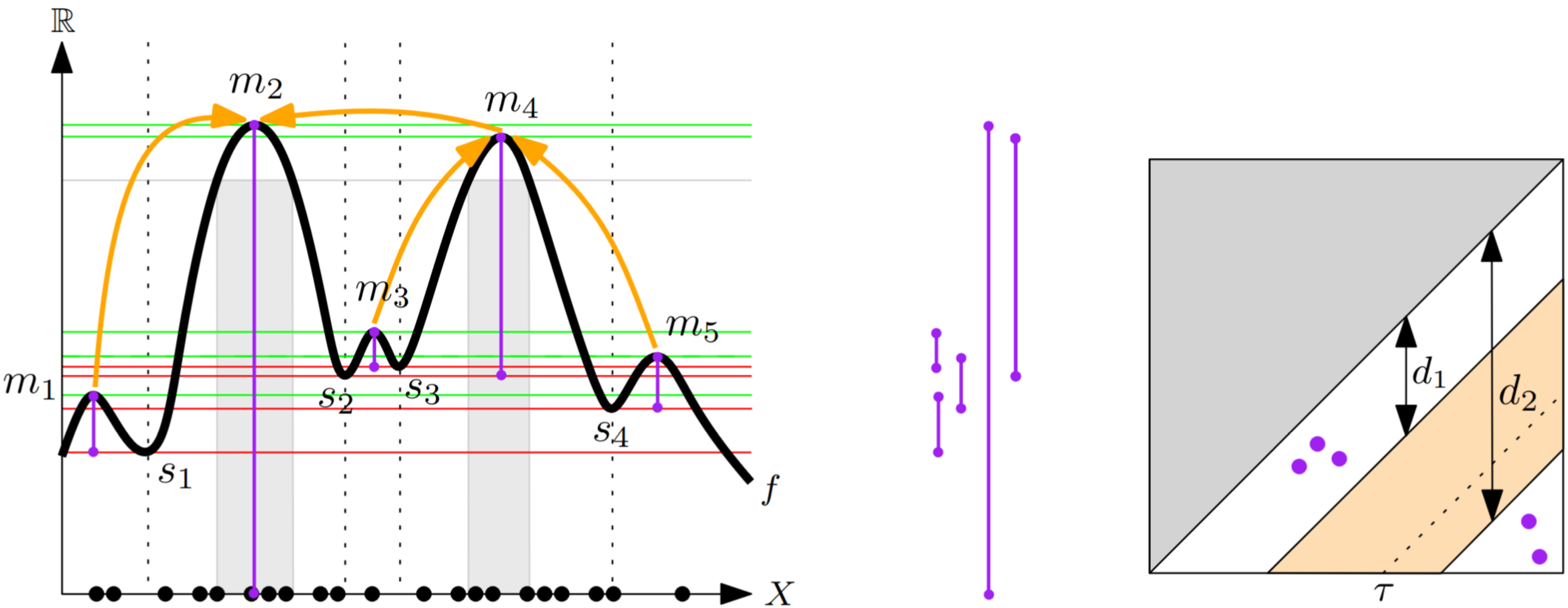
Persistence-based Clustering: ToMATo

ToMATo (Topological Mode Analysis Tool) uses 0-th persistence of a function f to cluster data, using an input graph G and persistence threshold τ.



Persistence-based Clustering: ToMATo

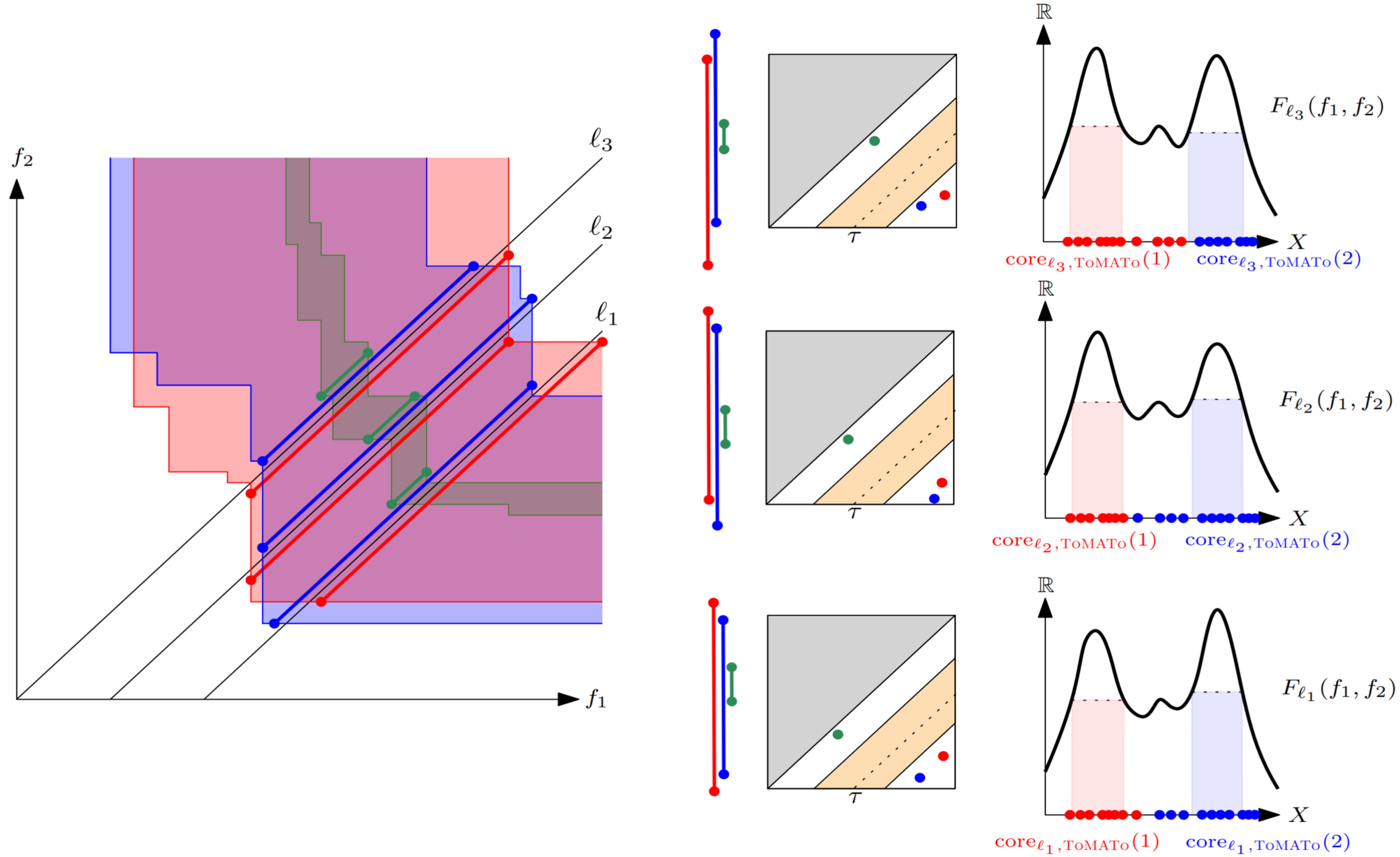
ToMATo (Topological Mode Analysis Tool) uses 0-th persistence of a function f to cluster data, using an input graph G and persistence threshold τ.



Goal: Use candidate decompositions to avoid dependence on G (using f + Rips) + improve robustness to outlier values (using f + outlier score).

Multi-parameter Clustering: ToMAToMP

Idea: ToMAToMP: apply ToMATo on lines and pick majoritary labels.



Multi-parameter Clustering: ToMAToMP

Idea: ToMAToMP: apply ToMATo on lines and pick majoritary labels.

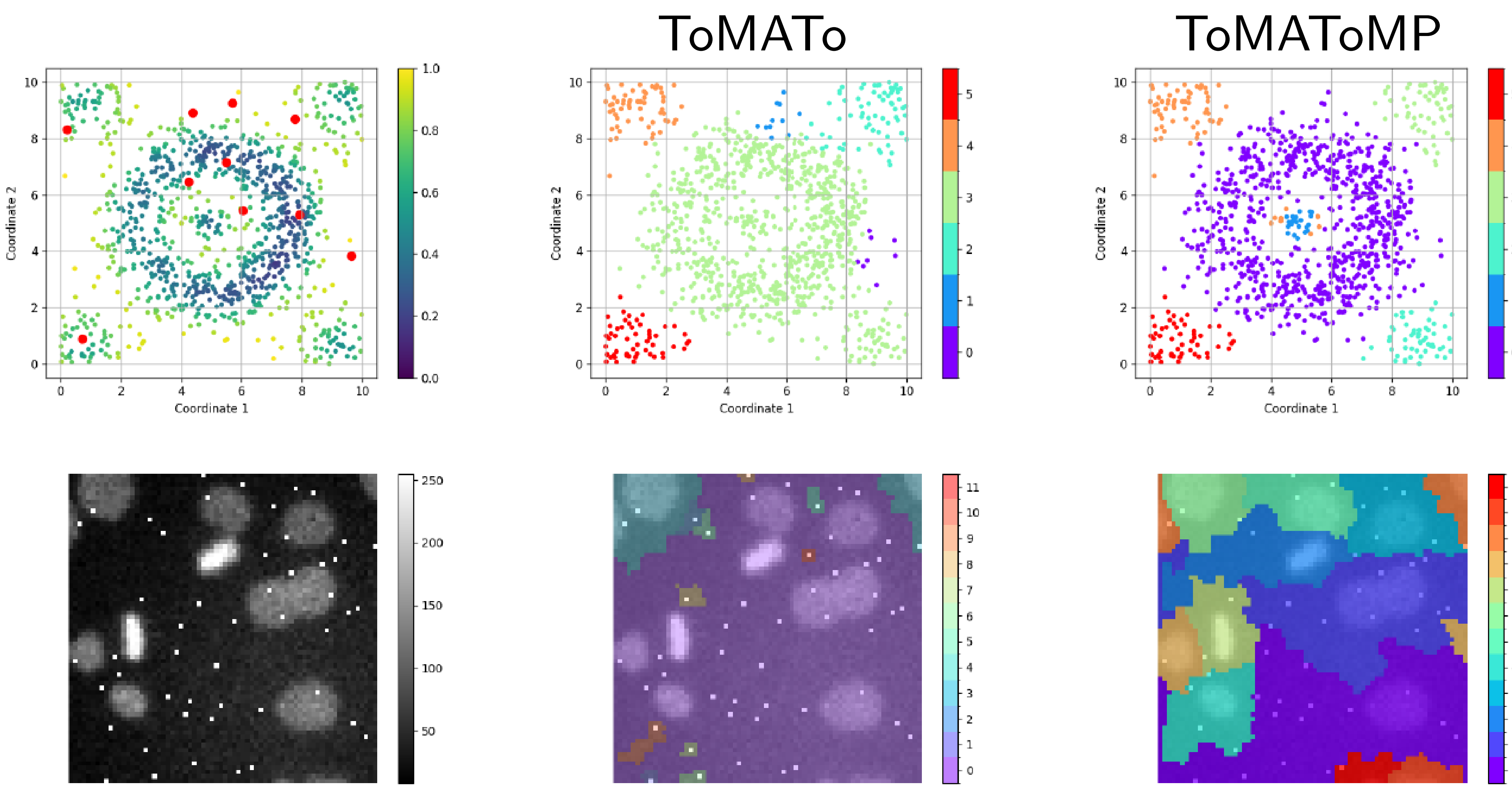
Th: If the fibered barcode is sufficiently separated, and τ is chosen accordingly, then ToMAToMP produces the right number of clusters.

Multi-parameter Clustering: ToMAToMP

Idea: ToMAToMP: apply ToMATo on lines and pick majoritary labels.

Th: If the fibered barcode is sufficiently separated, and τ is chosen accordingly, then ToMAToMP produces the right number of clusters.

Application: clustering with outliers values.

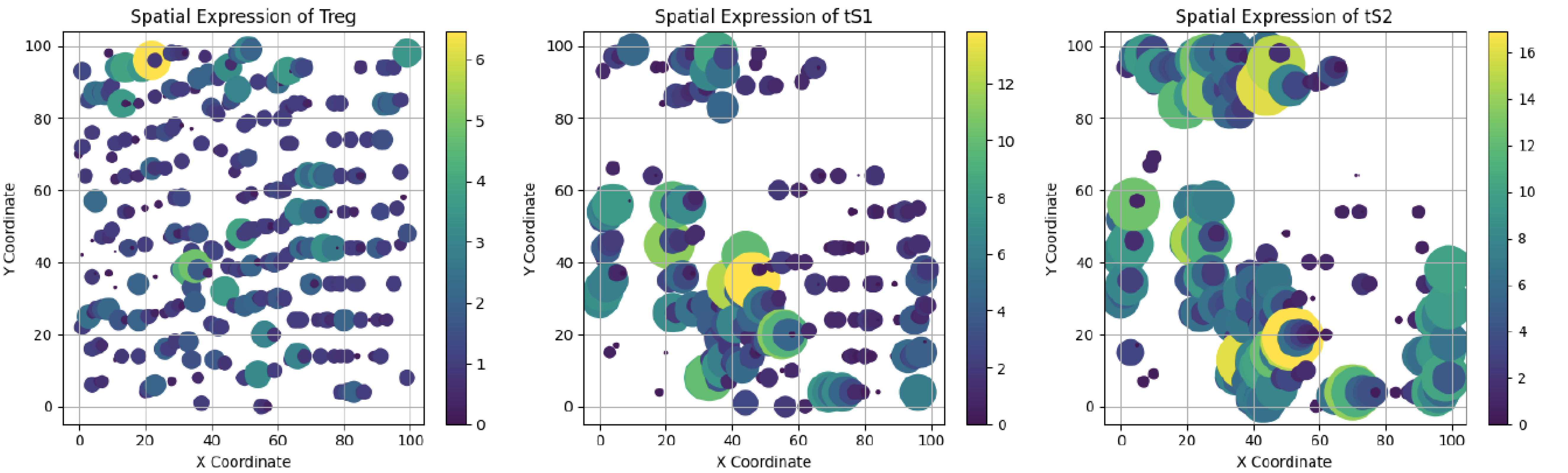


Multi-parameter Clustering: ToMAToMP

Idea: ToMAToMP: apply ToMATo on lines and pick majoritary labels.

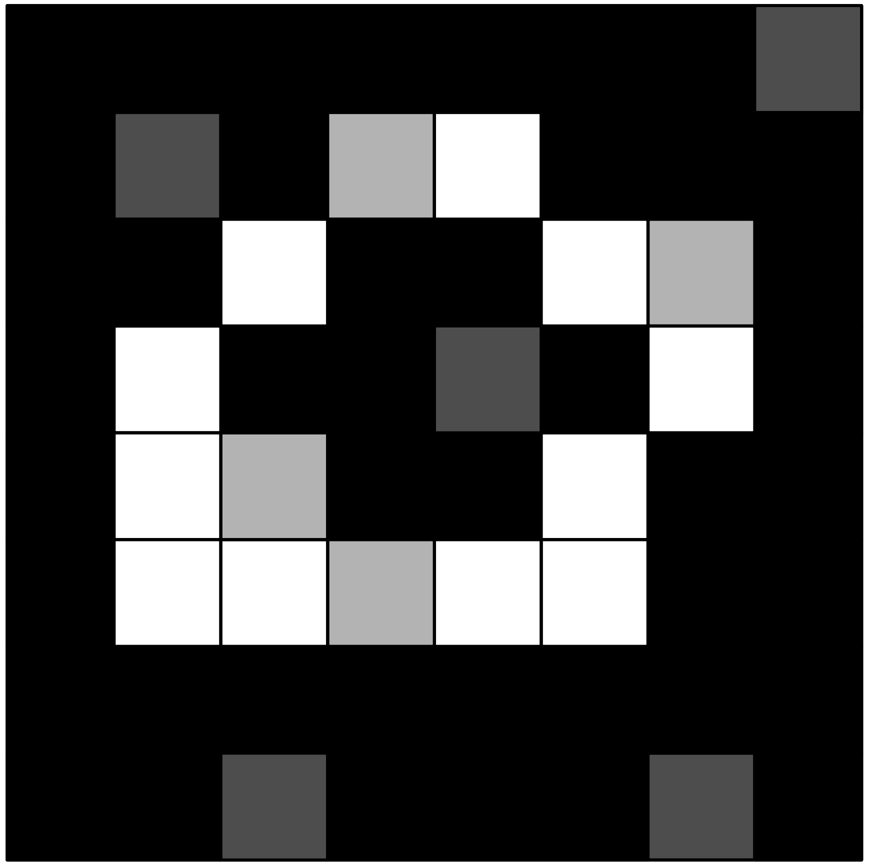
Th: If the fibered barcode is sufficiently separated, and τ is chosen accordingly, then ToMAToMP produces the right number of clusters.

Application: gene co-localization in spatial transcriptomics.



Thank you!

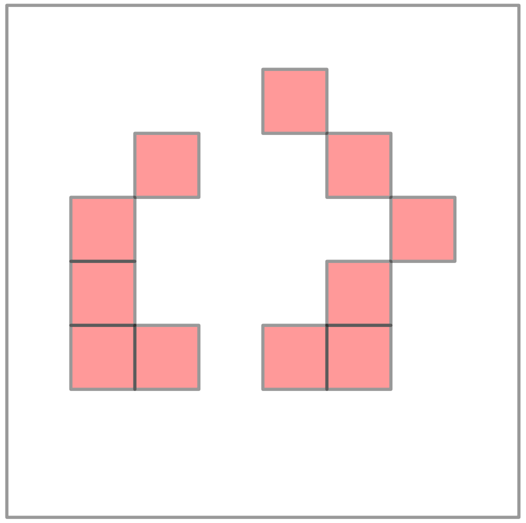
What is Single Parameter Persistence?



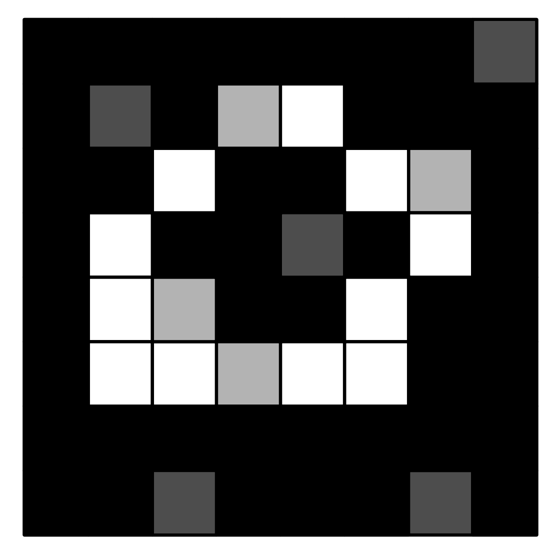
f is grey-level function

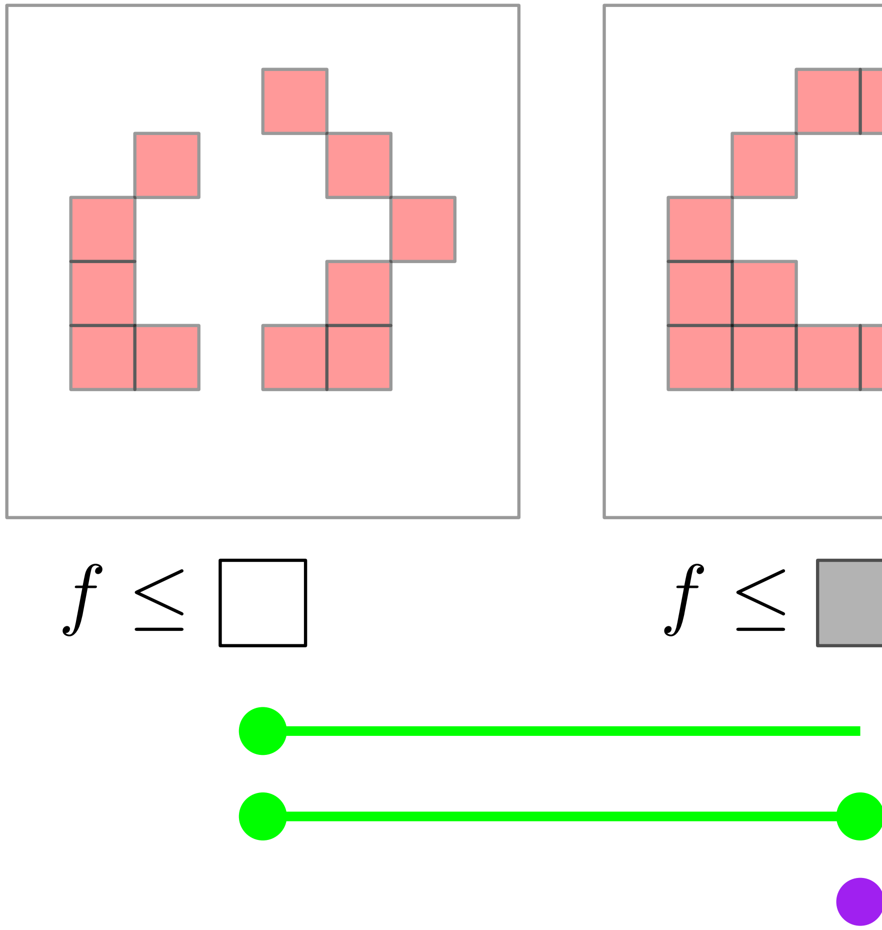
Encode topology of sublevel sets of f
(filtration induced by f)

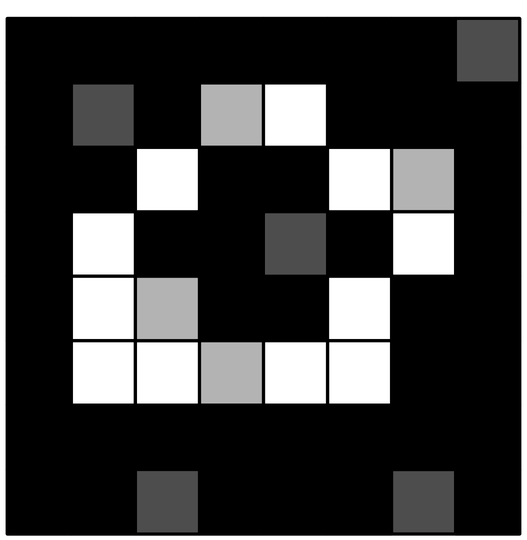
$$X_\alpha := f^{-1}((-\infty, \alpha]) \text{ for all } \alpha \in \mathbb{R} \cup \{+\infty\}$$

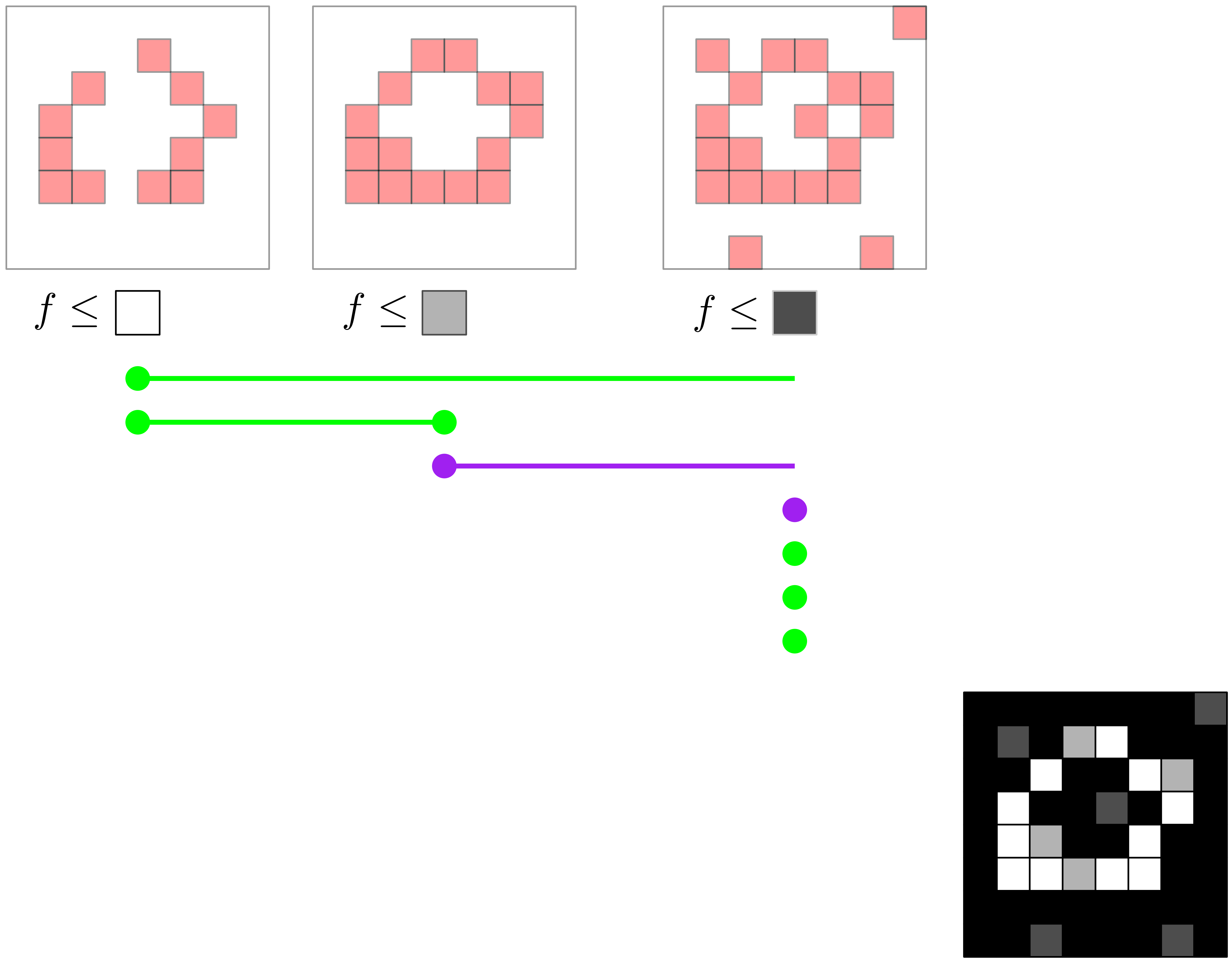


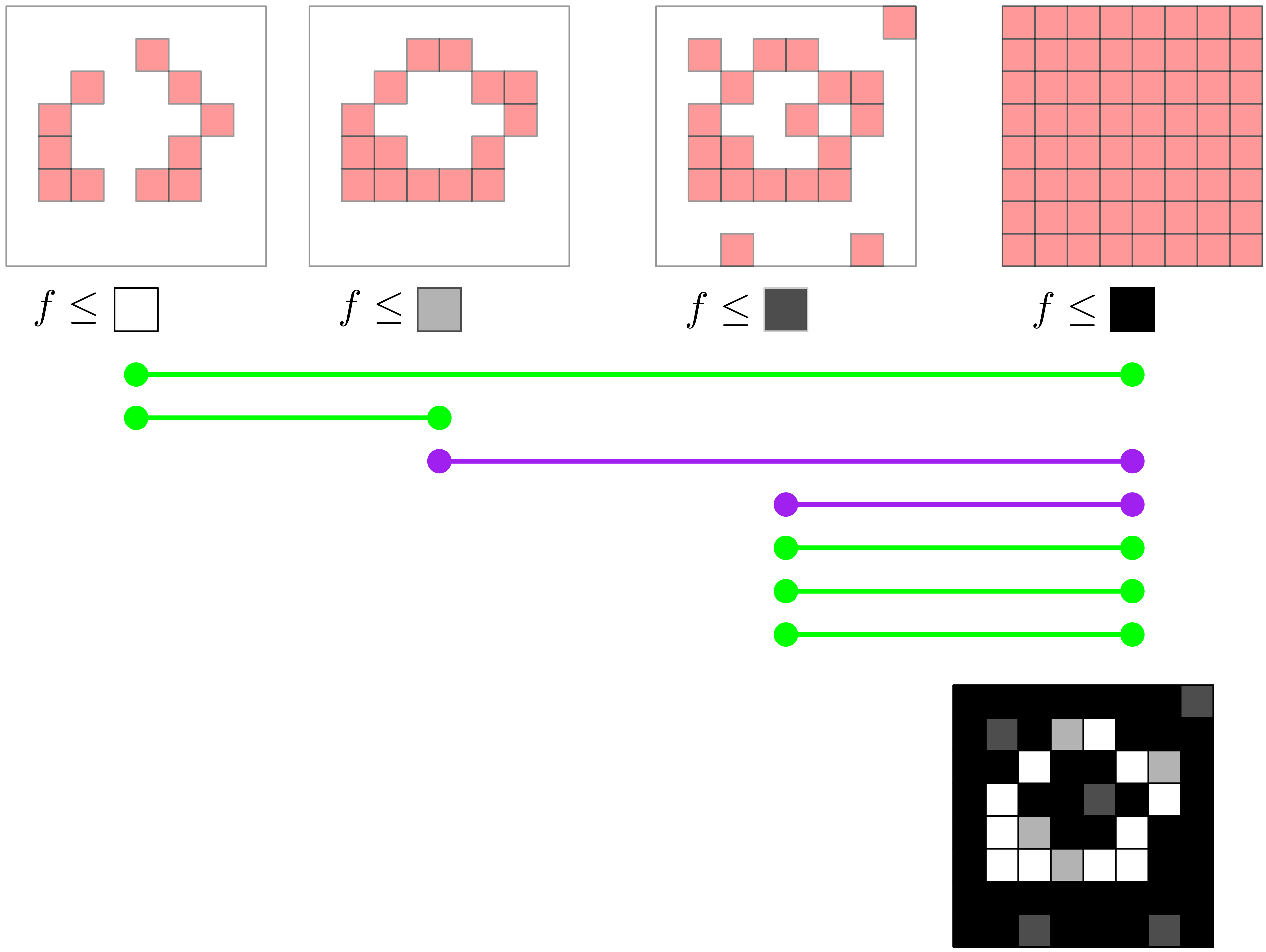
$f \leq \square$

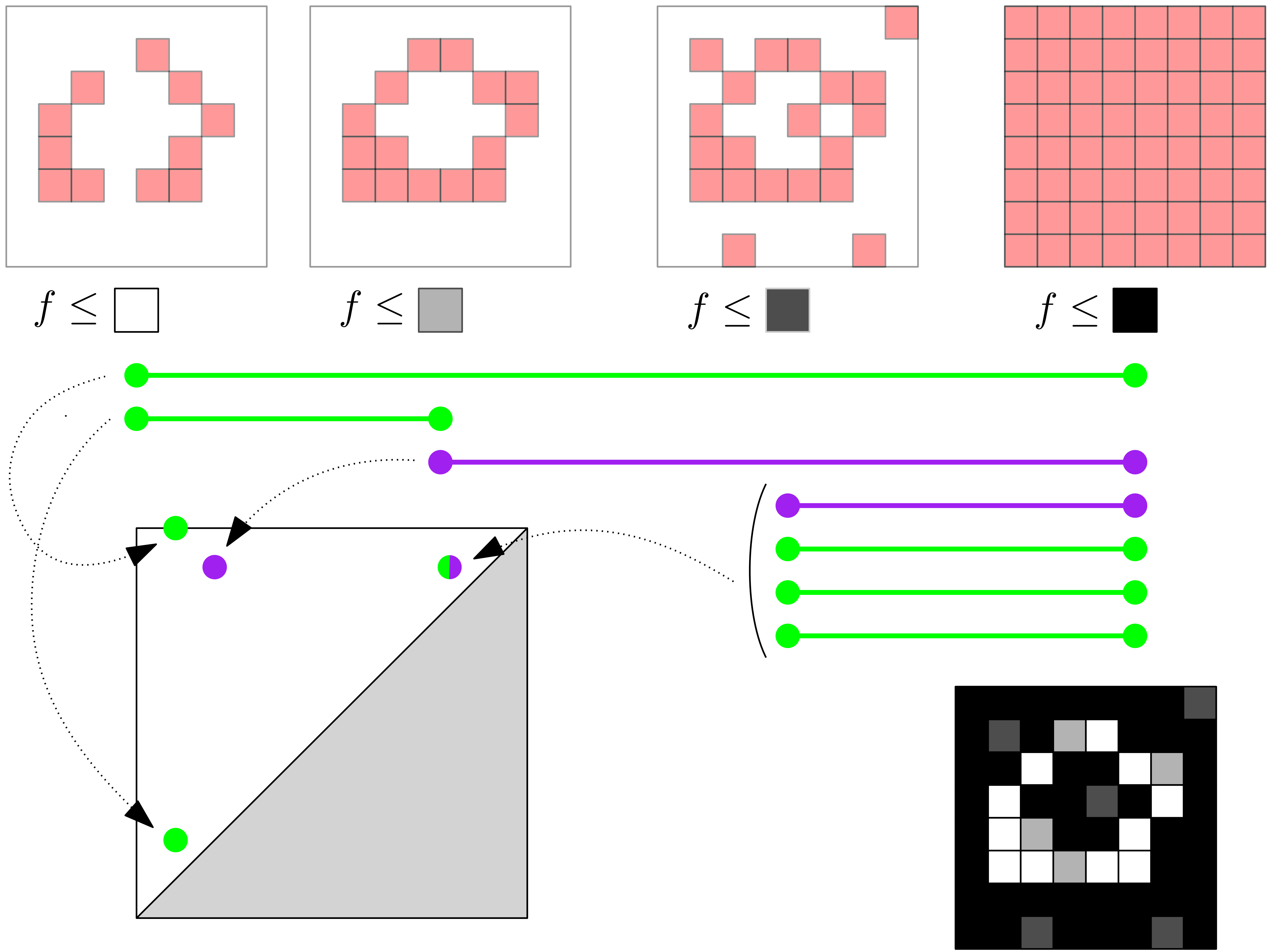


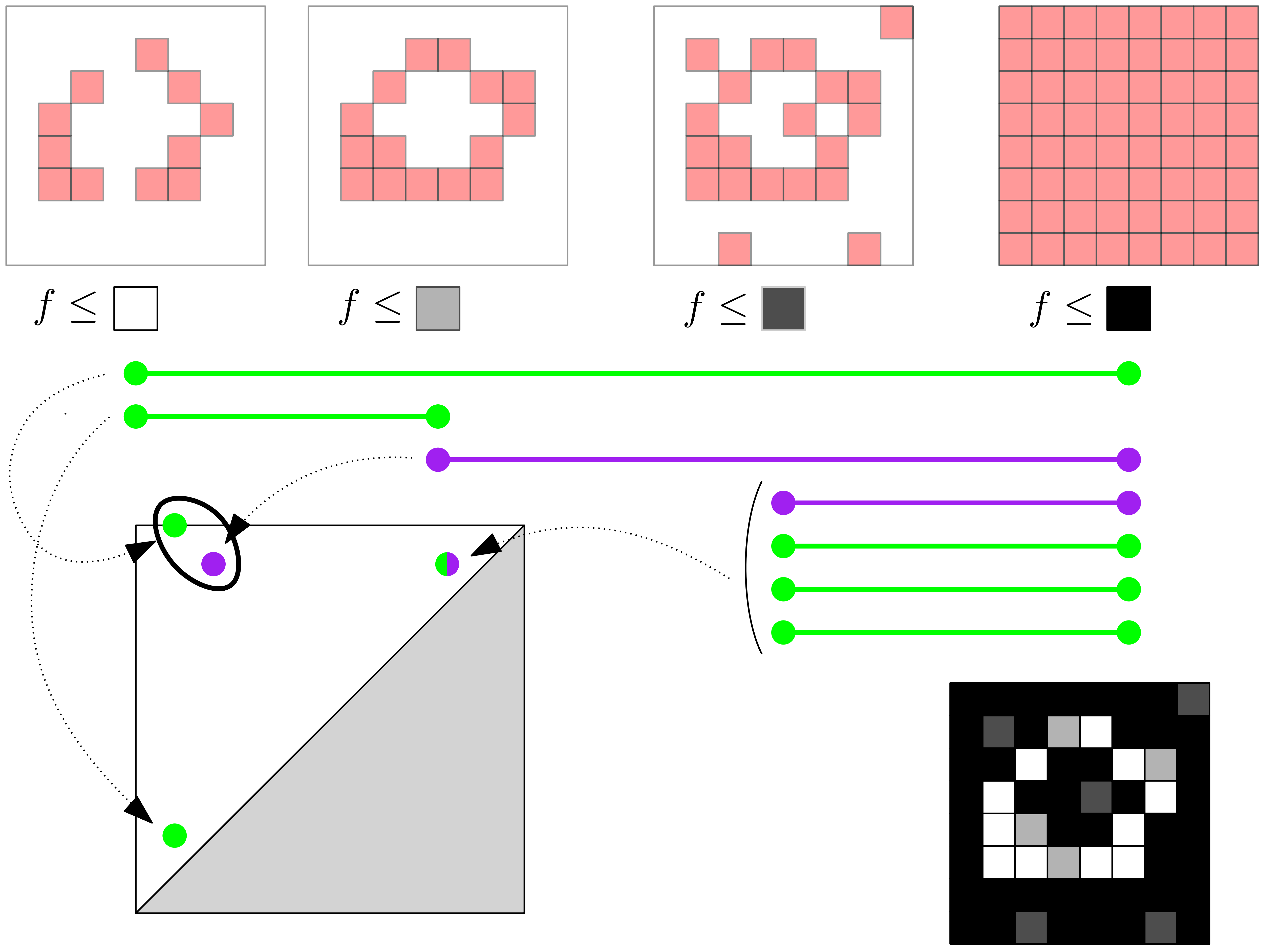

$f \leq$
$f \leq$




$f \leq$
$f \leq$
$f \leq$


$f \leq$
$f \leq$
$f \leq$
$f \leq$


$f \leq \square$
$f \leq \square$
$f \leq \square$
$f \leq \blacksquare$



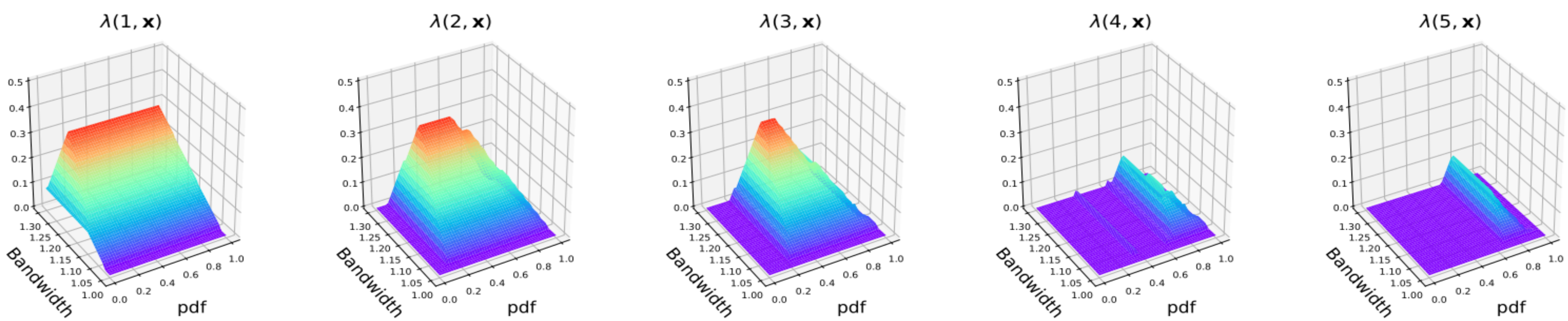
$f \leq$
$f \leq$
$f \leq$
$f \leq$

Statistics/Invariants Based on Persistent Betti Numbers

Def: Let M be a multiparameter persistence module. The multiparameter persistence landscape $\lambda(M)$ is defined as:

$$\lambda(M)(p,x) = \sup\{\epsilon > 0 \,:\, \beta^{x-h,x+h} \geq p, h \geq 0, \|h\|_\infty \leq \epsilon\}$$

Since $\mathrm{rk}(M)(x-h,x+h) \leq \mathrm{rk}(M)(x - \epsilon \cdot \begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}, x + \epsilon \cdot \begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix})$ the multiparameter persistence landscape simply amounts to computing the usual 1D persistence landscapes of all barcodes in the fibered barcode $\mathcal{L}(M)$.



[Vipond. **Multiparameter persistence landscapes.** 2020]

Statistics/Invariants Based on Persistent Betti Numbers

Def: Let M be a multiparameter persistence module. The multiparameter persistence landscape $\lambda(M)$ is defined as:

$$\lambda(M)(p,x) = \sup\{\epsilon > 0 \,:\, \beta^{x-h,x+h} \geq p, h \geq 0, \|h\|_\infty \leq \epsilon\}$$

Since $\mathrm{rk}(M)(x-h,x+h) \leq \mathrm{rk}(M)(x - \epsilon \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, x + \epsilon \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix})$
the multiparameter persistence landscape simply amounts to computing the usual 1D persistence landscapes of all barcodes in the fibered barcode $\mathcal{L}(M)$.

Def: Let M, M' be multiparameter persistence modules. Let K be a kernel ("scalar product") between 1D persistence diagrams and L be a family of lines. The multiparameter persistence kernel is defined as:

$$K(M,M') = \frac{1}{|L|}\sum_{\ell\in L} w(\ell)\cdot K(D(M|_\ell), D(M'|_\ell))$$

[Corbet et al. **A kernel for multi-parameter persistent homology.** 2019]

[Vipond. **Multiparameter persistence landscapes.** 2020]

What are Convergence Rates and Confidence Intervals?

A convergence rate is an inequality of the form:

$$\mathbb{E}[d(\hat{X}_n, X^*)] \leq f(n) \text{ with } f(n) \underset{n\to\infty}{\longrightarrow} 0,$$

where $\hat{X}_n$ is a (random) estimator of a target X^*.

What are Convergence Rates and Confidence Intervals?

A convergence rate is an inequality of the form:

$$\mathbb{E}[d(\hat{X}_n, X^*)] \leq f(n) \text{ with } f(n) \underset{n\to\infty}{\longrightarrow} 0,$$

where $\hat{X}_n$ is a (random) estimator of a target X^*.

Confidence intervals at level $\alpha \in [0,1]$ are quantities ε_α such that:

$$\mathbb{P}[d(\hat{X}_n, X^*) > \varepsilon_\alpha] \leq \alpha$$

What are Convergence Rates and Confidence Intervals?

A convergence rate is an inequality of the form:

$$\mathbb{E}[d(\hat{X}_n, X^*)] \leq f(n) \text{ with } f(n) \underset{n\to\infty}{\longrightarrow} 0,$$

where $\hat{X}_n$ is a (random) estimator of a target X^*.

Confidence intervals at level $\alpha \in [0,1]$ are quantities ε_α such that:

$$\mathbb{P}[d(\hat{X}_n, X^*) > \varepsilon_\alpha] \leq \alpha$$

Deriving a convergence rate gives confidence intervals "for free" with:

$$\mathbb{E}[d(\hat{X}_n, X^*)] = \int_{\varepsilon=0}^{+\infty} \mathbb{P}[d(\hat{X}_n, X^*) > \varepsilon]\,\mathrm{d}\varepsilon$$

Application to Support Estimation

Let $\hat{X}_n$ be a point cloud obtained from sampling a probability distribution μ_X with compact support $X \subseteq \mathbb{R}^d$, and let $\mu_{\hat{X}_n}$ be its corresponding distribution.

Convergence rates for probability distributions supported on $\mathbb{R}^d$:

$$\mathbb{E}[W_p(\mu_{\hat{X}_n}, \mu_X)] \leq C_p n^{-1/d},$$

where W_p is the p-Wasserstein distance between probability distributions.

[Lei. **Convergence and Concentration of Empirical Measures under Wasserstein Distance in Unbounded Functional Spaces.** 2020]

Application to Support Estimation

Let $\hat{X}_n$ be a point cloud obtained from sampling a probability distribution μ_X with compact support $X \subseteq \mathbb{R}^d$, and let $\mu_{\hat{X}_n}$ be its corresponding distribution.

Convergence rates for probability distributions supported on $\mathbb{R}^d$:

$$\mathbb{E}[W_p(\mu_{\hat{X}_n}, \mu_X)] \leq C_p n^{-1/d},$$

where W_p is the p-Wasserstein distance between probability distributions.

Let $\hat{M}_n, M$ be the persistence bimodules computed from the *measure bifiltrations* on X and $\hat{X}_n$: $X^{\mu}_{r,t} = \{x \in \mathbb{R}^d \mid \mu(B(x,r)) \leq t\}$.

Then: $d_I(\hat{M}_n, M) \leq d_{\mathrm{Pr}}(\mu_{\hat{X}_n}, \mu_X) \leq W_p(\mu_{\hat{X}_n}, \mu_X)^{\frac{p}{p+1}}$

[Blumberg, Lesnick. **Stability of 2-Parameter Persistent Homology.** 2022]

[Lei. **Convergence and Concentration of Empirical Measures under Wasserstein Distance in Unbounded Functional Spaces.** 2020]

Application to Support Estimation

Let $\hat{X}_n$ be a point cloud obtained from sampling a probability distribution μ_X with compact support $X \subseteq \mathbb{R}^d$, and let $\mu_{\hat{X}_n}$ be its corresponding distribution.

Convergence rates for probability distributions supported on $\mathbb{R}^d$:

$$\mathbb{E}[W_p(\mu_{\hat{X}_n}, \mu_X)] \leq C_p n^{-1/d},$$

where W_p is the p-Wasserstein distance between probability distributions.

Let $\hat{M}_n, M$ be the persistence bimodules computed from the *measure bifiltrations* on X and $\hat{X}_n$: $X^{\mu}_{r,t} = \{x \in \mathbb{R}^d \,|\, \mu(B(x,r)) \leq t\}$.

Then: $d_I(\hat{M}_n, M) \leq d_{\mathrm{Pr}}(\mu_{\hat{X}_n}, \mu_X) \leq W_p(\mu_{\hat{X}_n}, \mu_X)^{\frac{p}{p+1}}$

Hence, it only suffices to find Lipschitz persistence module invariants $\Phi(M)$, i.e., such that $\|\Phi(M) - \Phi(M')\| \leq C \cdot d_I(M, M')$, in order to retrieve convergence rates/confidence intervals (for $\Phi(M)$)!!

[Blumberg, Lesnick. **Stability of 2-Parameter Persistent Homology.** 2022]

[Lei. **Convergence and Concentration of Empirical Measures under Wasserstein Distance in Unbounded Functional Spaces.** 2020]

Application to Sublevel Sets

Stability theorem for multiparameter persistence modules obtained from sublevel sets of multivariate function:

Then: $d_I(M(f), M(g)) \leq \|f - g\|_\infty$.

[Lesnick. **The Theory of the Interleaving Distance on Multidimensional Persistence Modules.** 2015]

Application to Sublevel Sets

Stability theorem for multiparameter persistence modules obtained from sublevel sets of multivariate function:

Then: $d_I(M(f), M(g)) \leq \|f - g\|_\infty$.

In particular, the convergence rates of kernel density estimators f_h (with bandwidth $h > 0$) of some density f are also known:

$$\mathbb{E}\left[\|f - f_h\|_\infty\right] \leq C \cdot h + \sqrt{\frac{\log(1/h)}{n \cdot h^d}},$$

[Lesnick. **The Theory of the Interleaving Distance on Multidimensional Persistence Modules.** 2015]

[Kim et al. **Uniform convergence rate of the kernel density estimator adaptive to intrinsic volume dimension.** 2019]

Application to Sublevel Sets

Stability theorem for multiparameter persistence modules obtained from sublevel sets of multivariate function:

Then: $d_I(M(f), M(g)) \leq \|f - g\|_\infty$.

In particular, the convergence rates of kernel density estimators f_h (with bandwidth $h > 0$) of some density f are also known:

$$\mathbb{E}\left[\|f - f_h\|_\infty\right] \leq C \cdot h + \sqrt{\frac{\log(1/h)}{n \cdot h^d}},$$

As well as the rate of convergence of Hausdorff distance:

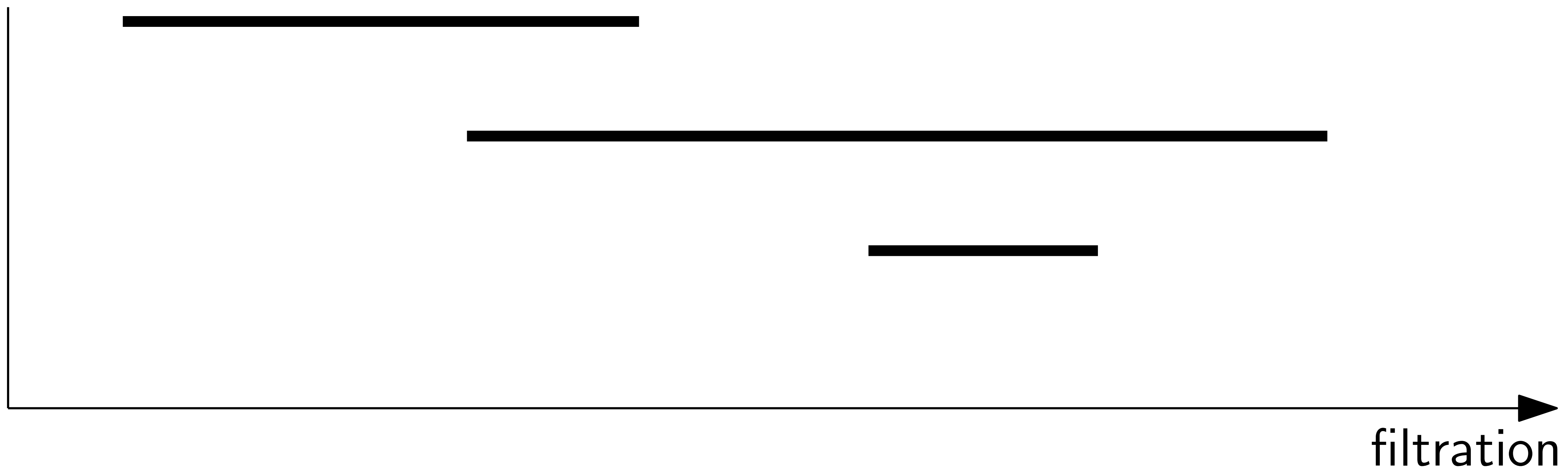
$$\mathbb{E}\left[d_H(X, \hat{X}_n)\right] \leq C \cdot \left(\frac{\log(n)}{n}\right)^{1/d}.$$

[Lesnick. **The Theory of the Interleaving Distance on Multidimensional Persistence Modules.** 2015]

[Kim et al. **Uniform convergence rate of the kernel density estimator adaptive to intrinsic volume dimension.** 2019]

The Single Parameter Case

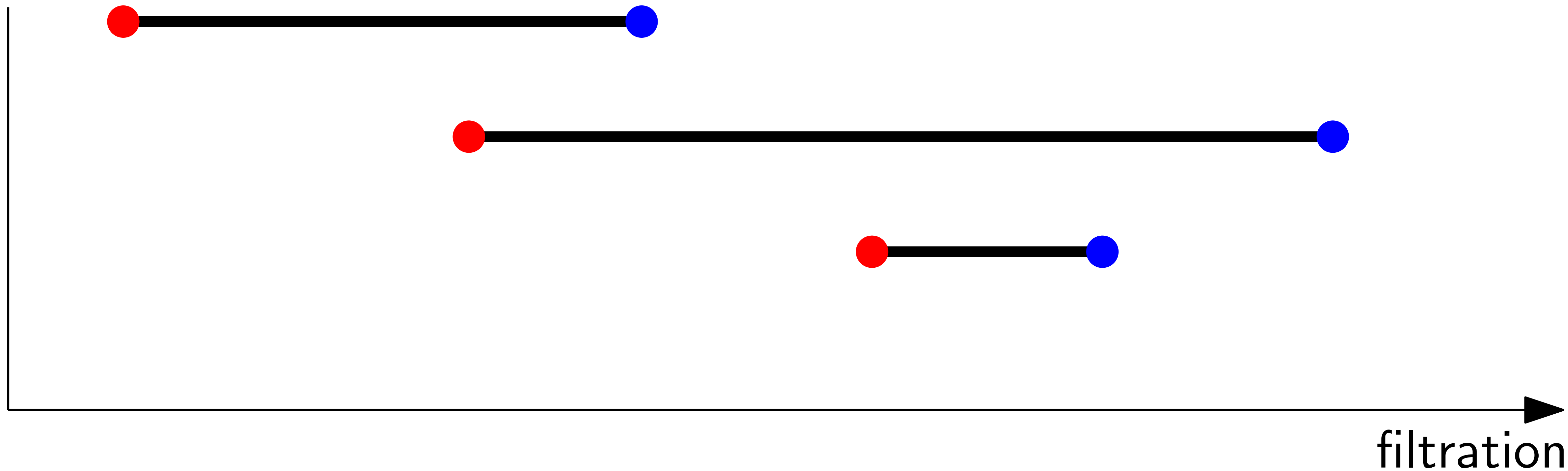
Let's look at the barcode:



The Single Parameter Case

Let's look at the barcode:

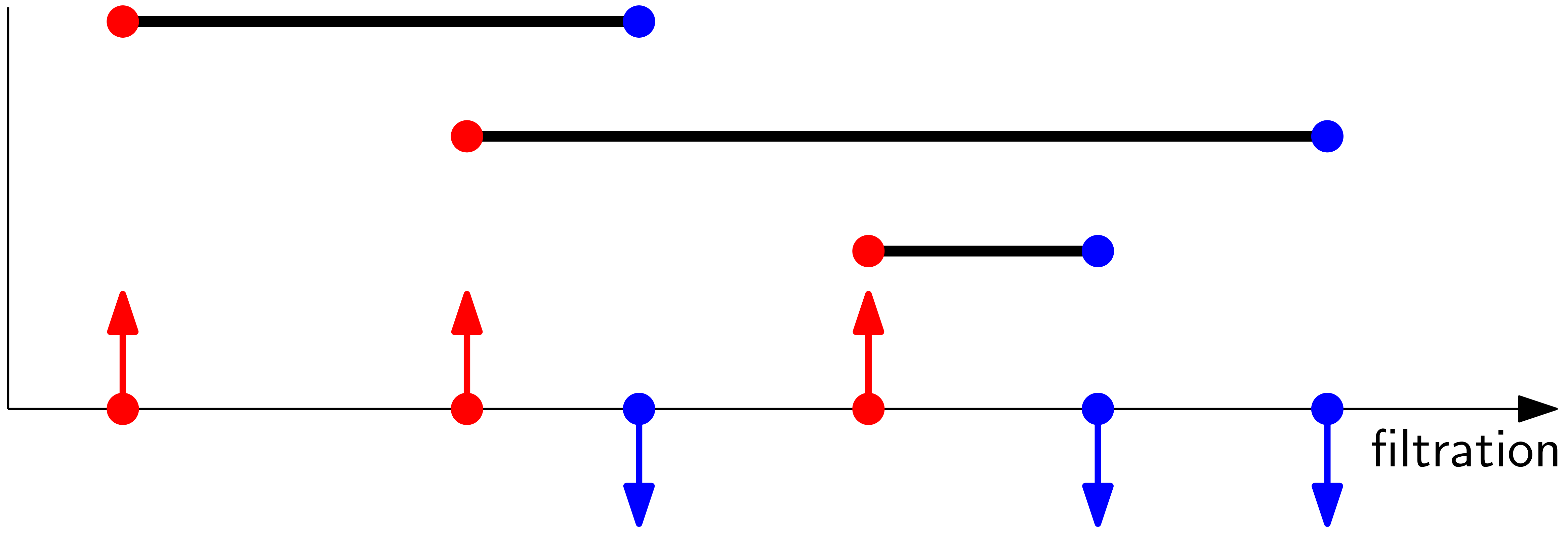
It has birth and death times,



The Single Parameter Case

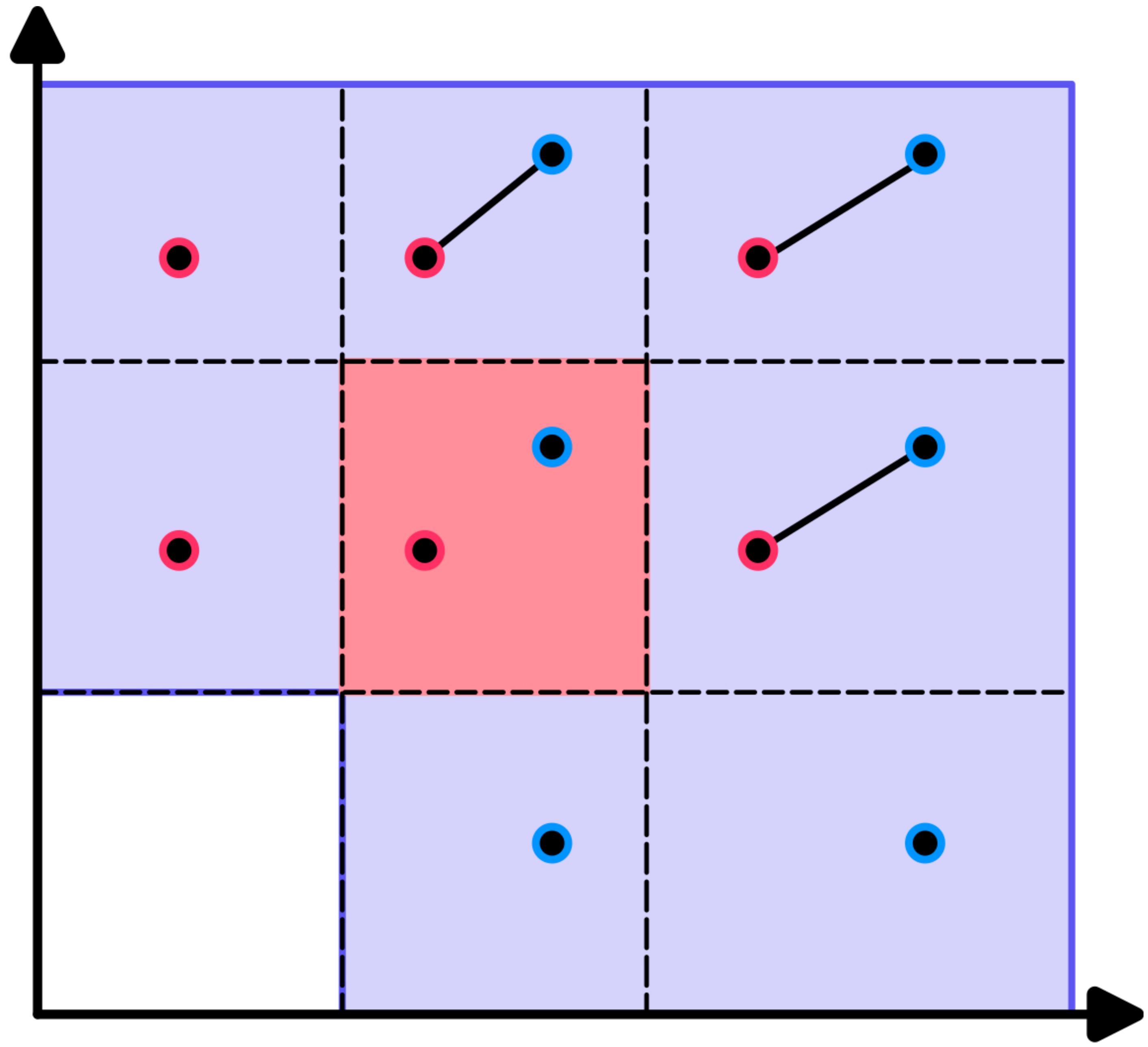
Let's look at the barcode:

It has birth and death times, that can be interpreted as an empirical measure, of total mass 0.



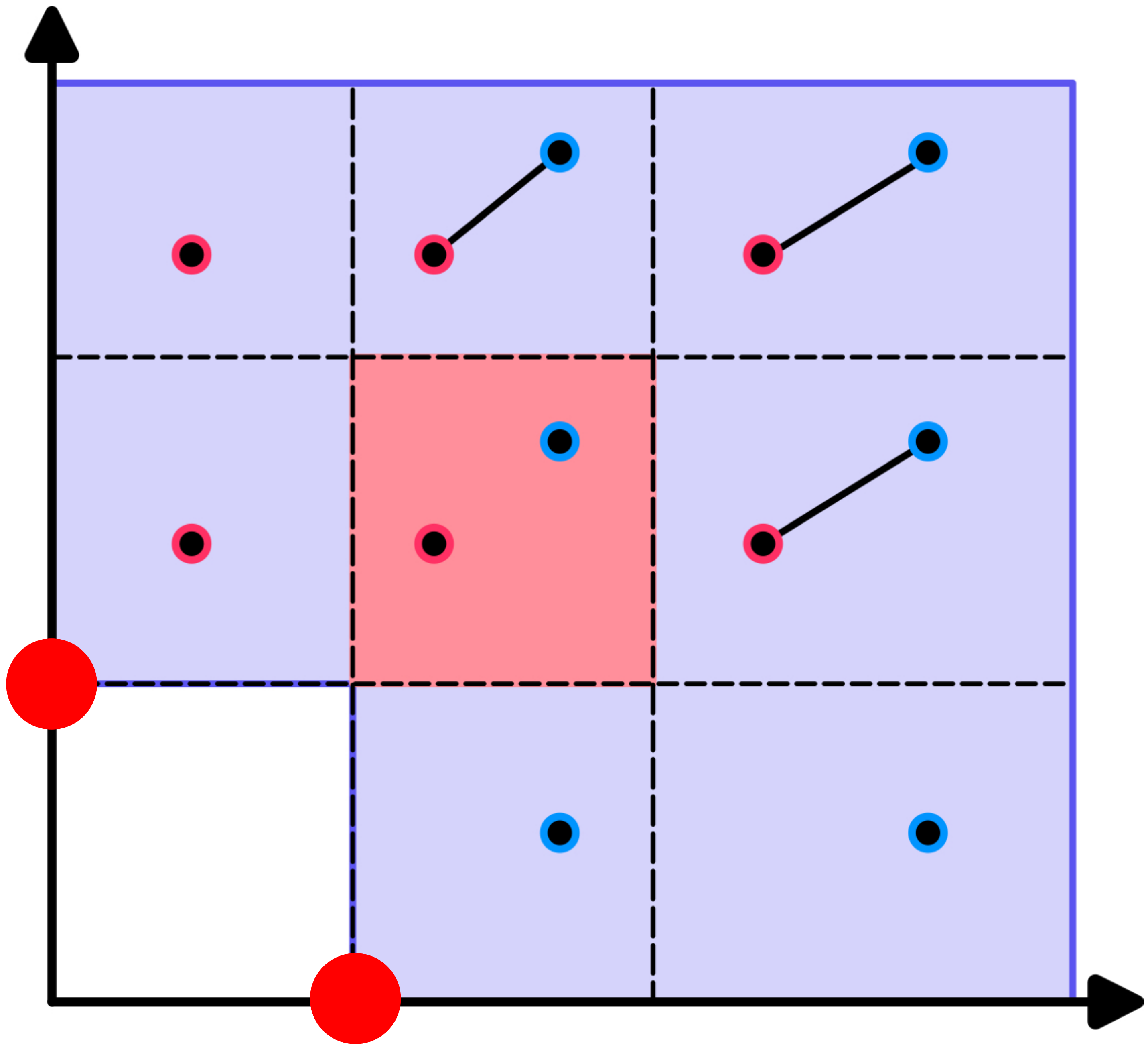
Extension to the Multiparameter Case

The same game can be played on multiparameter persistence modules!



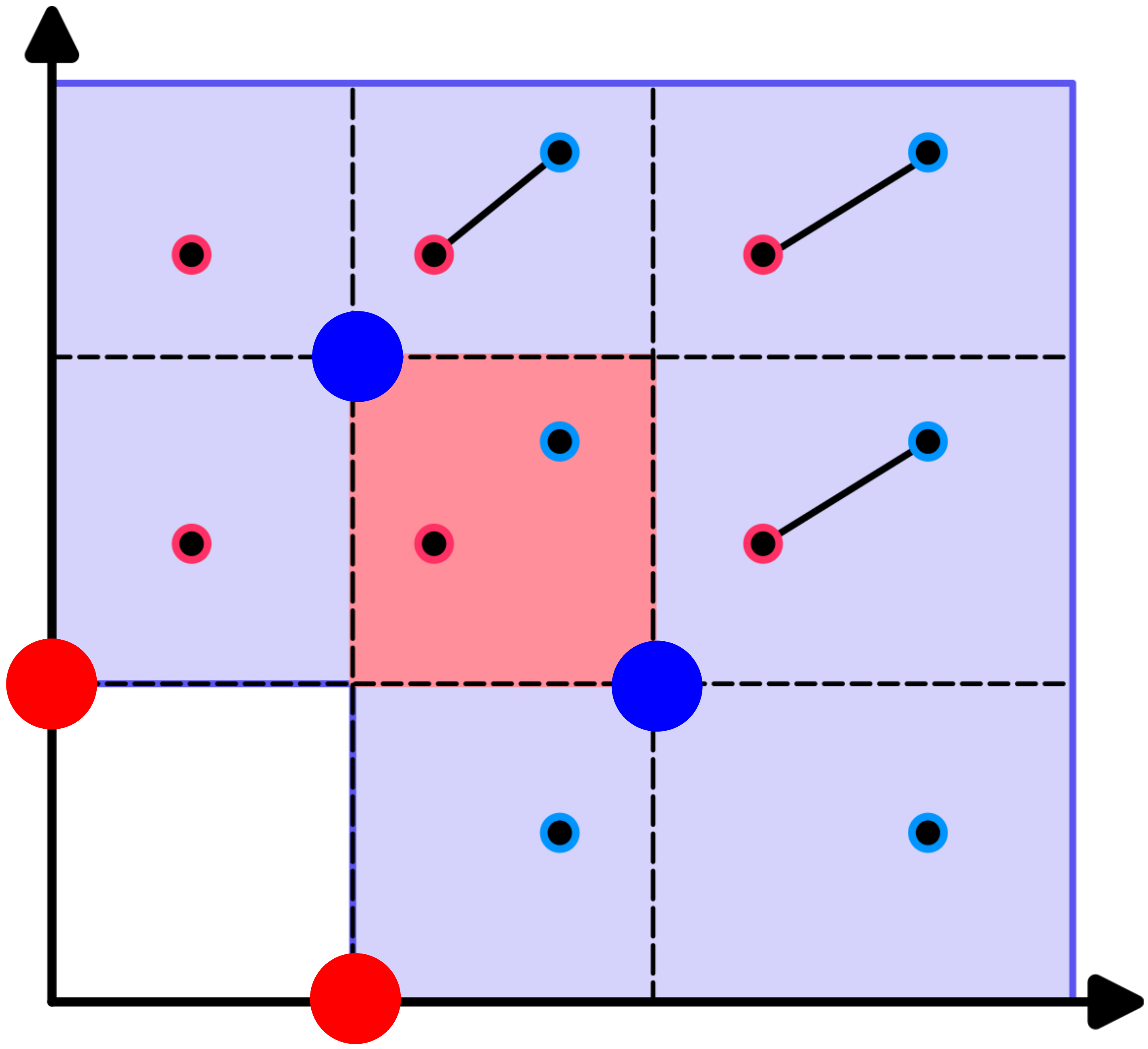
Extension to the Multiparameter Case

The same game can be played on multiparameter persistence modules!



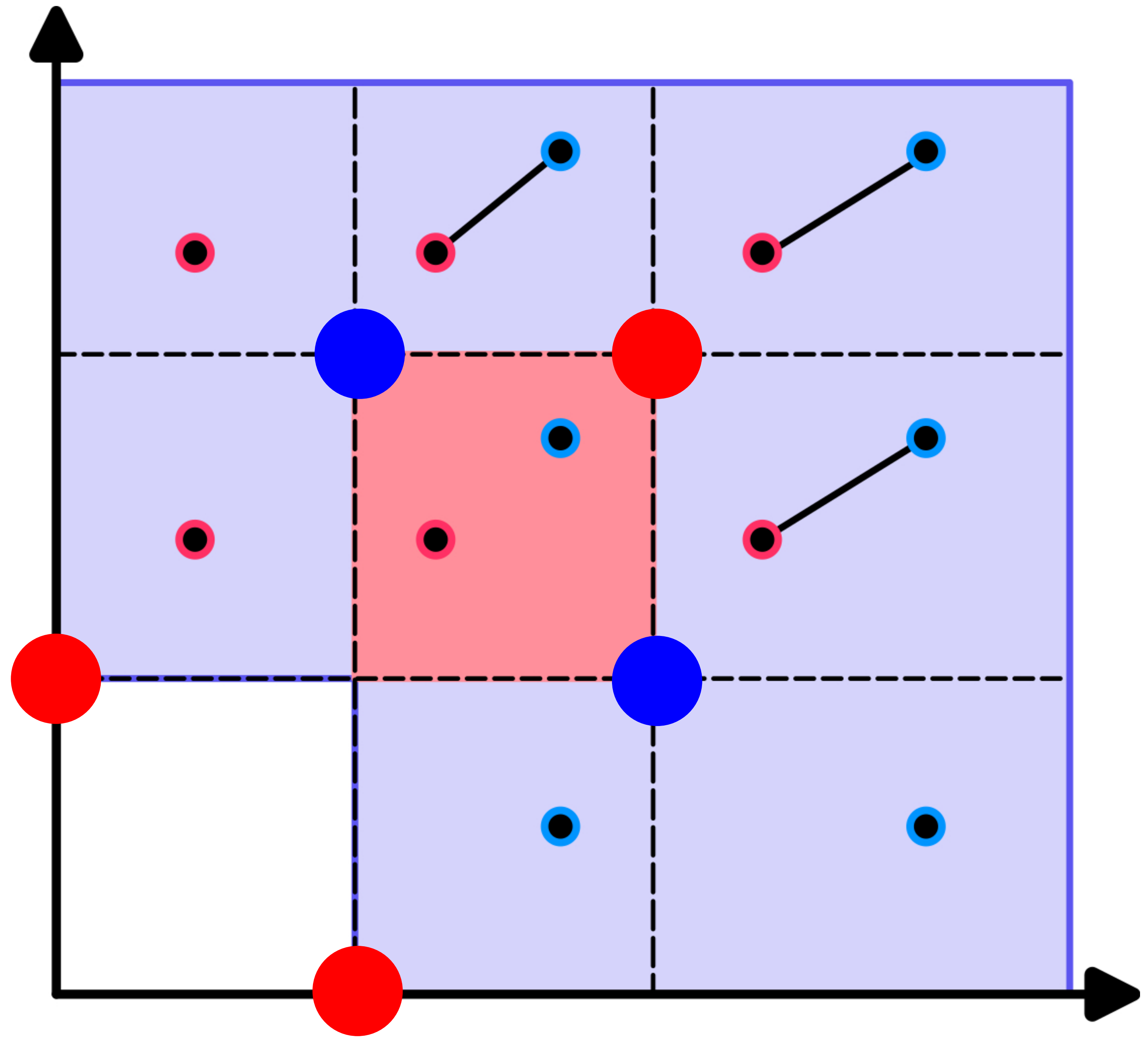
Extension to the Multiparameter Case

The same game can be played on multiparameter persistence modules!



Extension to the Multiparameter Case

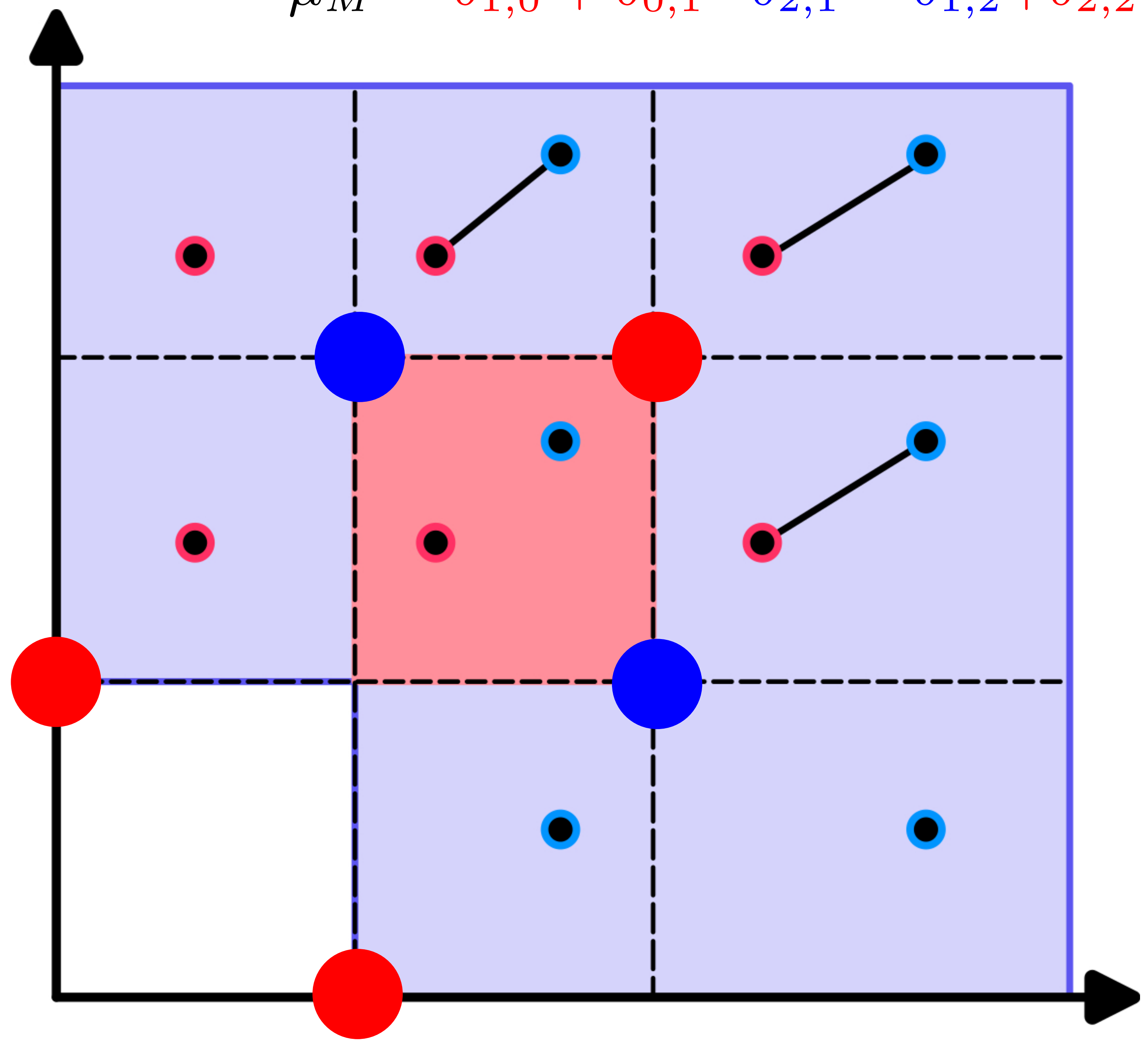
The same game can be played on multiparameter persistence modules!



Extension to the Multiparameter Case

The same game can be played on multiparameter persistence modules!

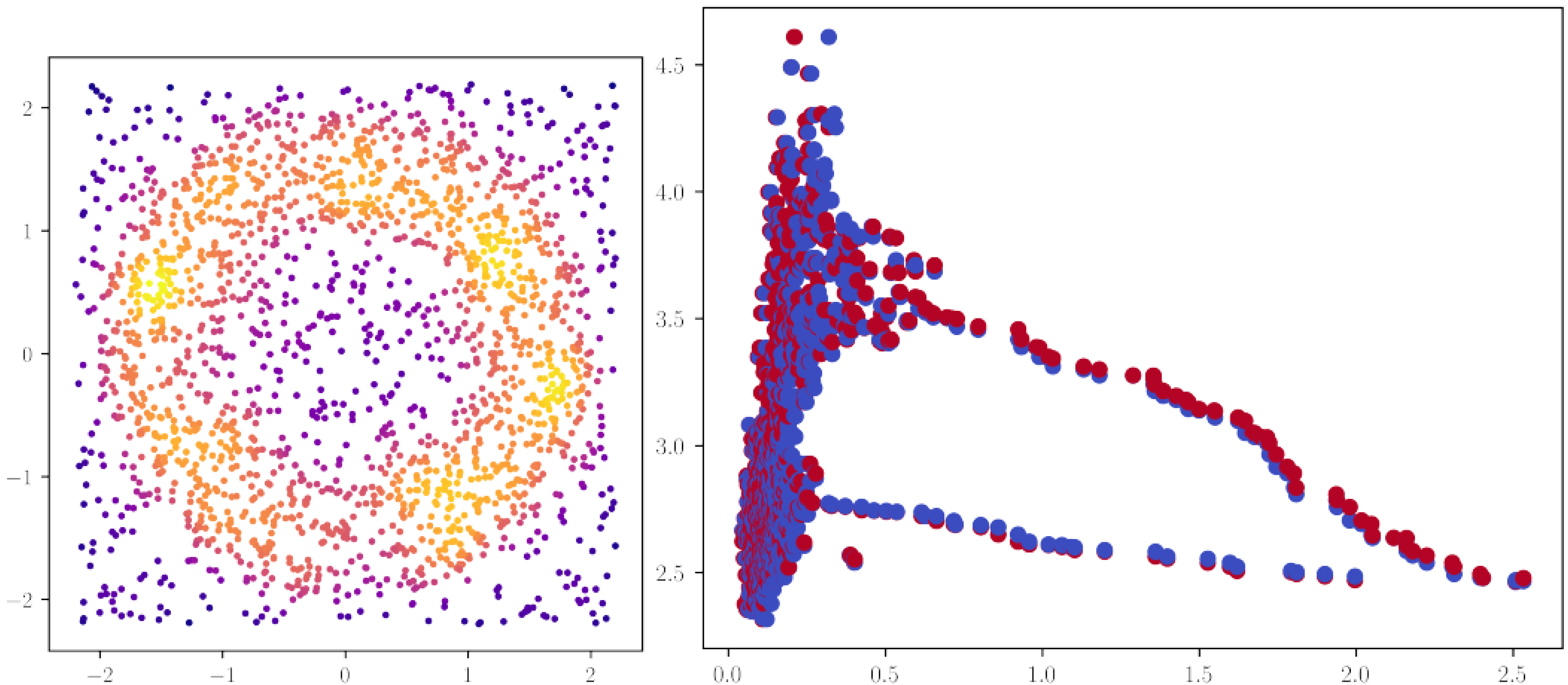
$$\mu_M = \delta_{1,0} + \delta_{0,1} - \delta_{2,1} - \delta_{1,2} + \delta_{2,2}$$



Extension to the Multiparameter Case

Prop: For any finitely presentable multiparameter persistence module M, there exists a unique discrete Radon measure μ_M such that:

$$\forall x \in \mathbb{R}^n, \dim(M_x) = \mu_M(\{y \in \mathbb{R}^n \mid y \leq x\}).$$



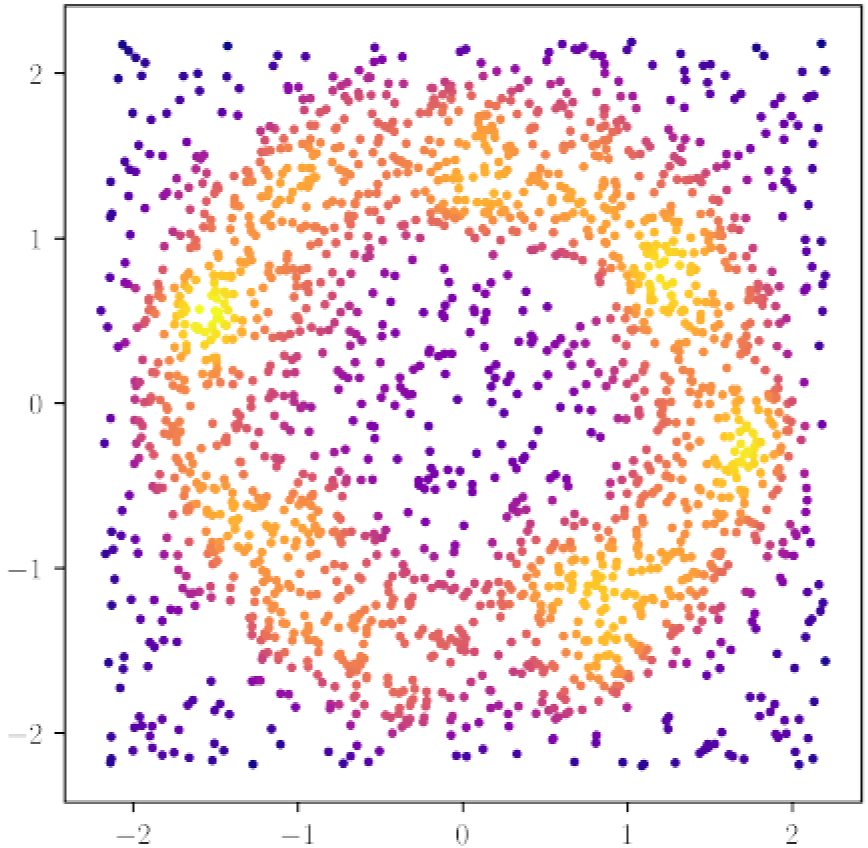
[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

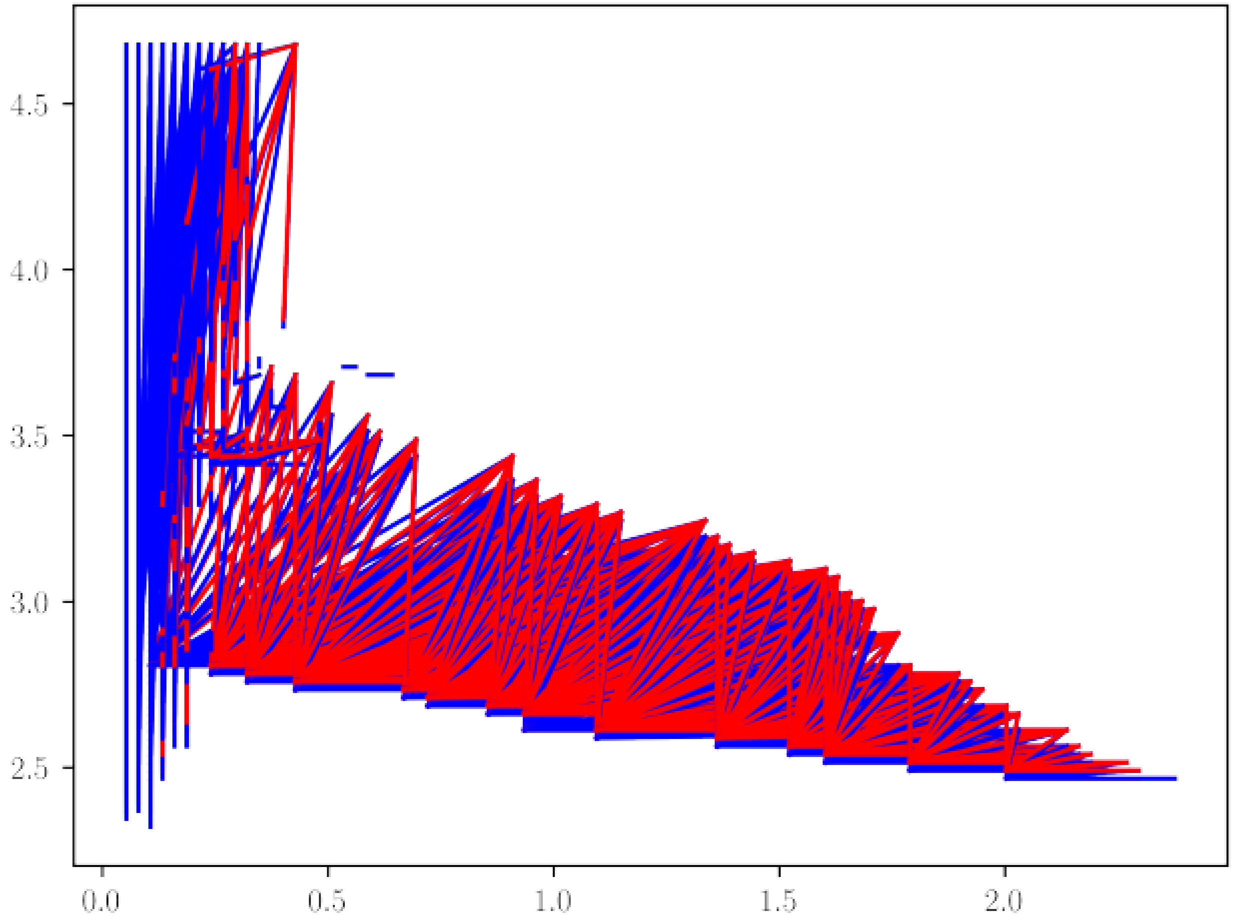
Extension to the Multiparameter Case

Prop: For any finitely presentable multiparameter persistence module M, there exists a unique discrete Radon measure μ_M such that:

$$\forall x \in \mathbb{R}^n, \dim(M_x) = \mu_M(\{y \in \mathbb{R}^n \,|\, y \leq x\}).$$

This decomposition also works for the Euler characteristic and rank invariant!

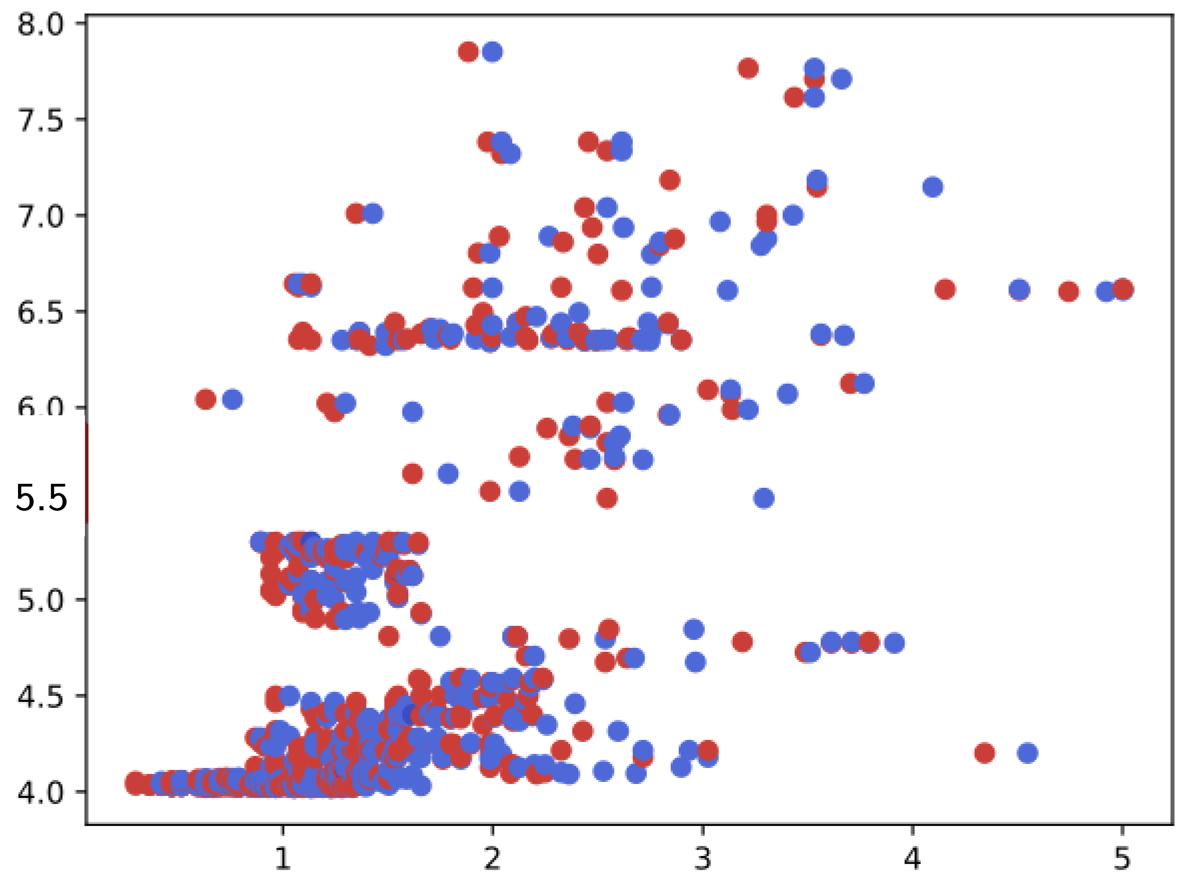




[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

Lipschitz Embedding of Signed Measures

μ_M

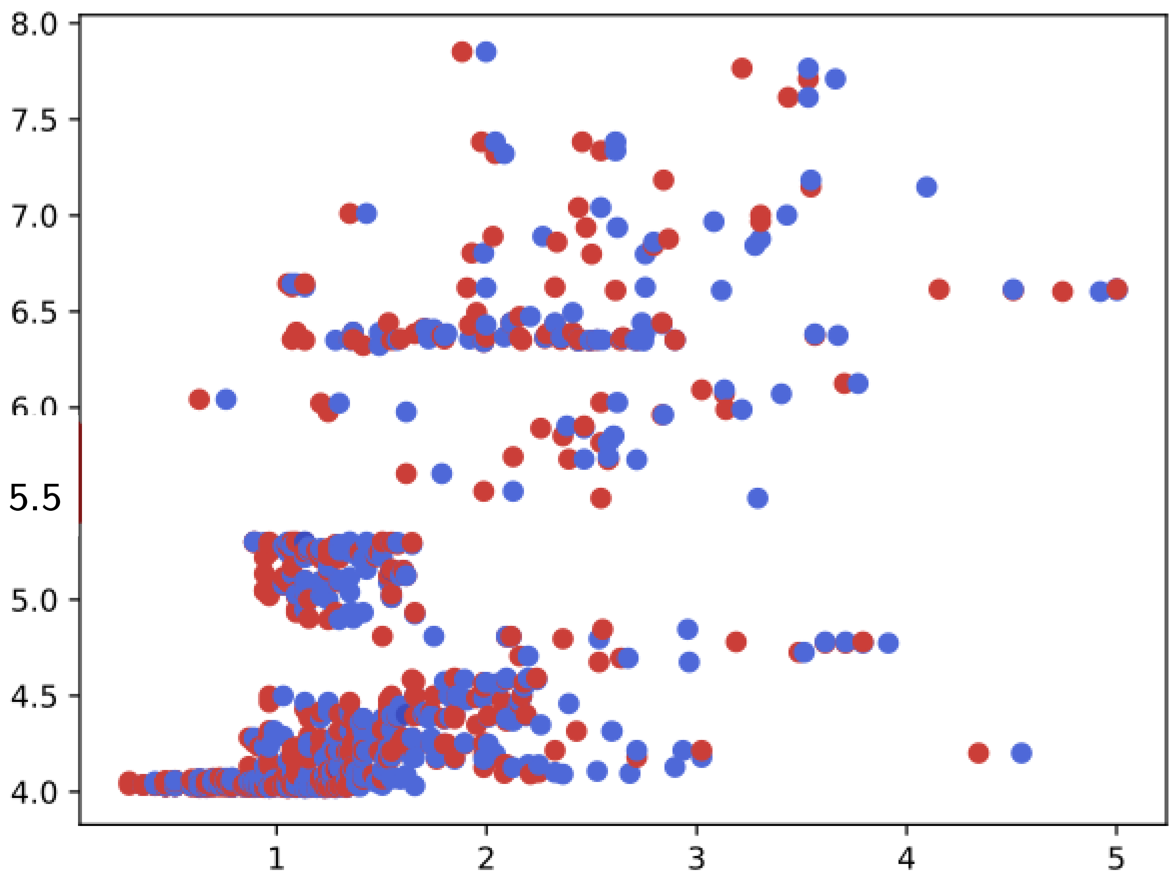


[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

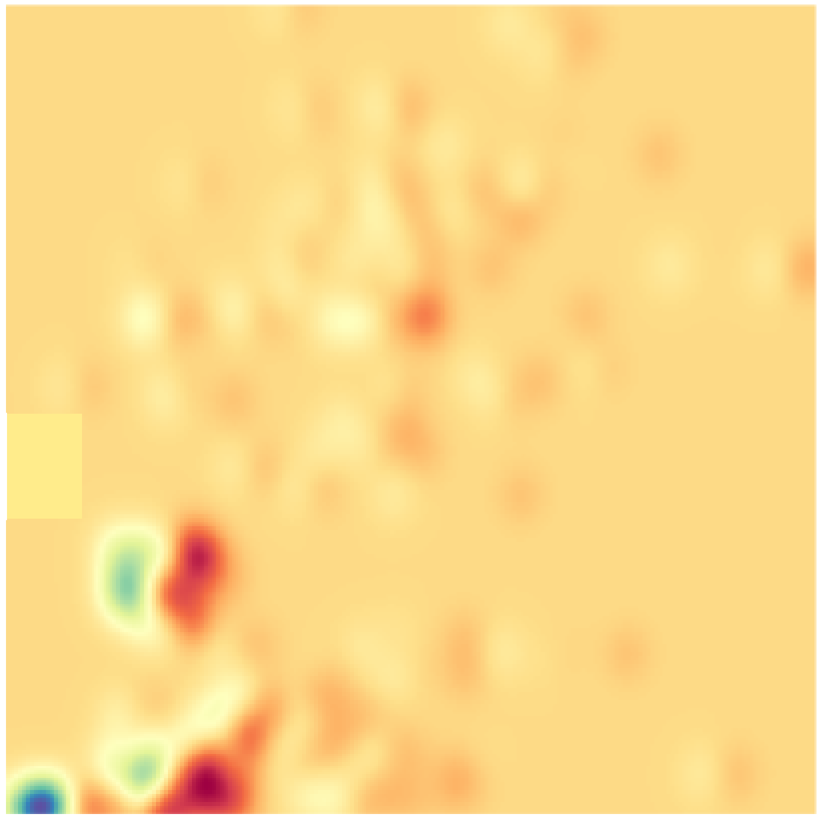
Lipschitz Embedding of Signed Measures

Efficient embedding can be done with *convolution* with a kernel k, e.g., $k(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\|x-y\|_2^2}{2\sigma^2}\right)$:

μ_M



$$\Phi(M(f))(x) := \mu_M \star k(x) = \int_y k(x,y)\,\mathrm{d}\mu_M(y)$$



[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

Lipschitz Embedding of these Signed Measures

Def: Decompose μ_M into $\mu_M = \mu_M^+ - \mu_M^-$. Then:

$$\|\mu_M\|_p^K := W_p(\mu_M^+, \mu_M^-)$$

$$d_{K,p}(\mu_M, \mu_{M'}) := \|\mu_M - \mu_{M'}\|_p^K = W_p(\mu_M^+ + \mu_{M'}^-, \mu_M^- + \mu_{M'}^+)$$

[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

Lipschitz Embedding of these Signed Measures

Def: Decompose μ_M into $\mu_M = \mu_M^+ - \mu_M^-$. Then:

$$\|\mu_M\|_p^K := W_p(\mu_M^+, \mu_M^-)$$

$$d_{K,p}(\mu_M, \mu_{M'}) := \|\mu_M - \mu_{M'}\|_p^K = W_p(\mu_M^+ + \mu_{M'}^-, \mu_M^- + \mu_{M'}^+)$$

Prop: For any functions $f, g : S \to \mathbb{R}^2$, where S is a finite simplicial complex, one has $d_{K,1}(\mu_{M(f)}, \mu_{M(g)}) = \|\mu_{M(f)} - \mu_{M(g)}\|_1^K \leq 2\|f - g\|_1$.

[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

Lipschitz Embedding of these Signed Measures

Def: Decompose μ_M into $\mu_M = \mu_M^+ - \mu_M^-$. Then:

$$\|\mu_M\|_p^K := W_p(\mu_M^+, \mu_M^-)$$

$$d_{K,p}(\mu_M, \mu_{M'}) := \|\mu_M - \mu_{M'}\|_p^K = W_p(\mu_M^+ + \mu_{M'}^-, \mu_M^- + \mu_{M'}^+)$$

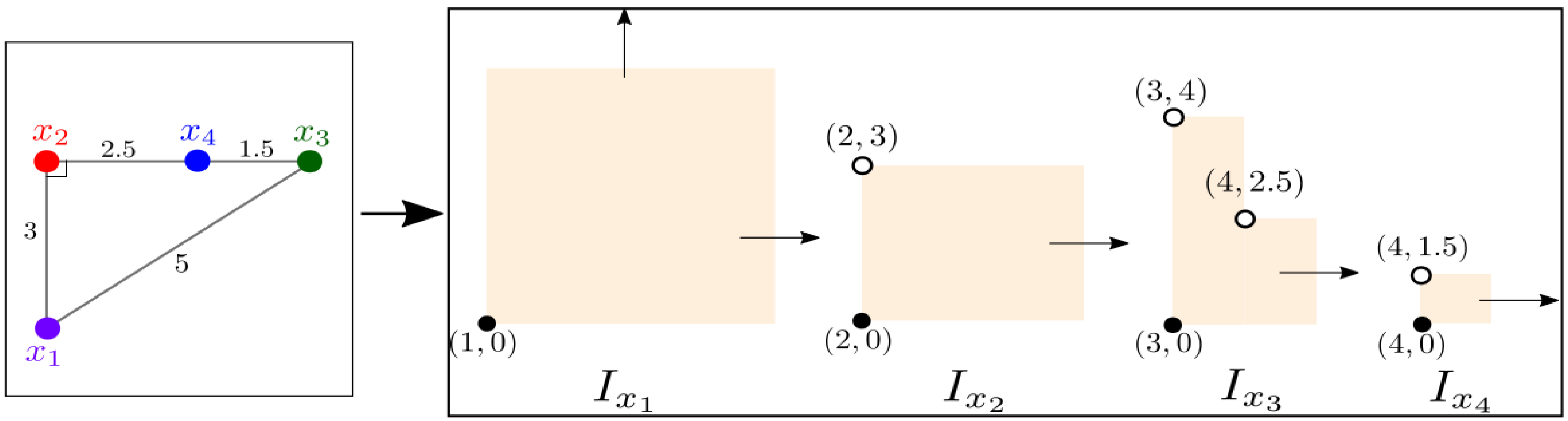
Prop: For any functions $f, g : S \to \mathbb{R}^2$, where S is a finite simplicial complex, one has $d_{K,1}(\mu_{M(f)}, \mu_{M(g)}) = \|\mu_{M(f)} - \mu_{M(g)}\|_1^K \leq 2\|f - g\|_1$.

Prop: Let k be a kernel such that $\|k(x, \cdot) - k(\cdot, y)\|_2 \leq C_k \cdot \|x - y\|_2$. Then, one has $\|k \star \mu - k \star \nu\|_2 \leq C'_k \|\mu - \nu\|_2^K \leq C'_k \|\mu - \nu\|_1^K$.

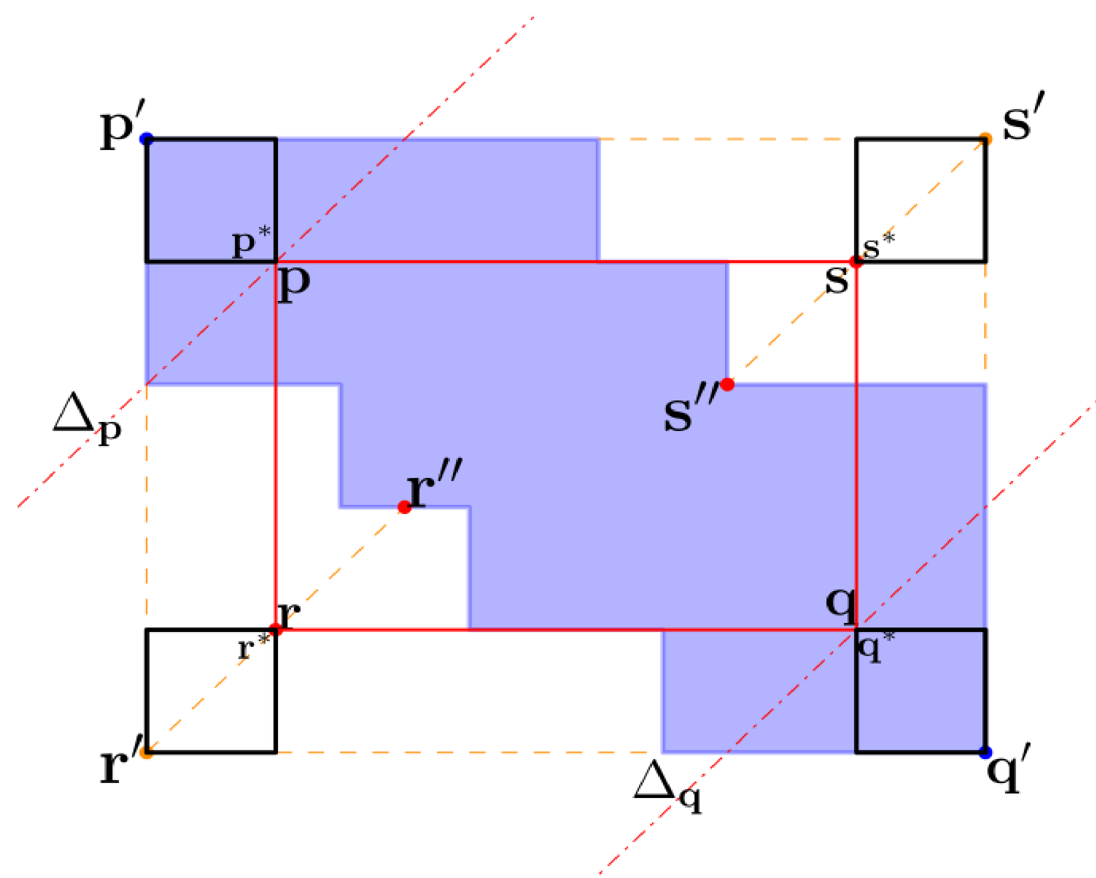
[Loiseaux et al. **Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures.** 2023]

Other Decomposition Candidates

[Cai et al. **Elder-Rule-Staircodes for Augmented Metric Spaces.** 2020]

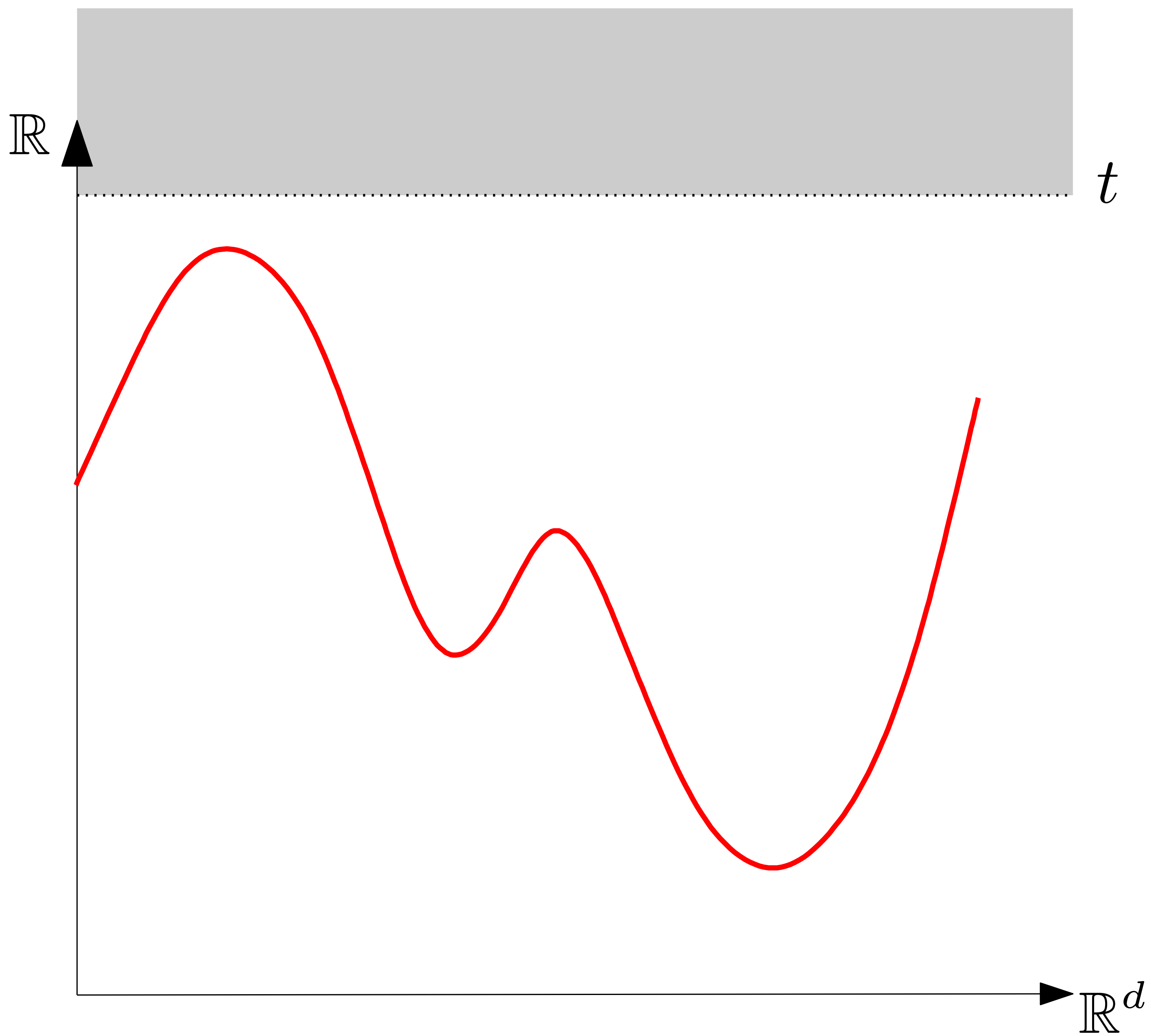


[Dey, Xin. **Rectangular Approximation and Stability of 2-parameter Persistence Modules.** 2021]



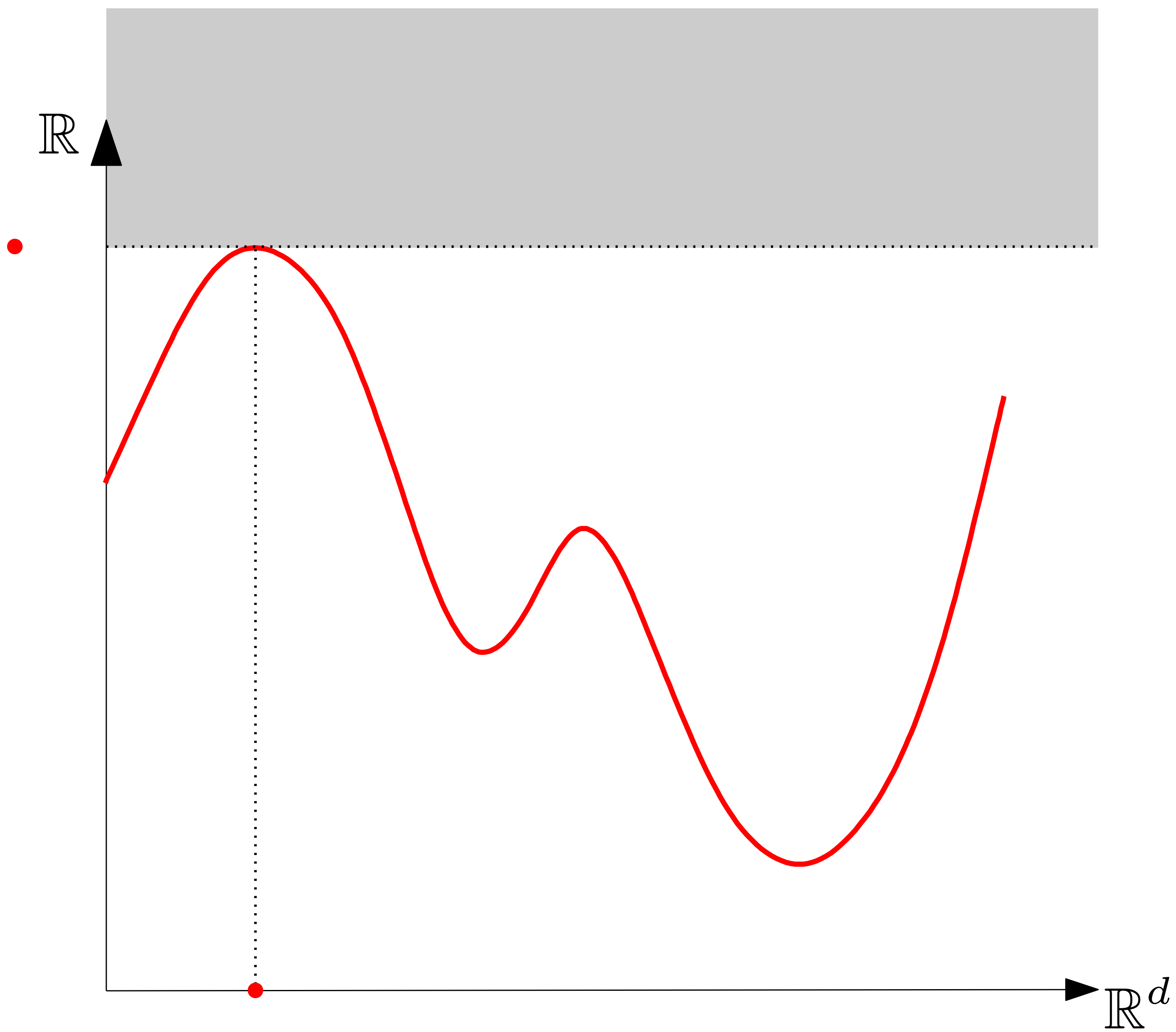
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



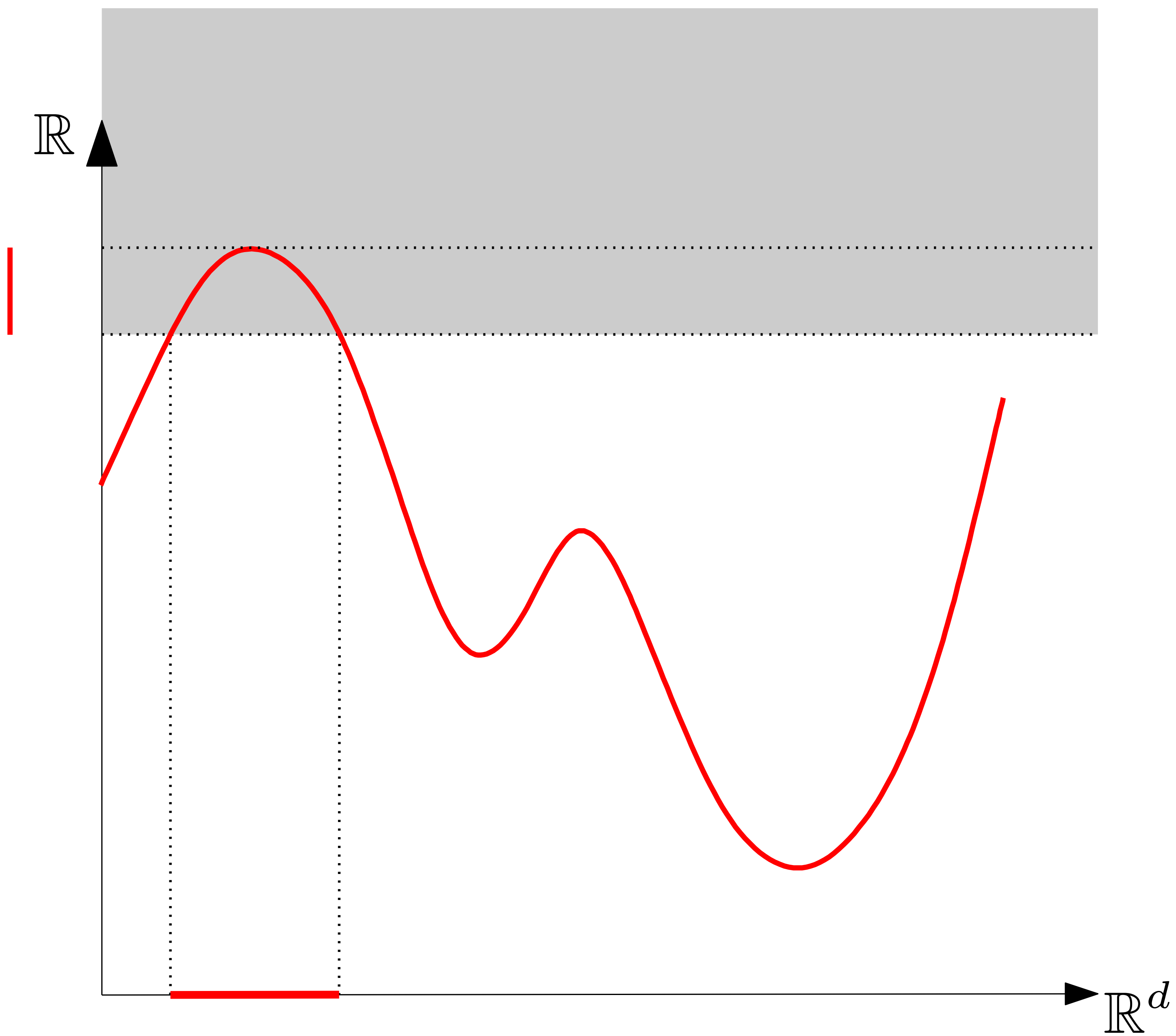
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t,+\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



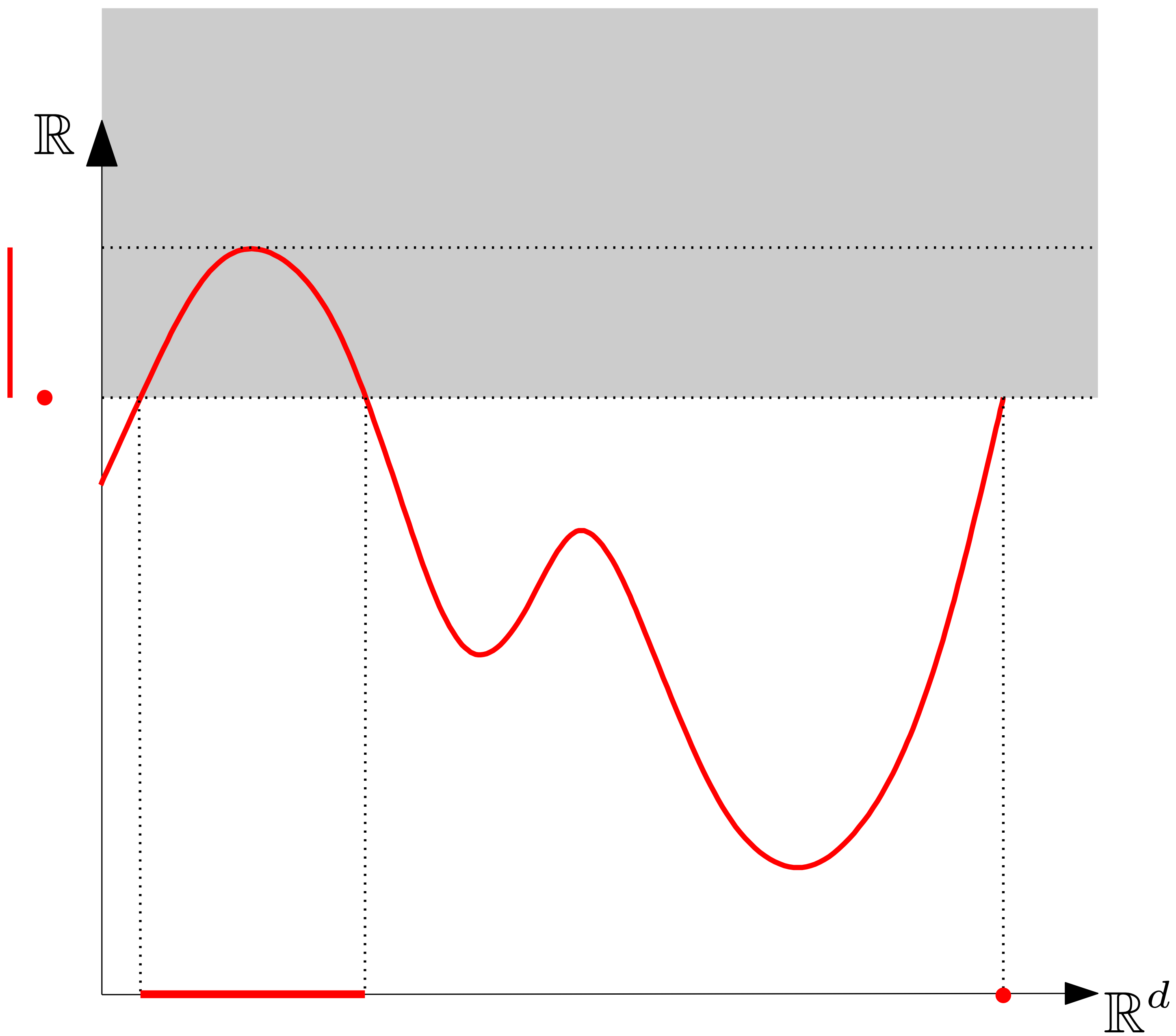
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t,+\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



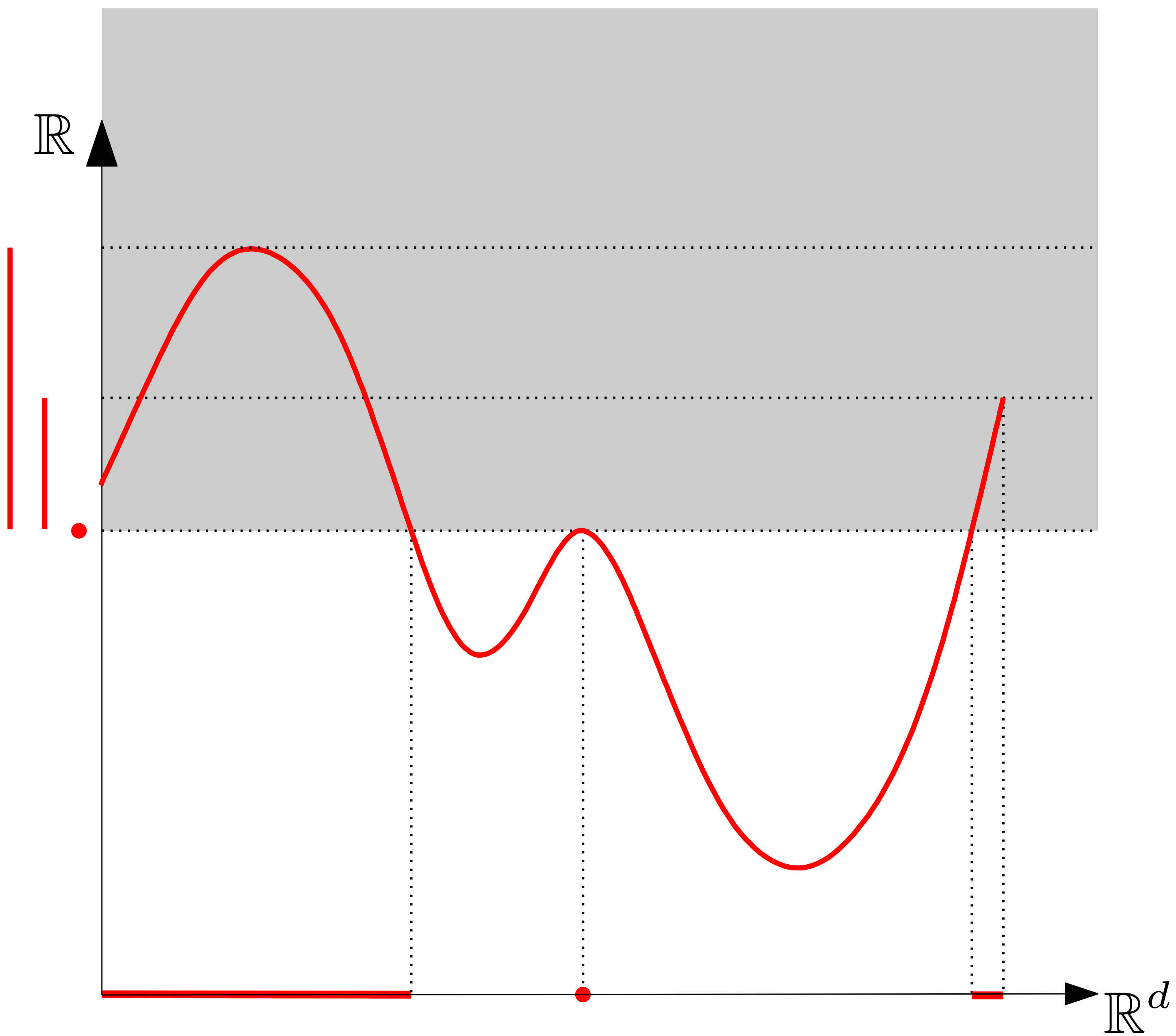
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



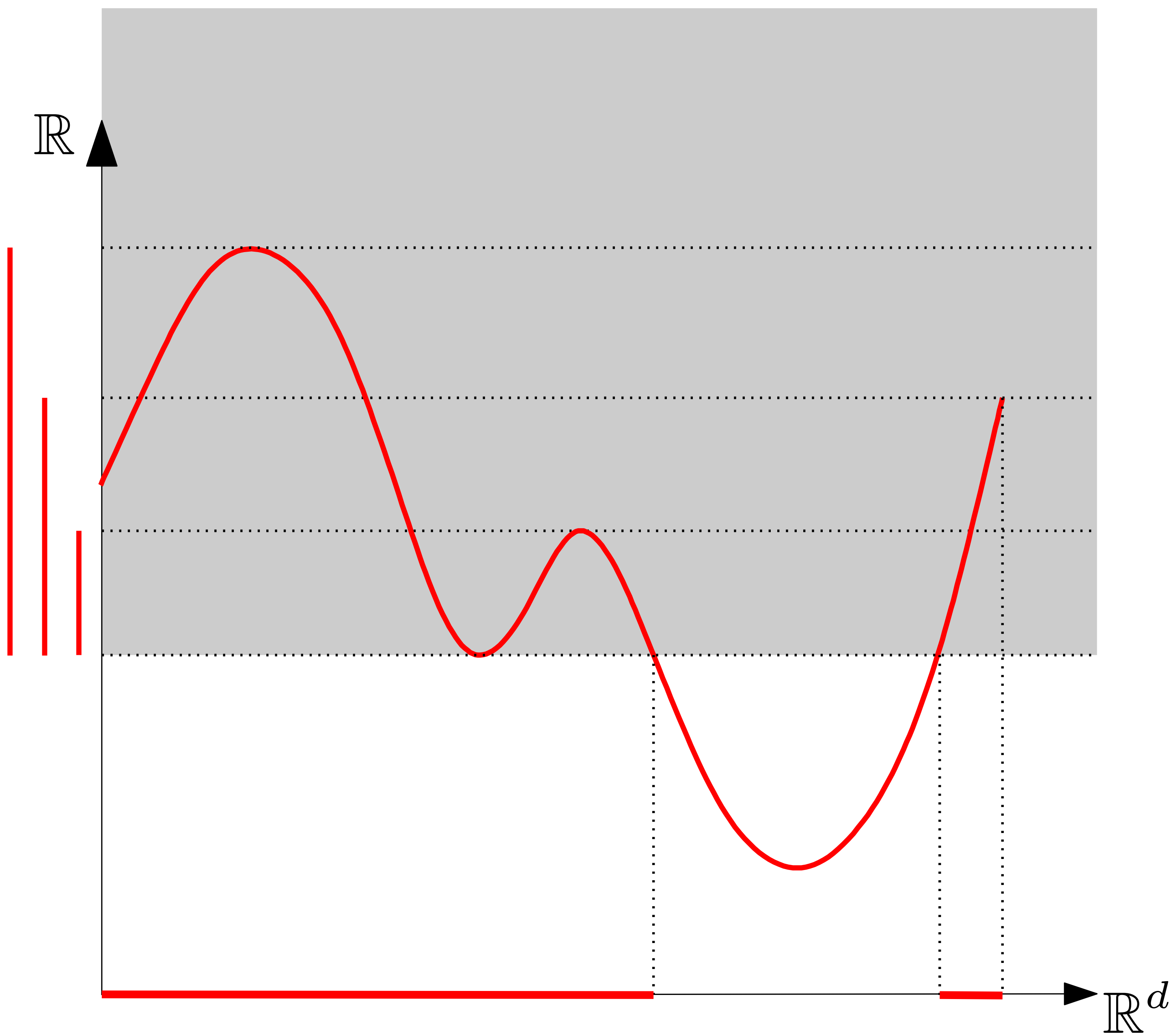
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



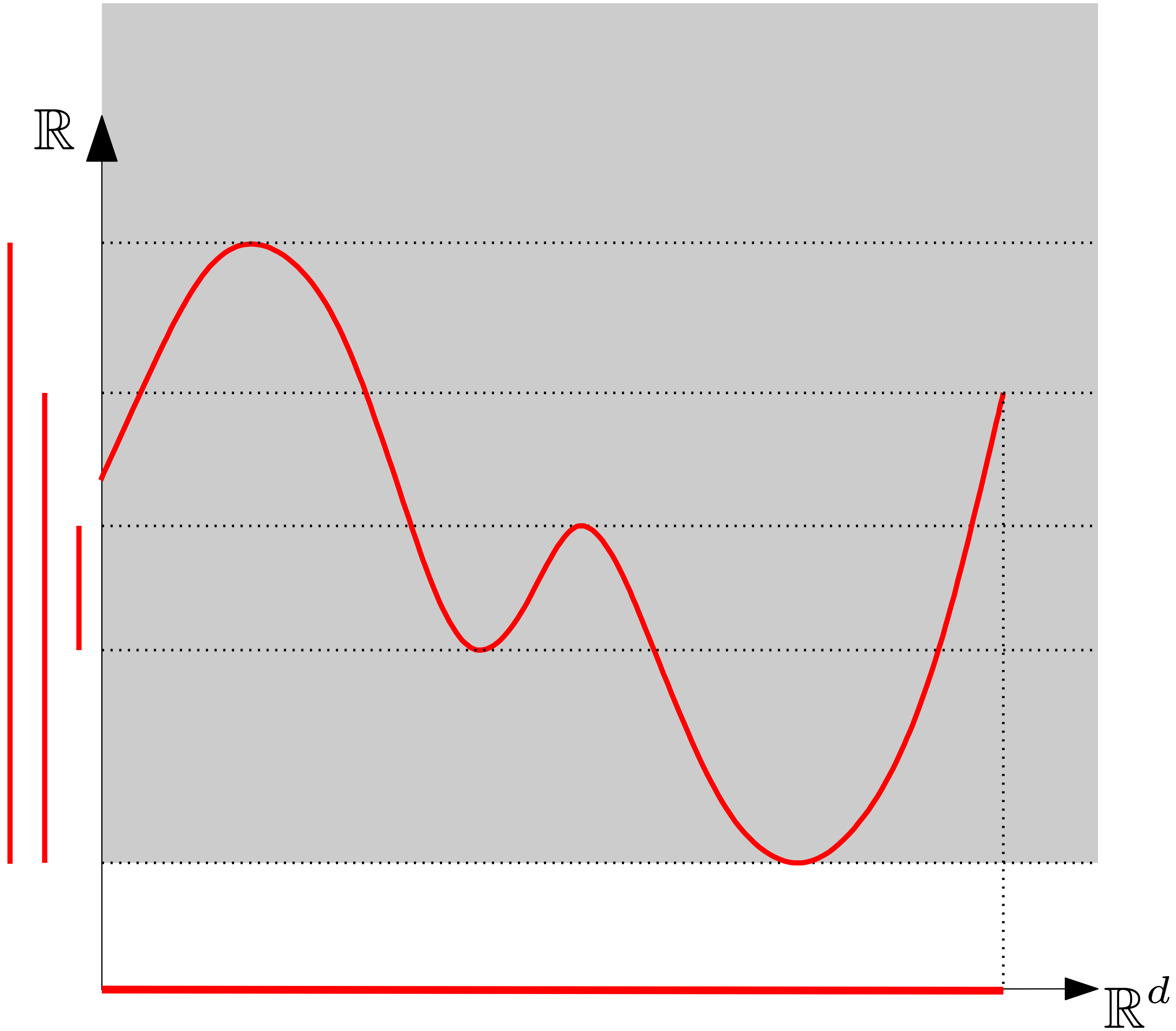
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



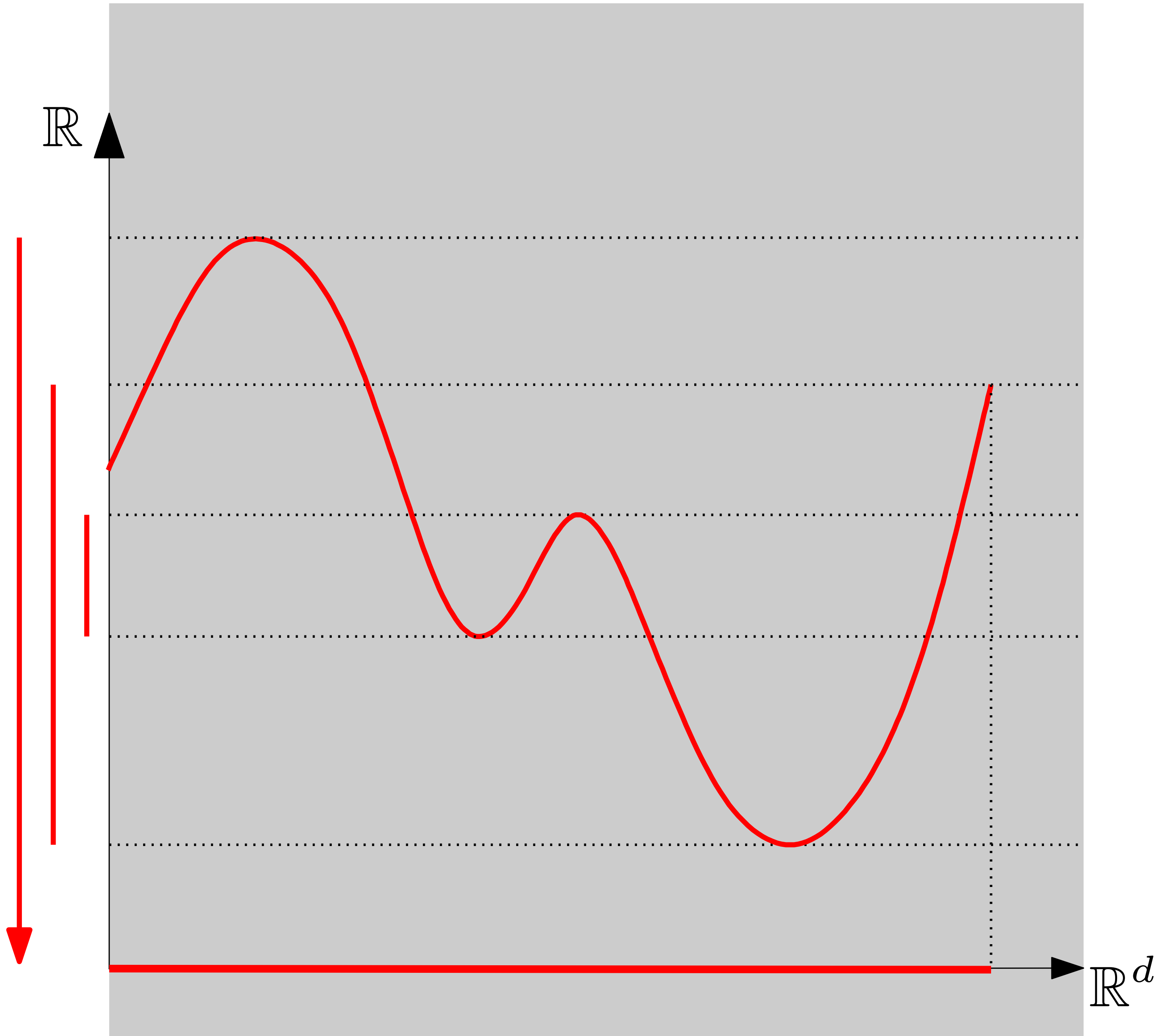
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t,+\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



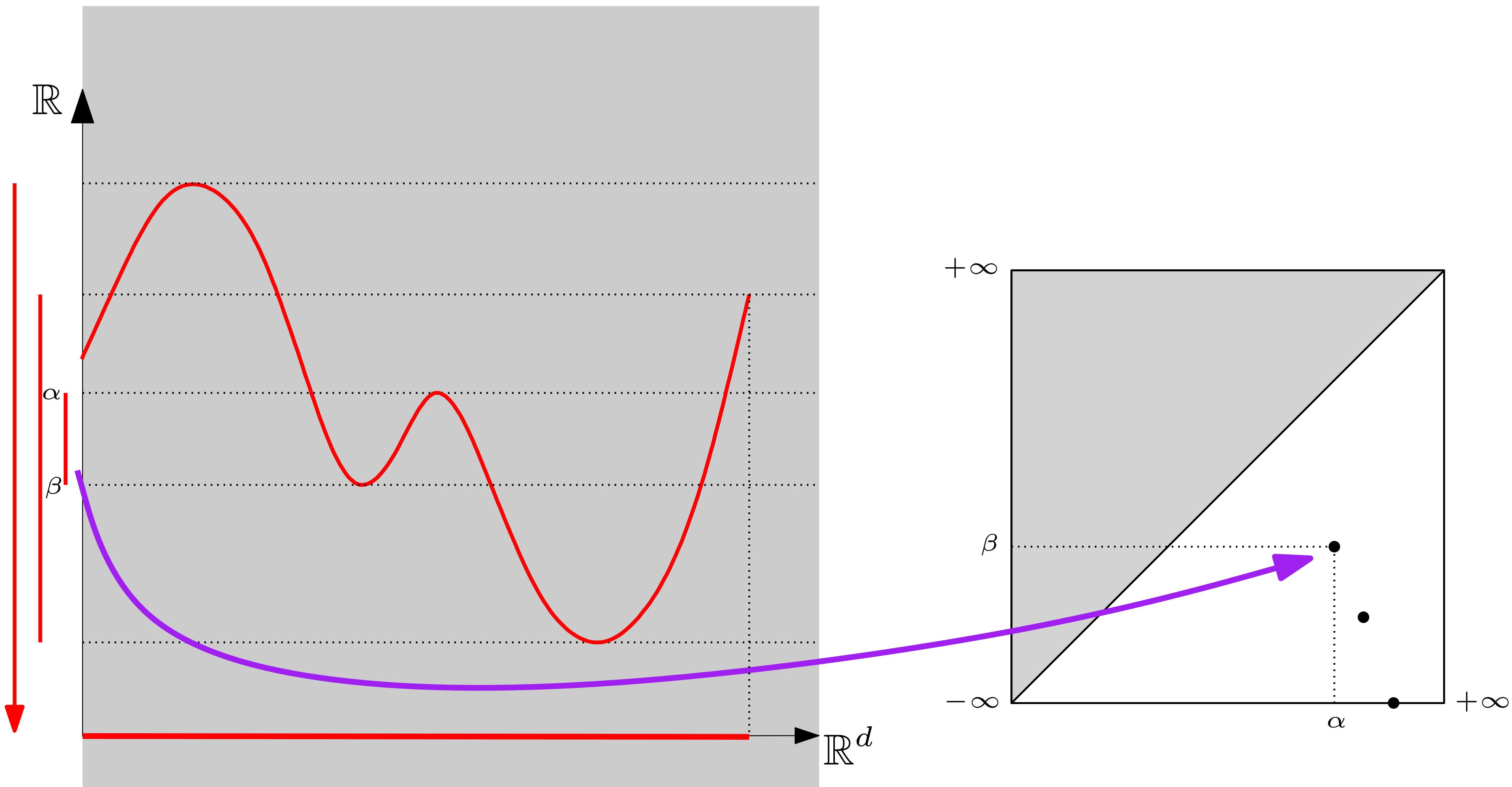
Reminder: 0-dimensional PH of density

Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.



Reminder: 0-dimensional PH of density

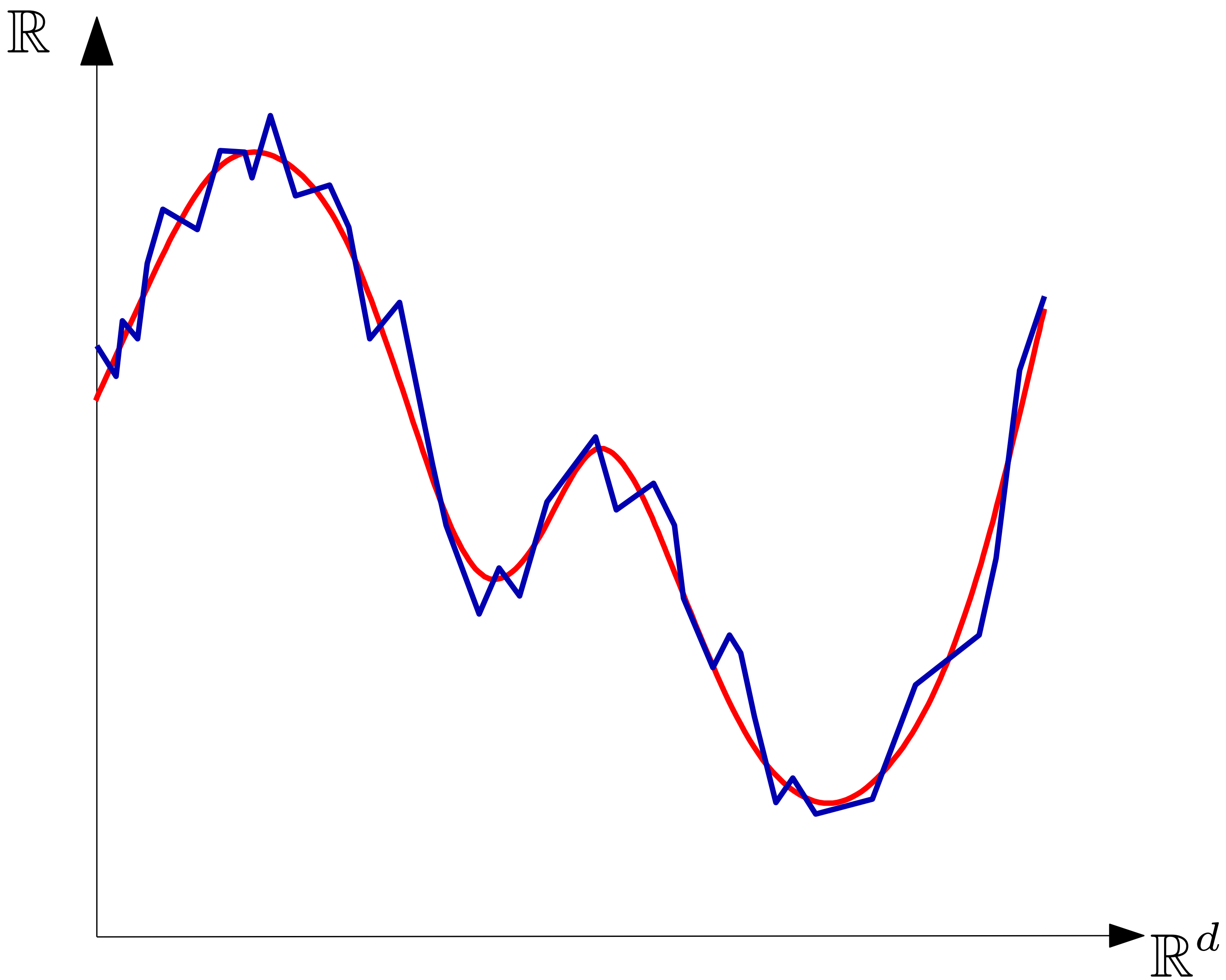
Given a probability density f, we will consider the superlevel-set filtration $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, instead of the sublevel-set filtration.

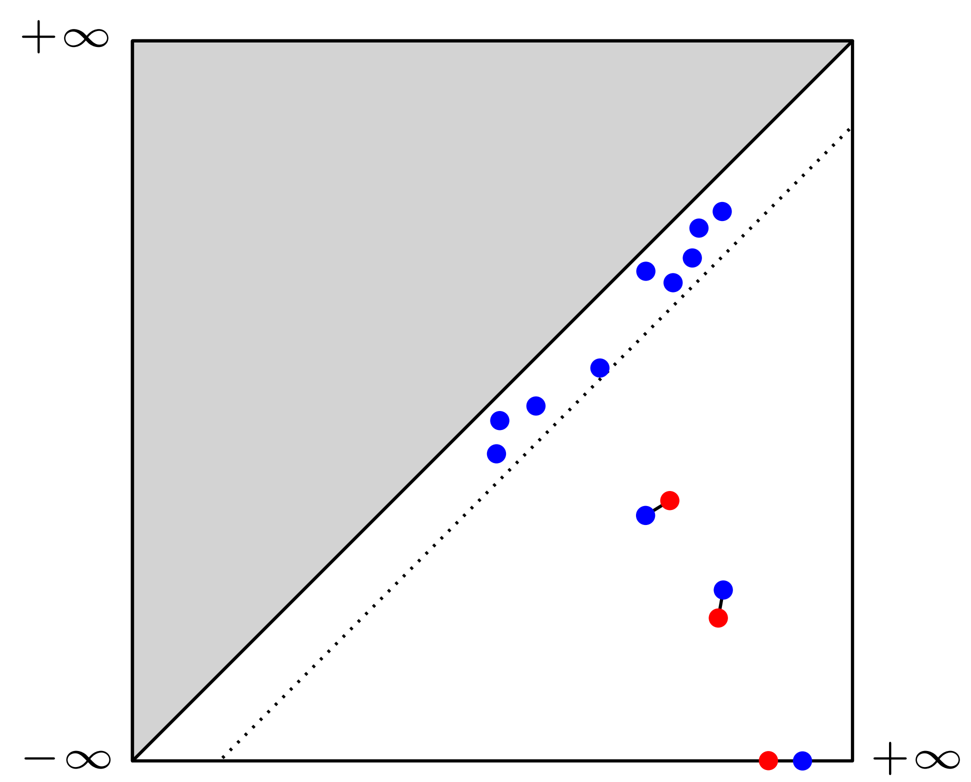


Reminder: 0-dimensional PH of density

Moreover, the stability theorem ensures that, given an underlying true density f, and an estimator $\hat{f}$ ot it, one has:

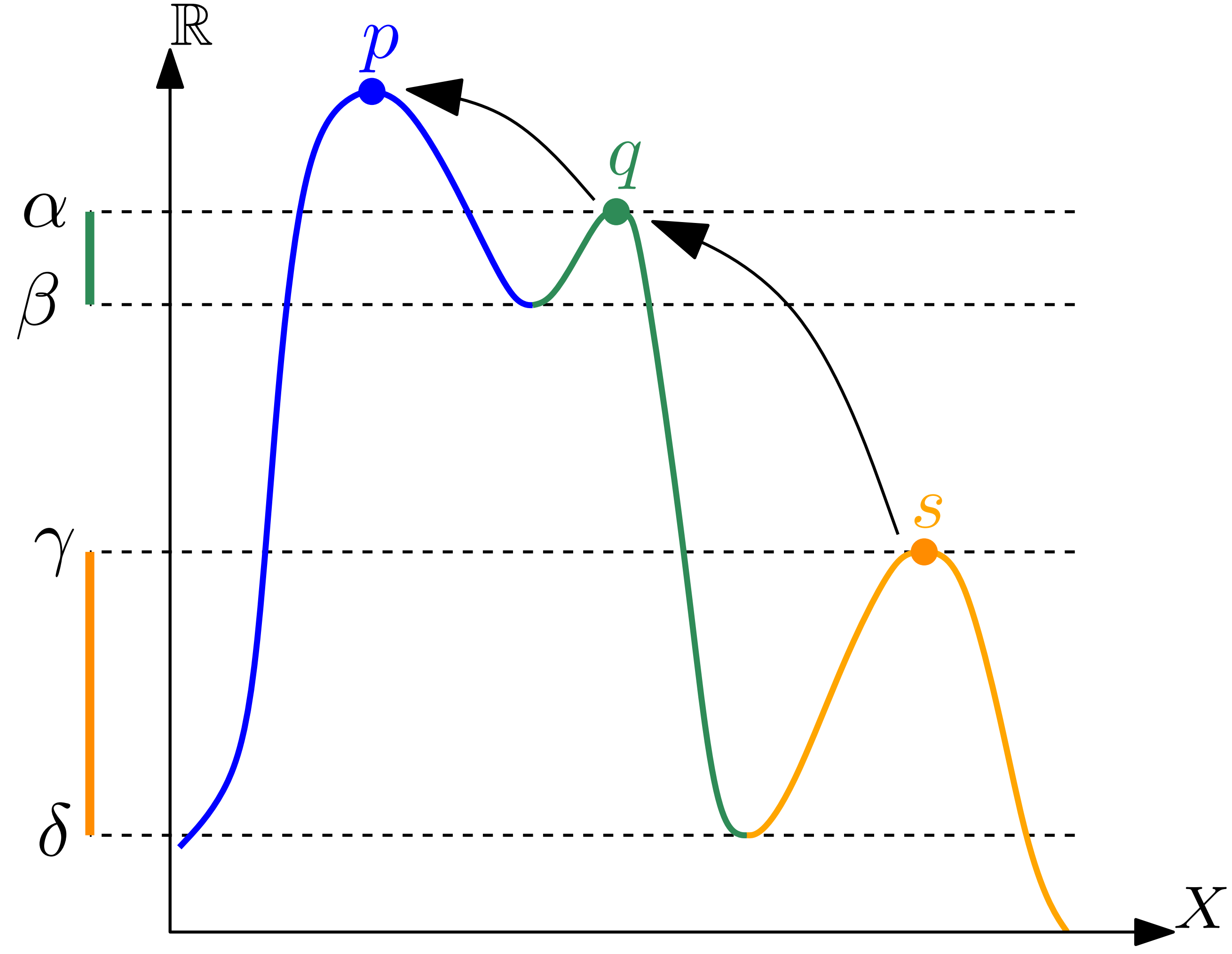
$$d_b(D_f, D_{\hat{f}}) \leq \|f - \hat{f}\|_\infty.$$





Building a hierarchy of cluster with 0-dimensional PH

In addition to being stable, 0-dimensional PH also remembers the connected components that were merged together during the filtration process and builds a hierarchy out of this information.

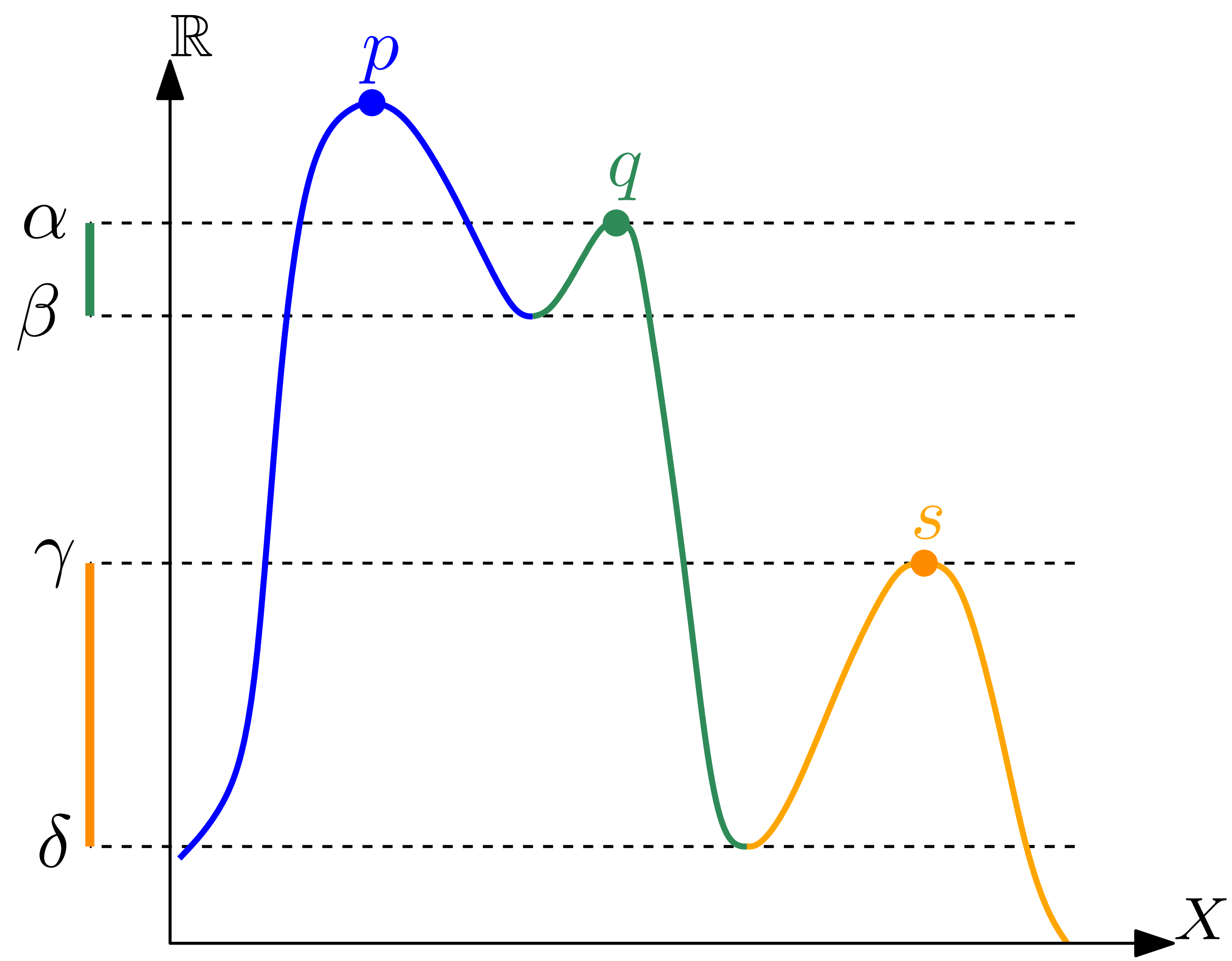


Building a hierarchy of cluster with 0-dimensional PH

In addition to being stable, 0-dimensional PH also remembers the connected components that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the clusters associated to all the bars of length (or prominence) $> \tau$!

$$0 \leq \tau \leq \alpha - \beta$$

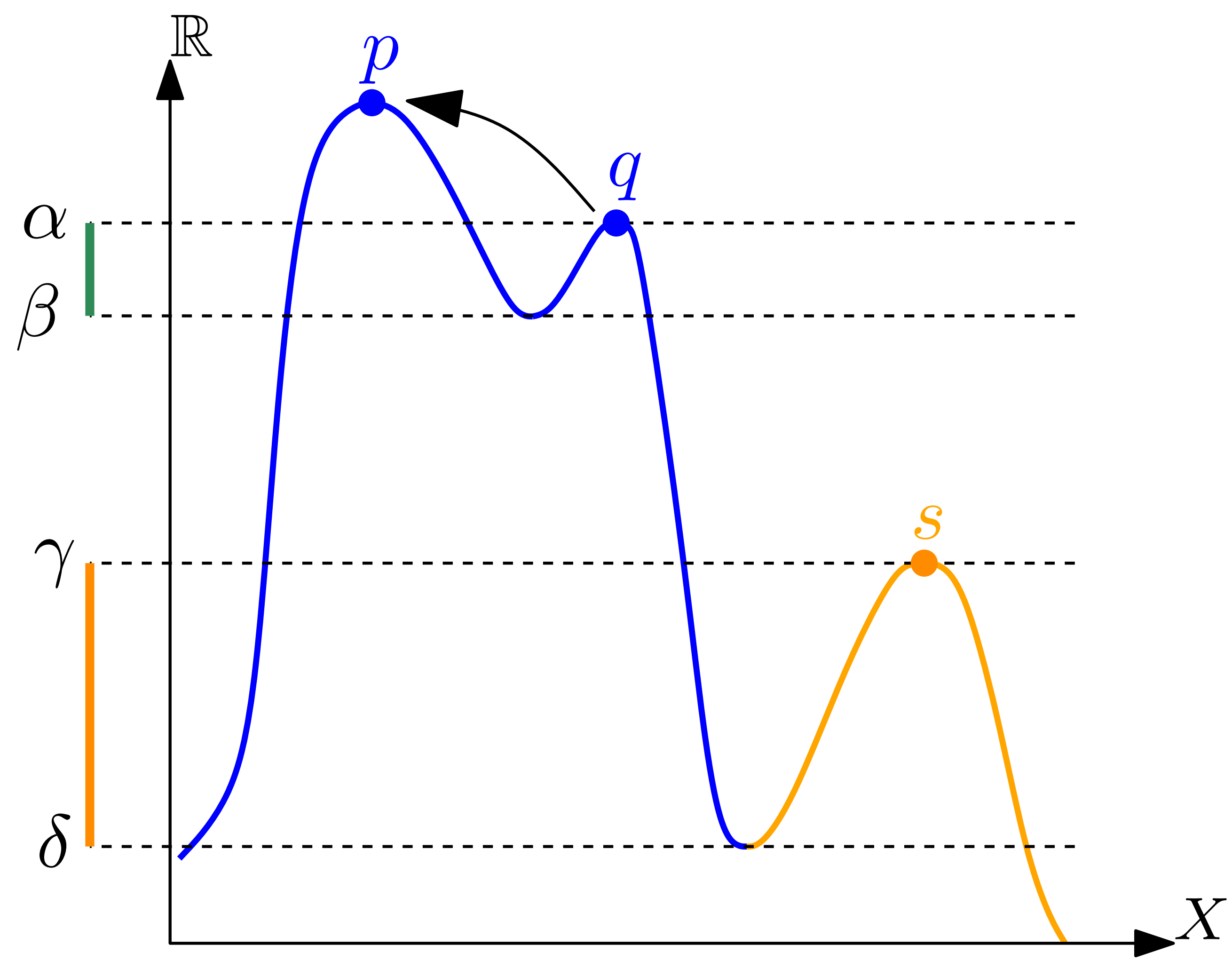


Building a hierarchy of cluster with 0-dimensional PH

In addition to being stable, 0-dimensional PH also remembers the connected components that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the clusters associated to all the bars of length (or prominence) $> \tau$!

$$\alpha - \beta < \tau \leq \gamma - \delta$$

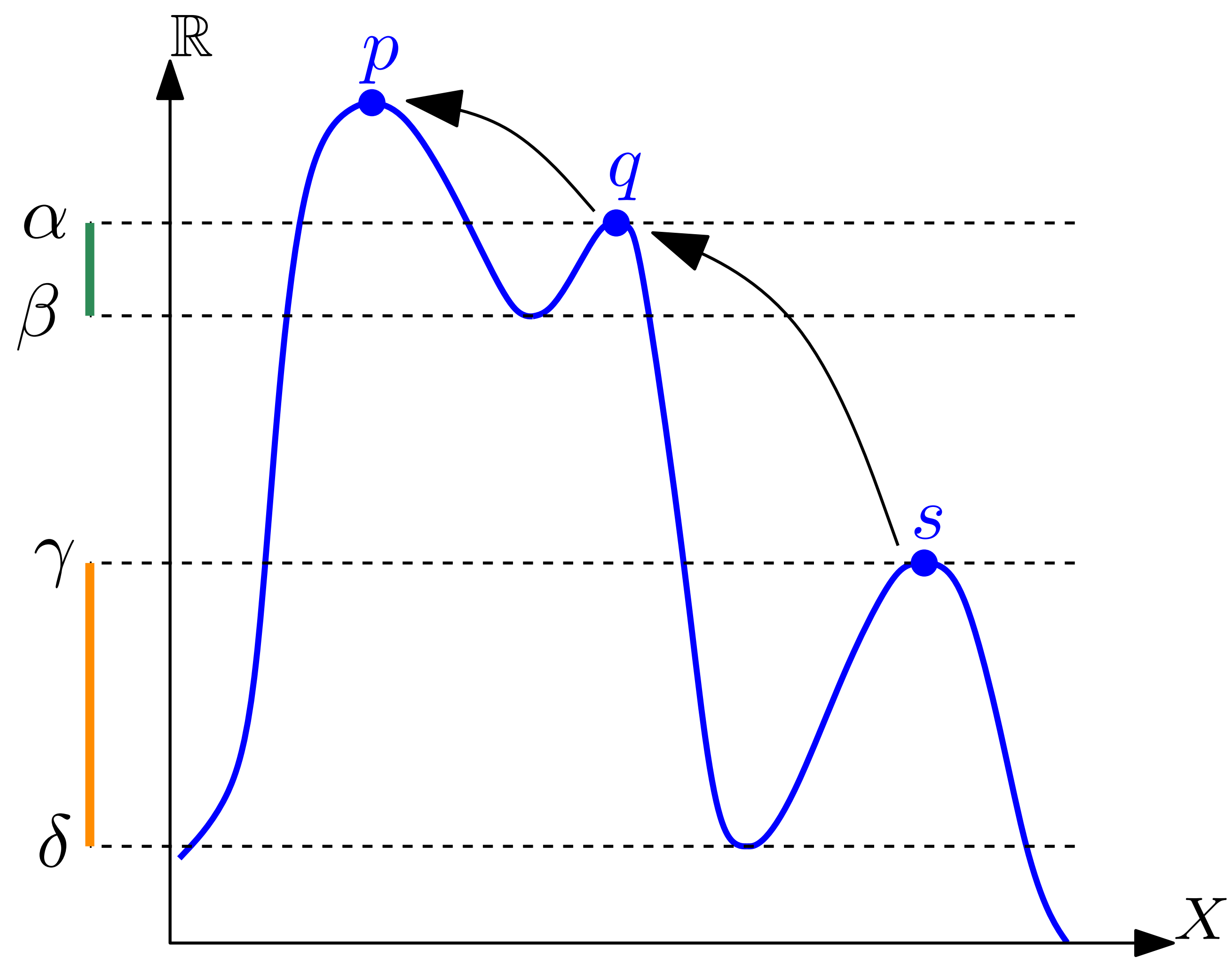


Building a hierarchy of cluster with 0-dimensional PH

In addition to being stable, 0-dimensional PH also remembers the connected components that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the clusters associated to all the bars of length (or prominence) $> \tau$!

$$\gamma - \delta < \tau \leq +\infty$$

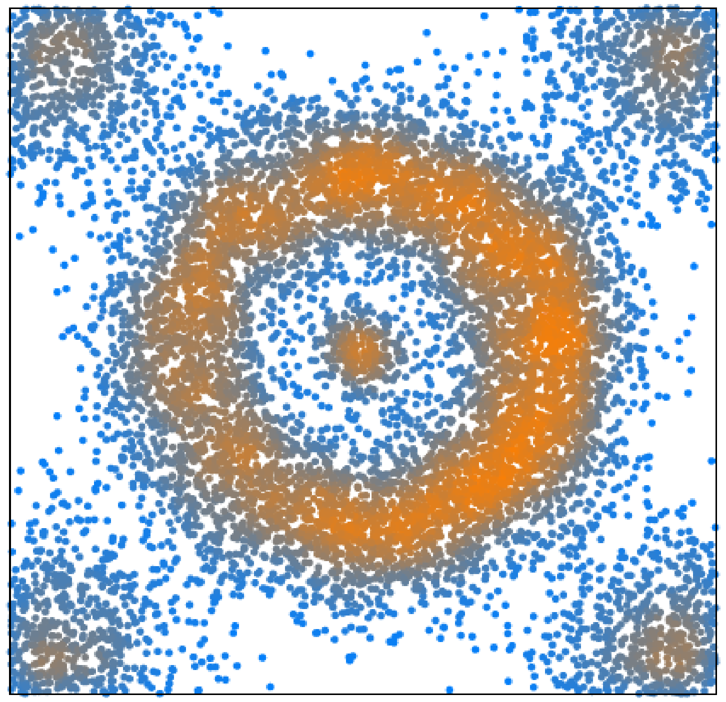


ToMATo: Topological Mode Analysis Tool

[*Persistence-Based Clustering in Riemannian Manifolds*, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

1. Define an order on the point cloud with a density estimator $\hat{f}$.

   (sort data points by **decreasing** estimated density values)



ToMATo: Topological Mode Analysis Tool

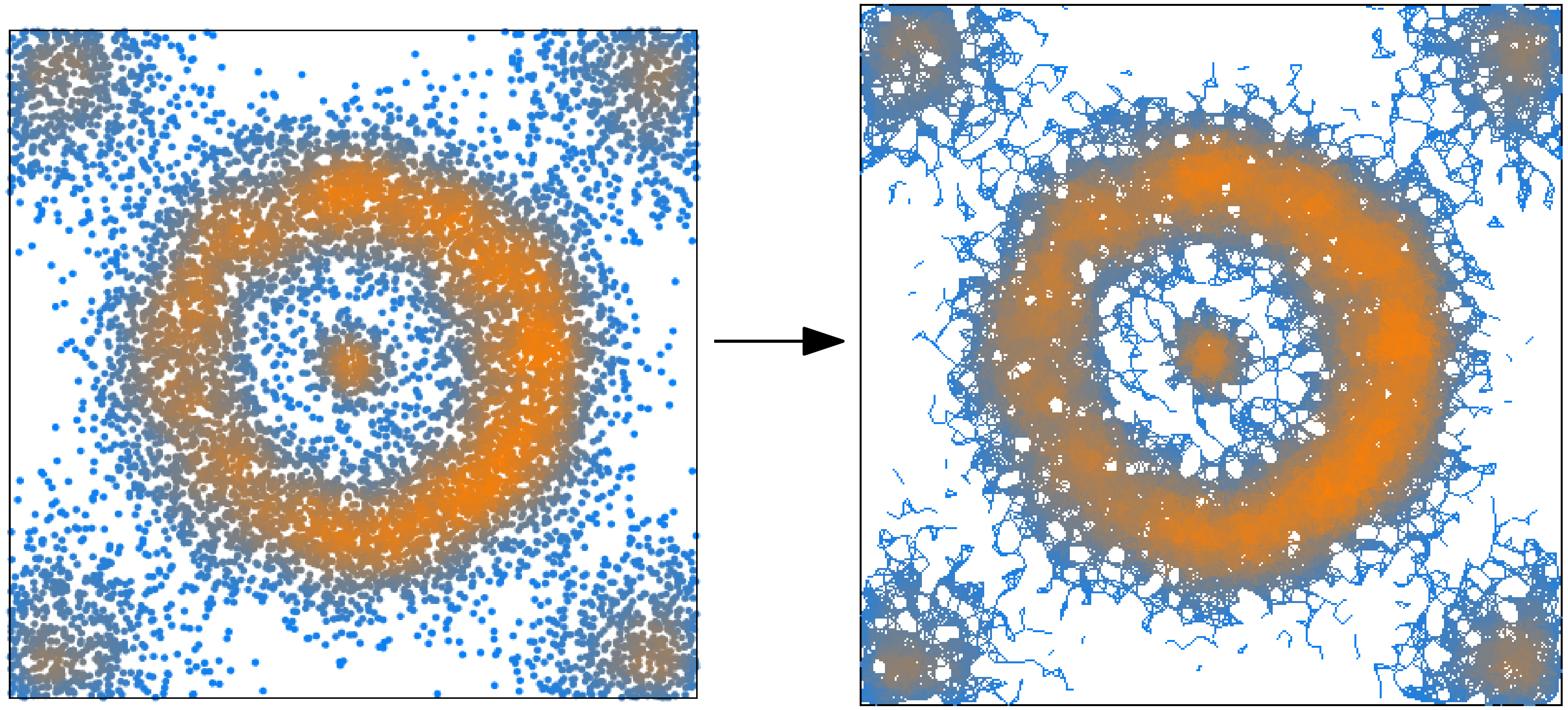
[*Persistence-Based Clustering in Riemannian Manifolds*, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

1. Define an order on the point cloud with a density estimator $\hat{f}$.

   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$



ToMATo: Topological Mode Analysis Tool

[*Persistence-Based Clustering in Riemannian Manifolds*, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

1. Define an order on the point cloud with a density estimator $\hat{f}$.

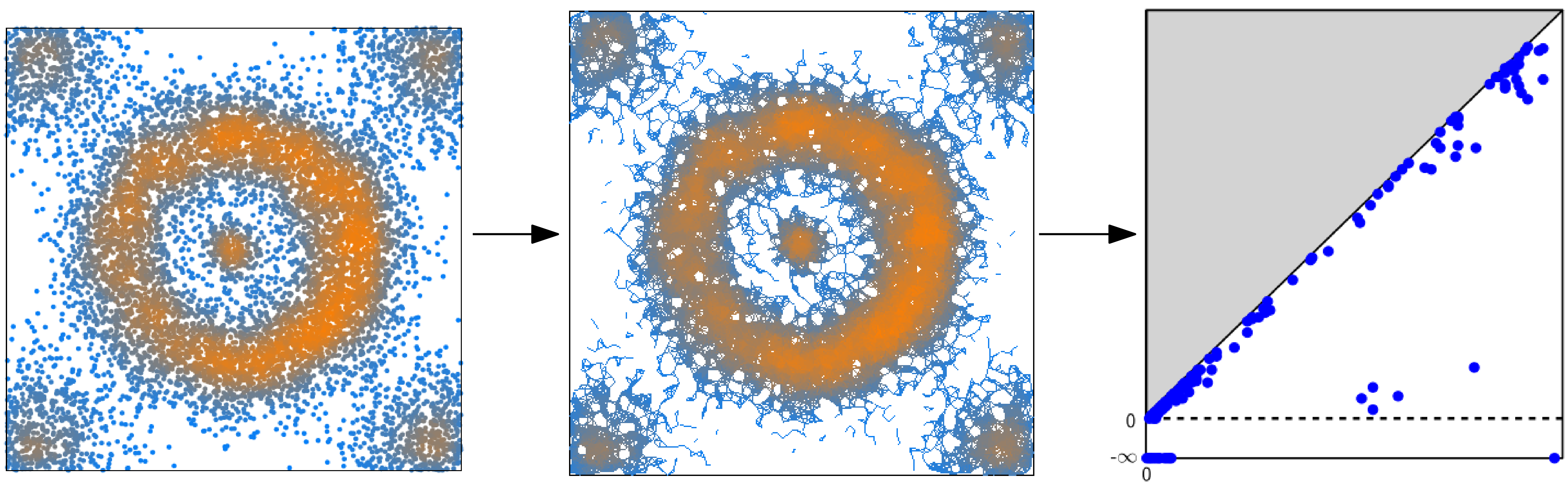
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



ToMATo: Topological Mode Analysis Tool

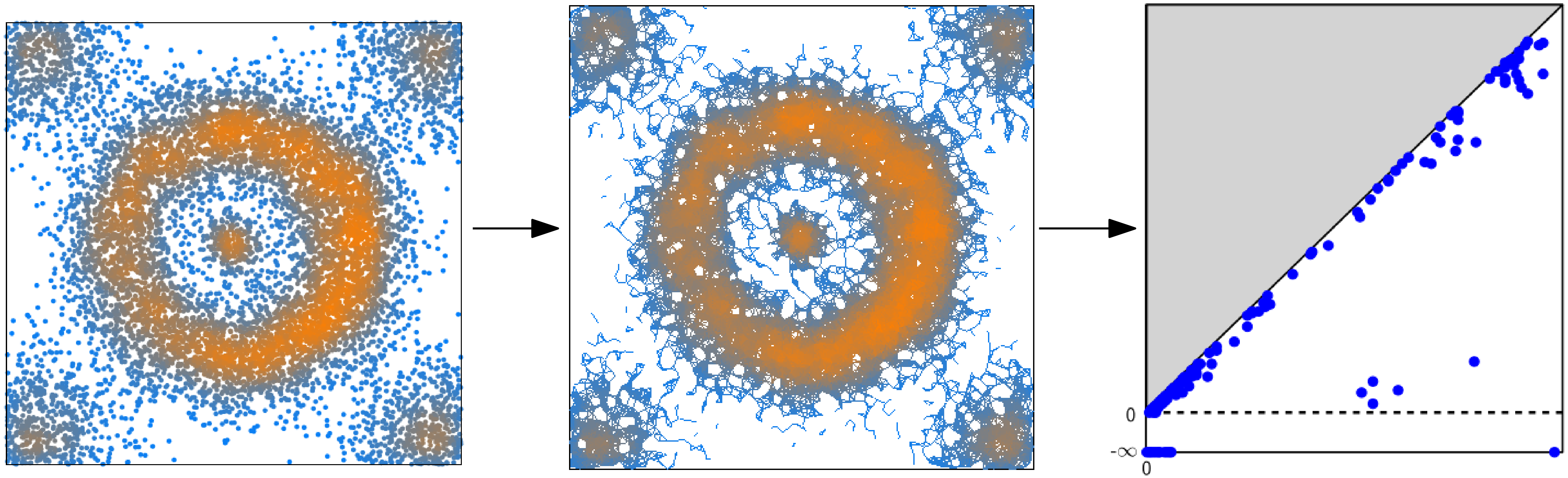
[*Persistence-Based Clustering in Riemannian Manifolds*, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

Given a neighborhood graph with n vertices and m edges:

1. the algorithm sorts the vertices by decreasing density values,

2. and then makes a single pass through the vertex set, merging clusters on the fly using a union-find data structure.

$\rightarrow$ Running time: $O(n \log n + (n+m)\alpha(n))$
$\rightarrow$ Space complexity: $O(n+m)$
$\rightarrow$ Main memory usage: $O(n)$



Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator $\hat{f}$.

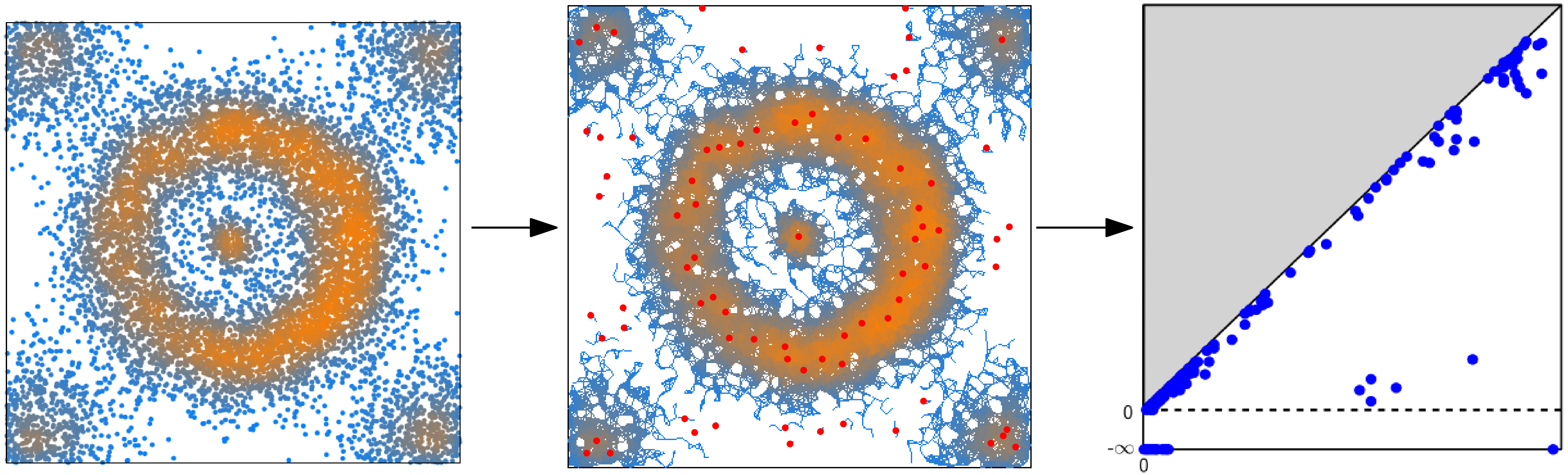
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator $\hat{f}$.

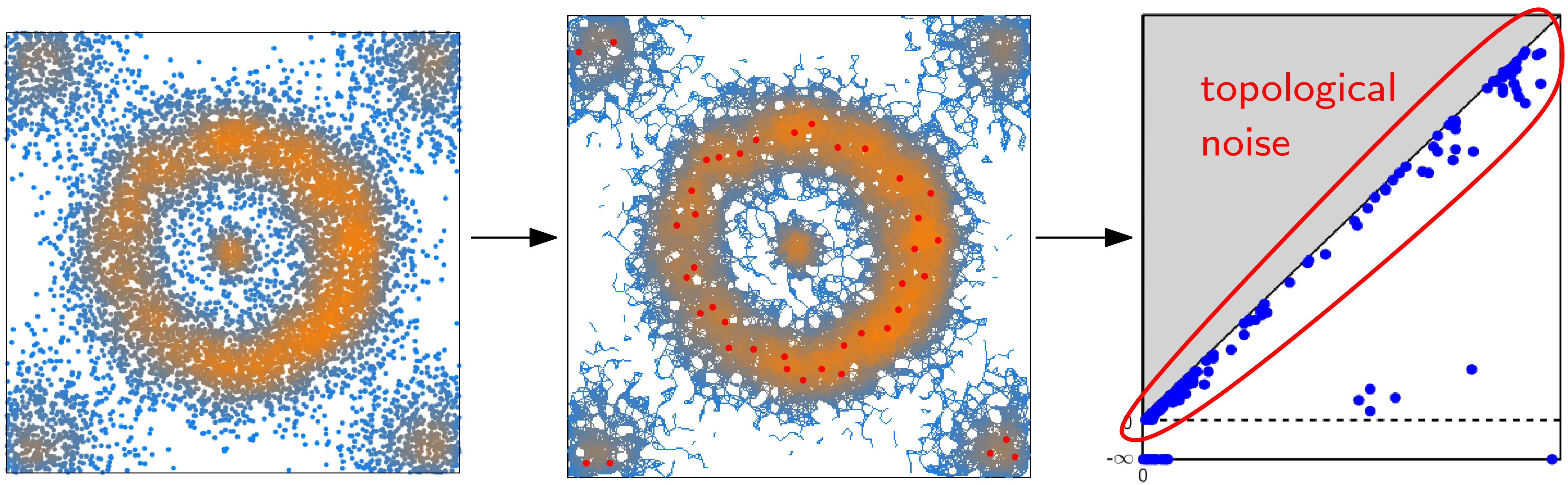
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator $\hat{f}$.

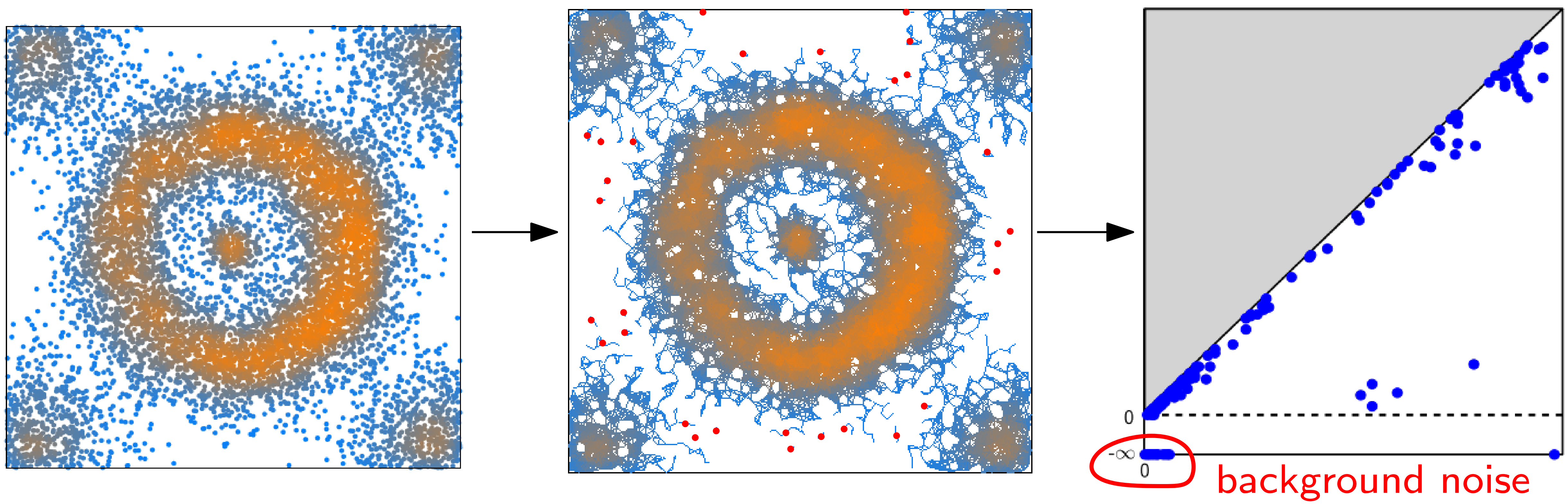
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator $\hat{f}$.

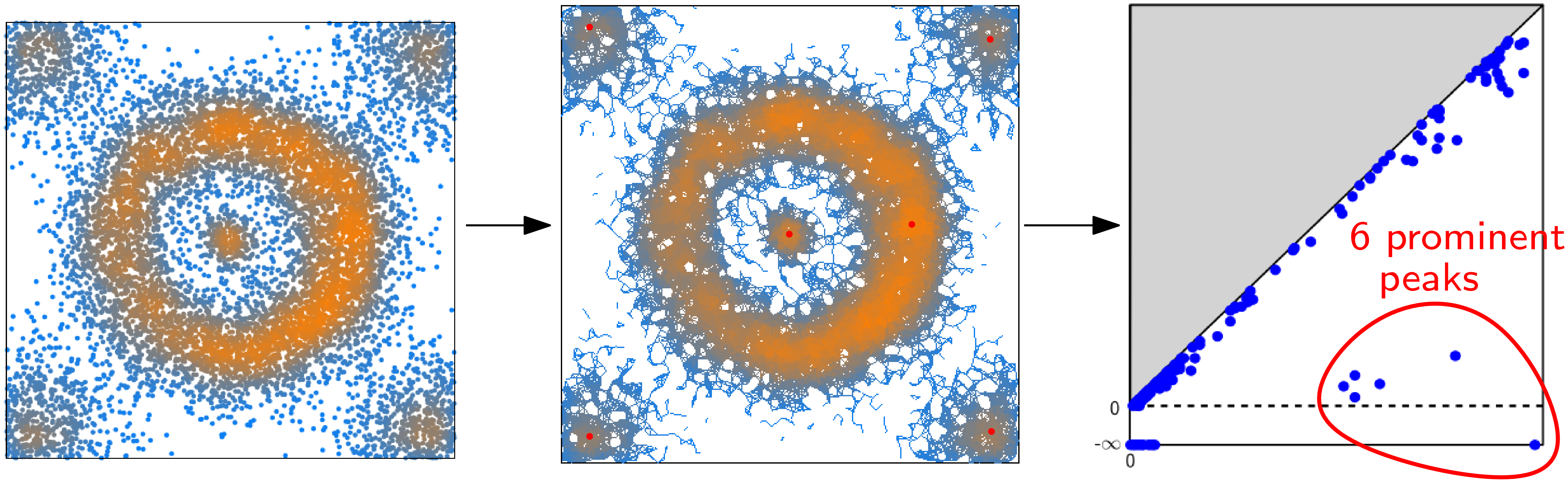
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u, v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator $\hat{f}$.

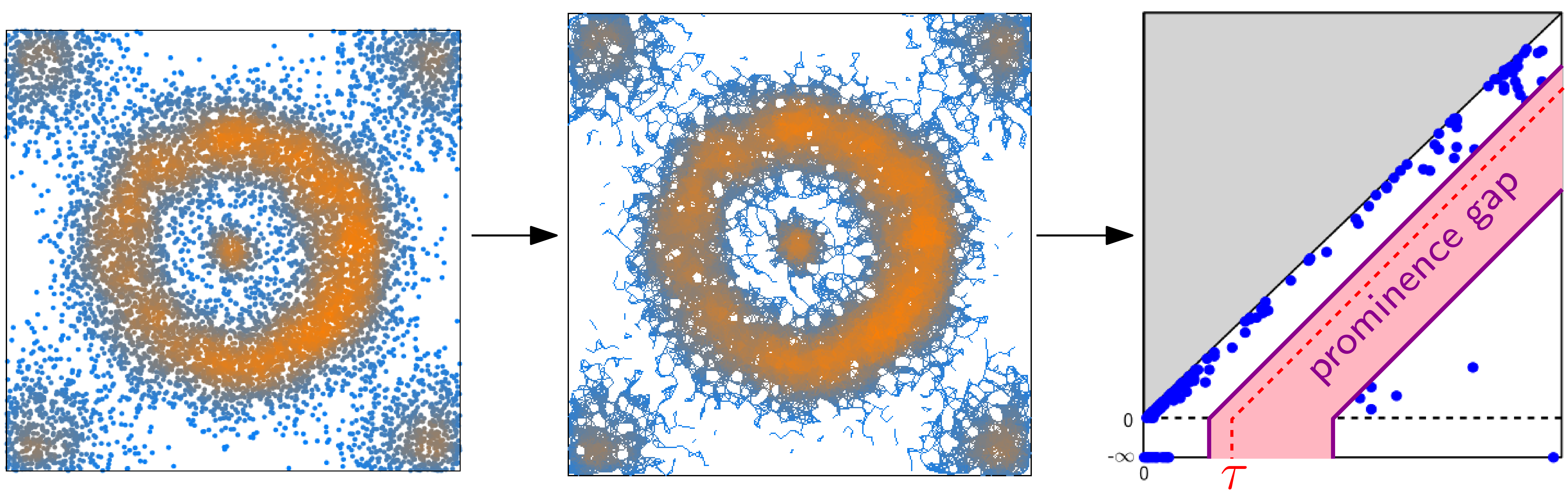
   (sort data points by **decreasing** estimated density values)

2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).

   $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

3. Compute the 0-dimensional persistence diagram of this filtration.

   (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the correct number of clusters

Hypotheses:

- $f : \mathbb{R}^d \to \mathbb{R}$ a c-Lipschitz probability density function,
- $P \subset \mathbb{R}^d$ a finite set of n points sampled i.i.d. according to f,
- $\hat{f} : P \to \mathbb{R}$ a density estimator s.t. $\eta := \max_{p \in P} |\hat{f}(p) - f(p)| < \Pi/5$,
- $G = (P, E)$ the δ-neighborhood graph for some positive $\delta < \frac{\Pi - 5\eta}{5c}$.

Note: Π is the prominence of the least prominent peak of f

Estimating the correct number of clusters

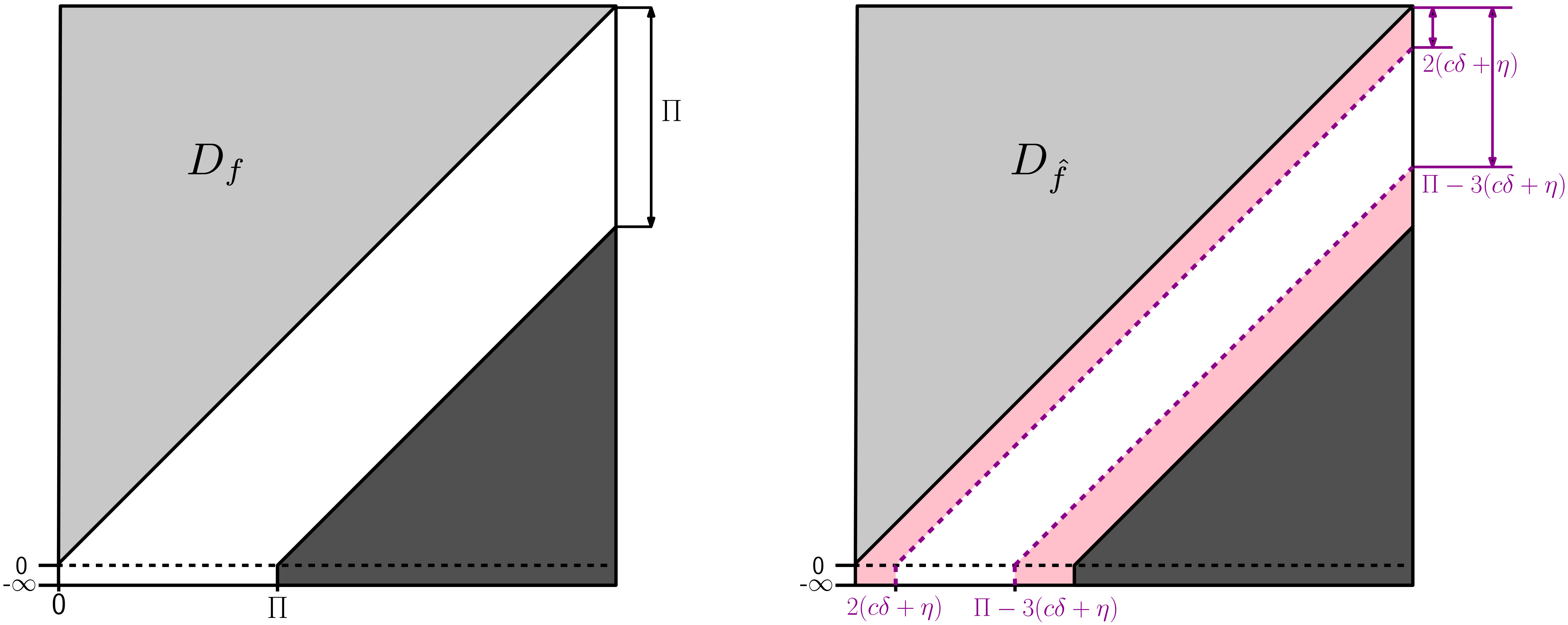
Hypotheses:

- $f : \mathbb{R}^d \to \mathbb{R}$ a c-Lipschitz probability density function,
- $P \subset \mathbb{R}^d$ a finite set of n points sampled i.i.d. according to f,
- $\hat{f} : P \to \mathbb{R}$ a density estimator s.t. $\eta := \max_{p \in P} |\hat{f}(p) - f(p)| < \Pi/5$,
- $G = (P, E)$ the δ-neighborhood graph for some positive $\delta < \frac{\Pi - 5\eta}{5c}$.

Note: Π is the prominence of the least prominent peak of f

Th: For any choice of τ such that $2(c\delta + \eta) < \tau < \Pi - 3(c\delta + \eta)$, the number of clusters computed by the algorithm is equal to the number of peaks of f with probability at least $1 - e^{-\Omega(n)}$. *(the Ω notation hides factors depending on c, δ)*

Estimating the correct number of clusters



Th: For any choice of τ such that $2(c\delta+\eta) < \tau < \Pi - 3(c\delta+\eta)$, the number of clusters computed by the algorithm is equal to the number of peaks of f with probability at least $1-e^{-\Omega(n)}$. *(the Ω notation hides factors depending on c, δ)*