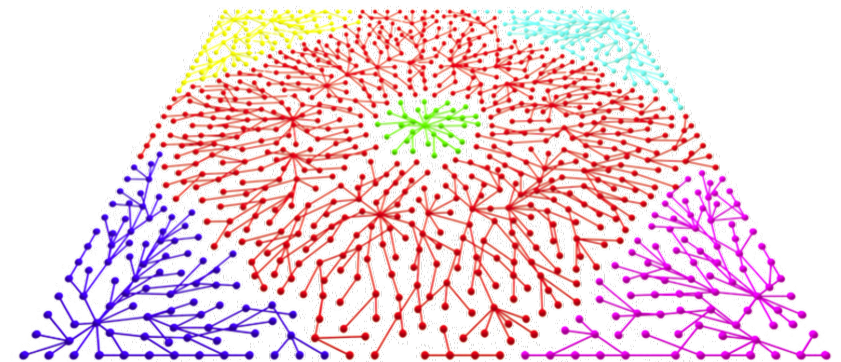
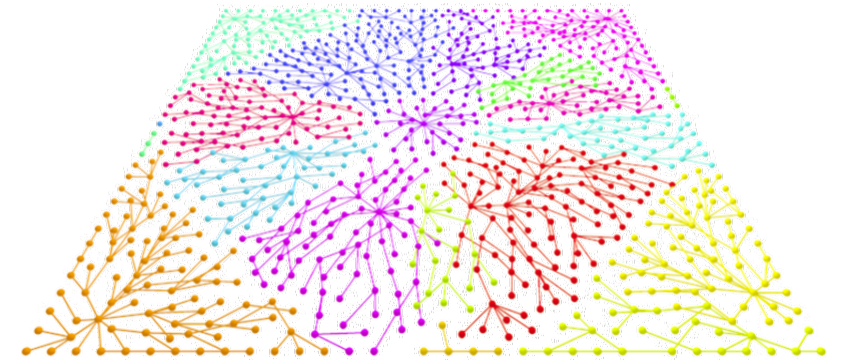


# CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas  
Computer Science Department  
Stanford University

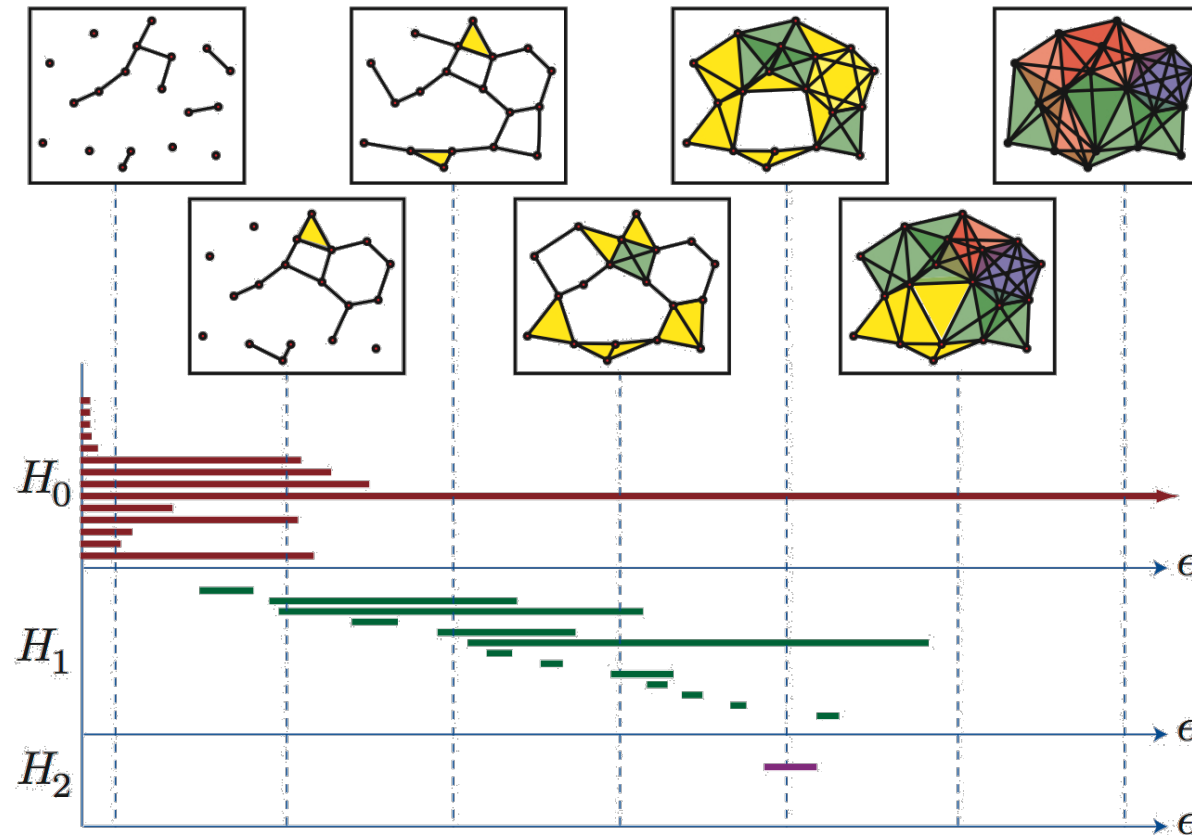


Lecture 9  
25 April 2022

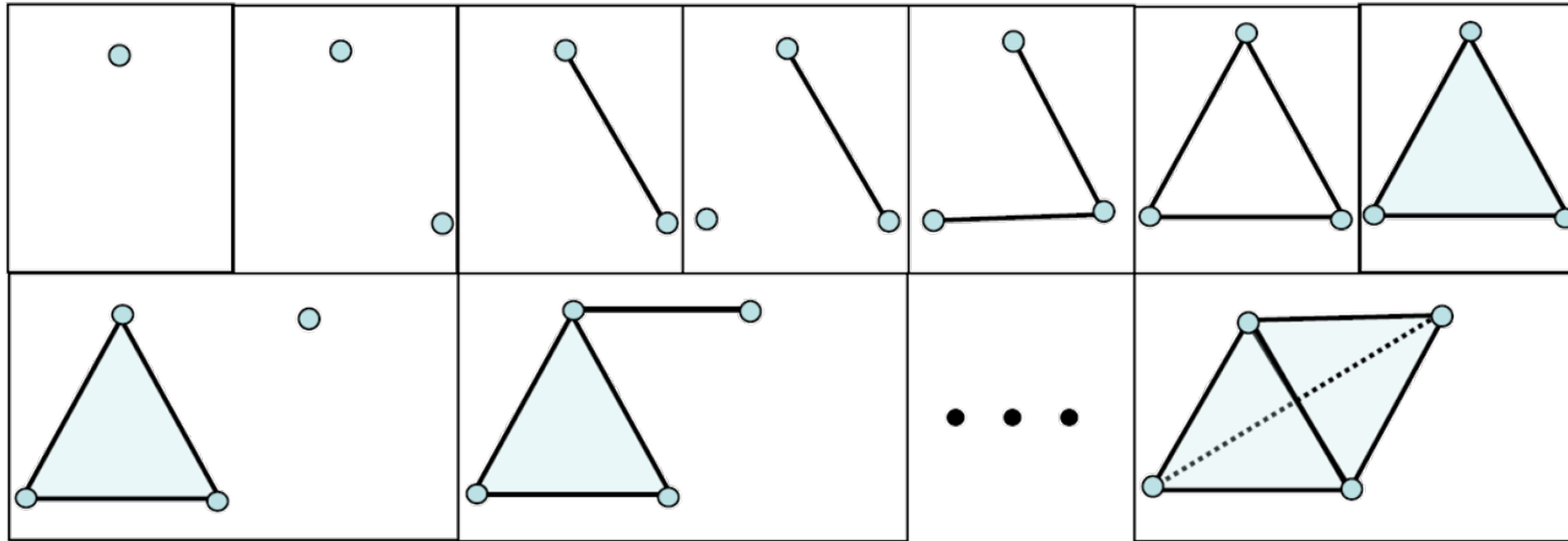


# Last Time: Persistent Homology

# Persistent Homology



# Filtrations



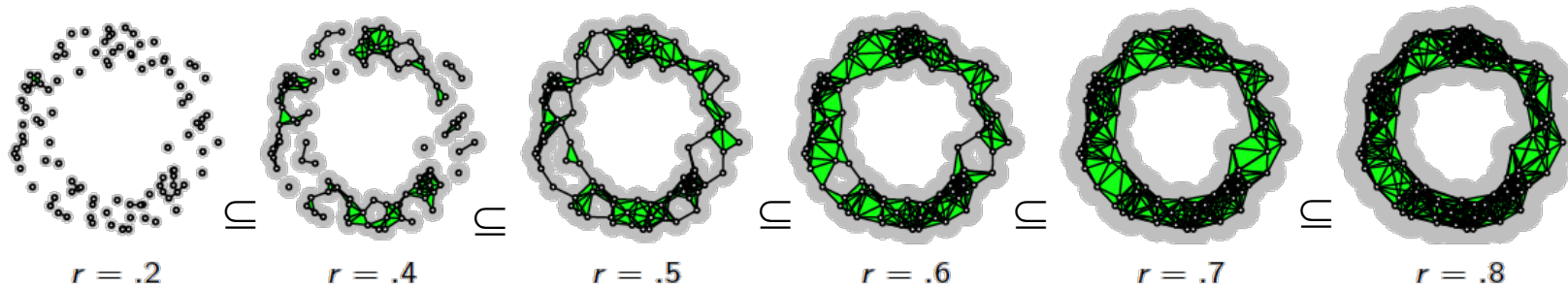
A **filtration** of a (finite) simplicial complex  $K$  is a sequence of subcomplexes such that

i)  $\emptyset = K^0 \subset K^1 \subset \dots \subset K^m = K$ ,

ii)  $K^{i+1} = K^i \cup \sigma^{i+1}$  where  $\sigma^{i+1}$  is a simplex of  $K$ .

Sub-simplices of a simplex must be added before the simplex!

# Persistent Homology is Functorial Homology

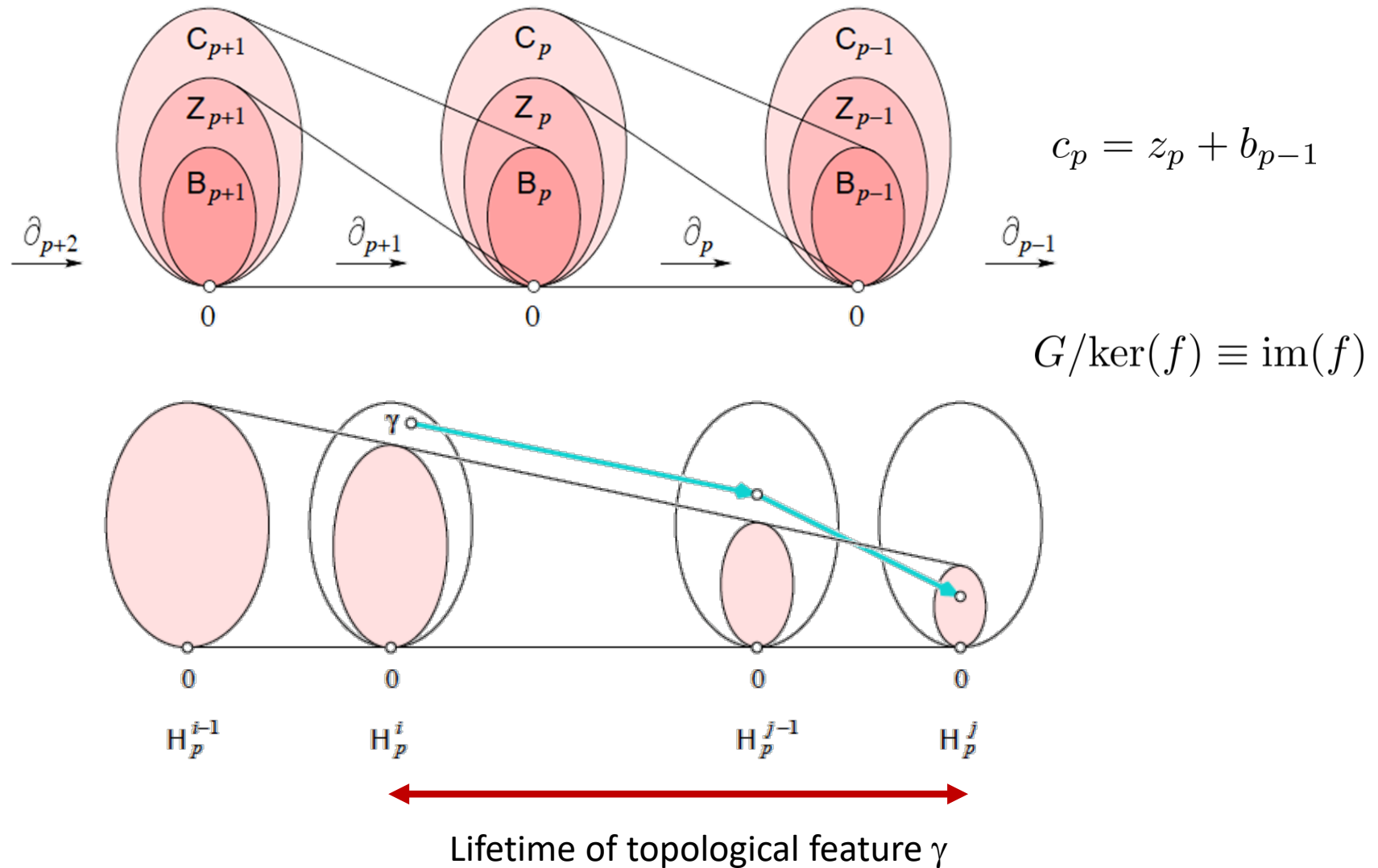


$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Homology of the entire filtration

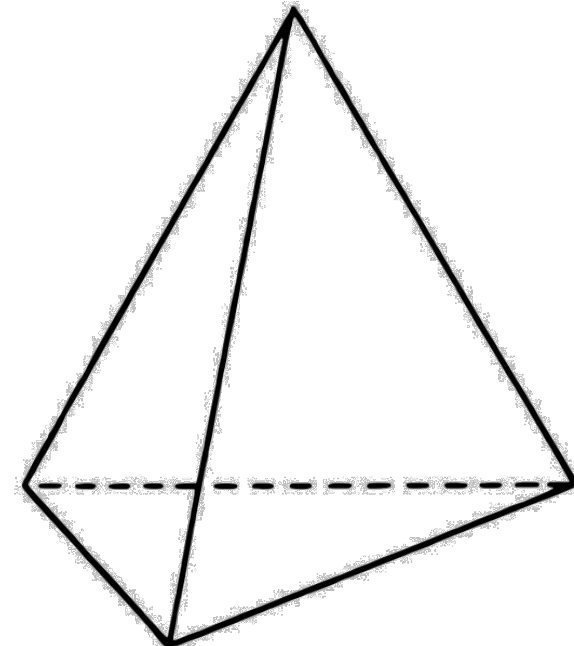
Homomorphisms at the homology level allow us to track homology classes – i.e., topological features

# Track Lifetimes of Homology Classes



# Betti Numbers $\beta_i$

- Ranks/dimensions of the free part of homology groups  $H_i$
- $\beta_0$  counts the number of connected components
- $\beta_1$  counts the number of independent loops
- $\beta_2$  counts the number of independent voids
- ...



Topology is fundamentally a tool for classification

# Reduction Algorithm for Homology\*

\*Cohen-Steiner, Edelsbrunner, Morozov - Vines and Vineyards by Updating Persistence in Linear Time

To get  $H_p$ , we need: (1) a basis for  $\ker \partial_p$ , and (2) a sub-basis for  $\text{im } \partial_{p+1}$ .

	$ab$	$ca$	$bc$	$cd$	$de$	$eb$	$abc$
$a$	1	1	0	0	0	0	0
$b$	1	0	1	0	0	1	0
$c$	0	1	1	1	0	0	0
$d$	0	0	0	1	1	0	0
$e$	0	0	0	0	1	1	0
$ab$	0	0	0	0	0	0	1
$ca$	0	0	0	0	0	0	1
$bc$	0	0	0	0	0	0	1
$cd$	0	0	0	0	0	0	0
$de$	0	0	0	0	0	0	0
$eb$	0	0	0	0	0	0	0

$R$

Notation:  $\text{low}_R(j)$  = row index of bottom 1 in column  $j$  of matrix  $R$

Algorithm:

Initialize boundary information in incidence matrix  $D$

Initialize  $R := D$

For column  $j = 1, 2, \dots, n$

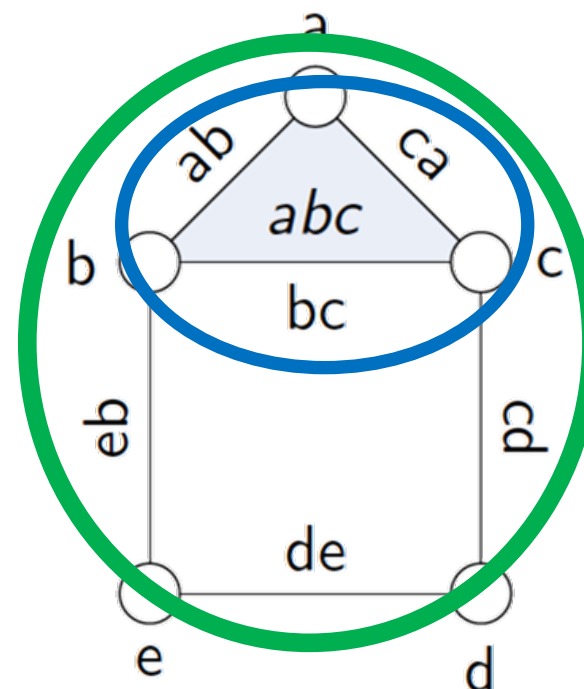
while  $\exists j' < j, \text{low}_R(j') = \text{low}_R(j)$  do  
add column  $j'$  to column  $j$

# Reduction Algorithm for Homology

$$H_0 \cong \mathbb{F}[a]$$

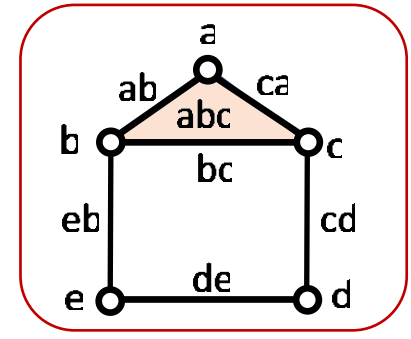
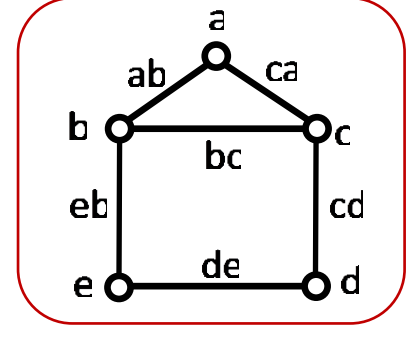
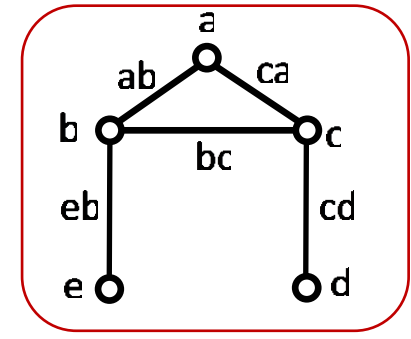
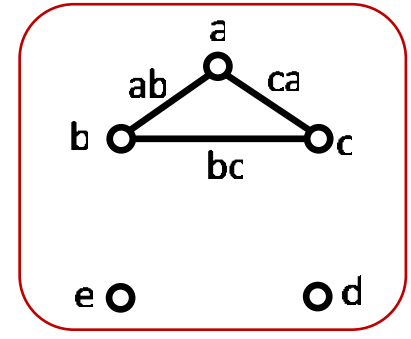
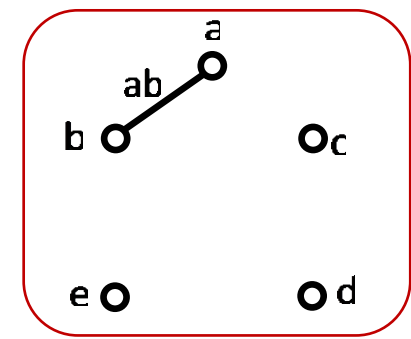
$$H_1 \cong \mathbb{F}[eb + de + cd + ca + ab]$$

	1	2	3	4	5			8		11		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>ab</i>	<i>ca</i>	<i>bc + ca + ab</i>	<i>cd</i>	<i>de</i>	<i>eb + de + cd + ca + ab</i>	<i>abc</i>
1	<i>a</i>	0	0	0	0	1	1	0	0	0	0	0
2	<i>b</i>	0	0	0	0	1	0	0	0	0	0	0
3	<i>c</i>	0	0	0	0	0	1	0	1	0	0	0
4	<i>d</i>	0	0	0	0	0	0	0	1	1	0	0
5	<i>e</i>	0	0	0	0	0	0	0	0	1	0	0
	<i>ab</i>	0	0	0	0	0	0	0	0	0	0	0
	<i>ca</i>	0	0	0	0	0	0	0	0	0	0	0
8	<i>bc + ca + ab</i>	0	0	0	0	0	0	0	0	0	0	1
	<i>cd</i>	0	0	0	0	0	0	0	0	0	0	0
	<i>de</i>	0	0	0	0	0	0	0	0	0	0	0
	<i>eb + de + cd</i>	0	0	0	0	0	0	0	0	0	0	0
11	<i>+ ca + ab</i>	0	0	0	0	0	0	0	0	0	0	0
	<i>abc</i>	0	0	0	0	0	0	0	0	0	0	0



# Reduction Algorithm for Persistent Homology

	1	2	3	4	5	$ab$	$ca$	8	$bc + ca + ab$	$cd$	$eb$	11	$de + eb + cd + ca + ab$	$abc$
	$a$	$b$	$c$	$d$	$e$									
1	$a$	0	0	0	0	1	1	0	0	0	0	0	0	0
2	$b$	0	0	0	0	1	0	0	0	1	0	0	0	0
3	$c$	0	0	0	0	0	1	0	1	0	0	0	0	0
4	$d$	0	0	0	0	0	0	0	0	1	0	0	0	0
5	$e$	0	0	0	0	0	0	0	0	0	1	0	0	0
	$ab$	0	0	0	0	0	0	0	0	0	0	0	0	0
	$ca$	0	0	0	0	0	0	0	0	0	0	0	0	0
8	$bc + ca + ab$	0	0	0	0	0	0	0	0	0	0	0	0	1
	$cd$	0	0	0	0	0	0	0	0	0	0	0	0	0
	$eb$	0	0	0	0	0	0	0	0	0	0	0	0	0
11	$de + eb + cd + ca + ab$	0	0	0	0	0	0	0	0	0	0	0	0	0
	$abc$	0	0	0	0	0	0	0	0	0	0	0	0	0



# Back to the House Example

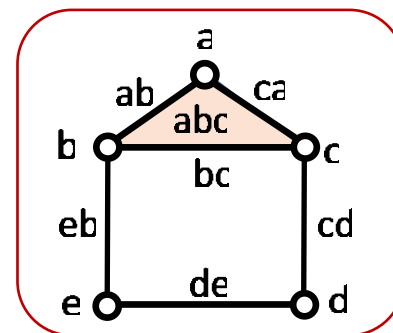
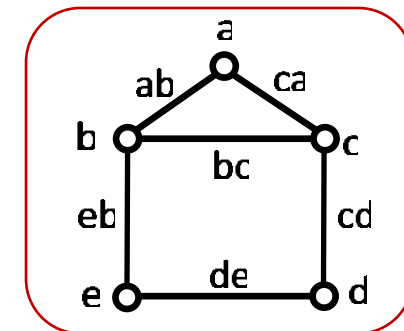
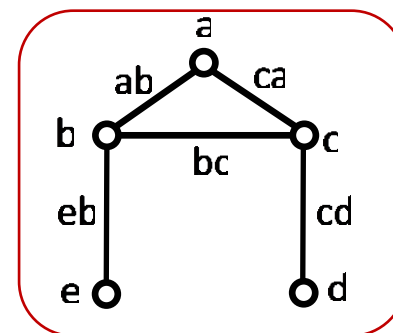
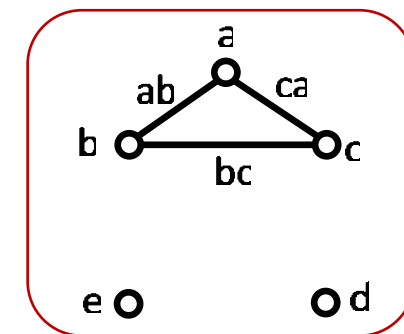
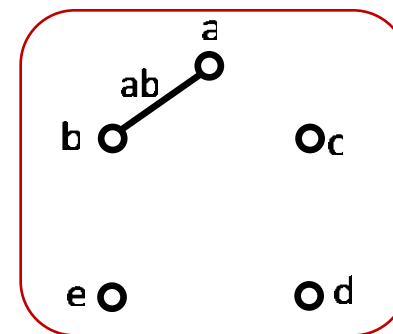
**Persistence pairs:**  $\{(2,6), (3,7), (4,9), (5,10), (8,12)\}$   
 (column indices)

**Lifetimes (dim 0):**  $\{[1,1), [1,2), [1,3), [1,3)\}$   
 (time indices)

**Lifetimes (dim 1):**  $\{[2,5)\}$   
 (time indices)

**Essential indices:**  $\{1, 11\}$   
 (column indices)

**Lifetimes:**  $\{[1, \infty)$  (connected component),  $[4, \infty)$  (loop) $\}$   
 (time indices)



# Simplices are “Creators” or “Destroyers”

- ◆ Every simplex either creates a homology class or destroys one
- ◆ Specifically, a  $k$ -simplex either creates a  $k$ -dimensional feature or destroys a  $(k - 1)$ -dimensional feature

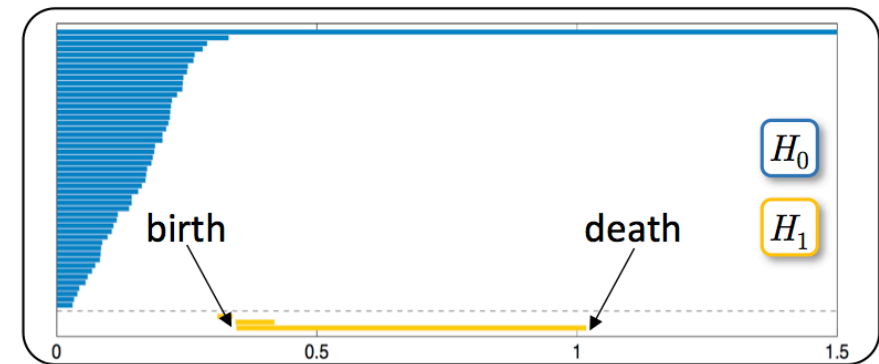
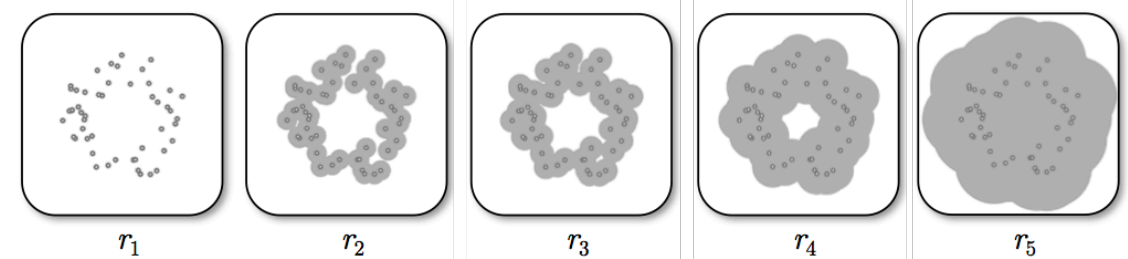
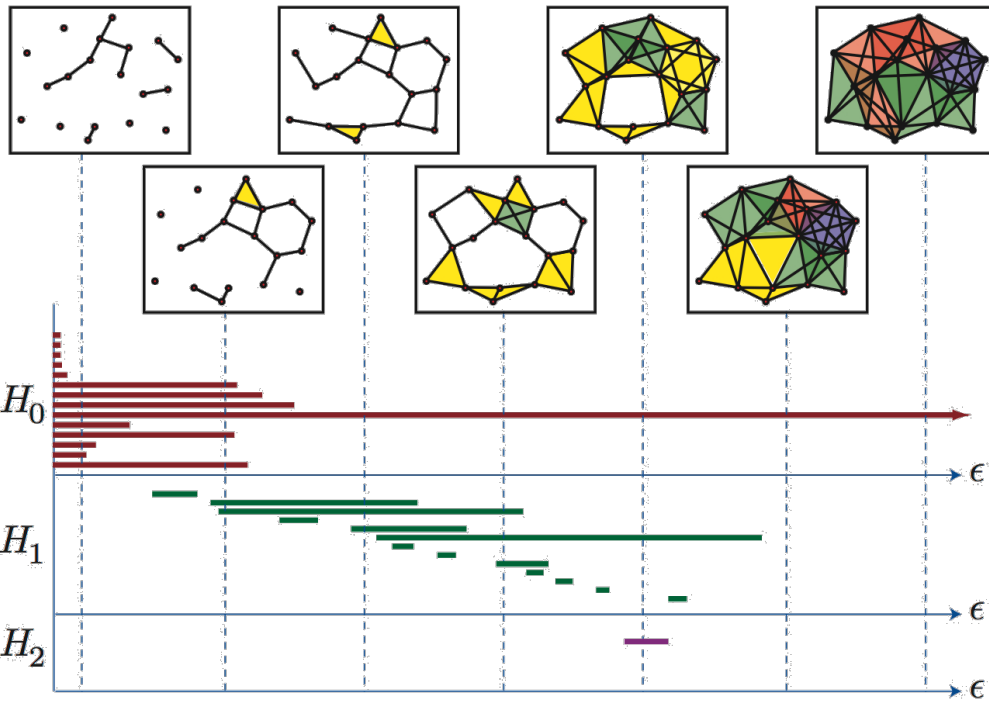
# Pipeline

Dataset:  
- Euclidean point cloud data  
- Metric spaces  
- Triangulated shapes  
- Graphs, matrices...

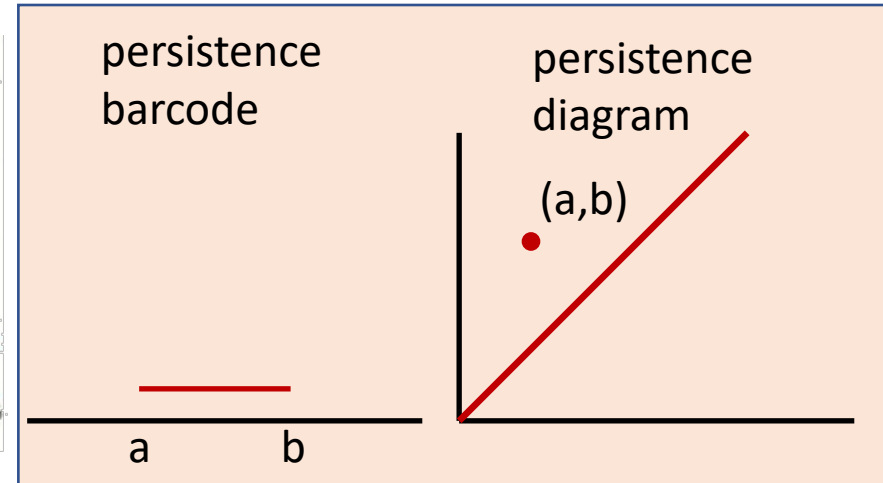
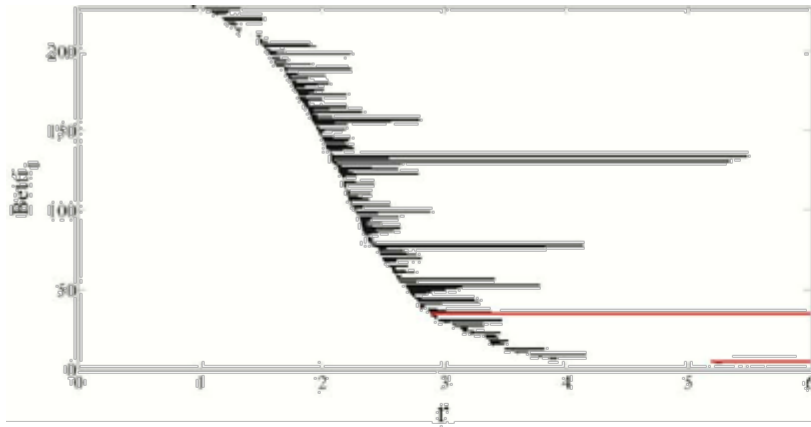
Filtered simplicial complex  
• Possible to have both inclusions and deletions (dynamic data)

Vector spaces with linear maps: a **persistent vector space**

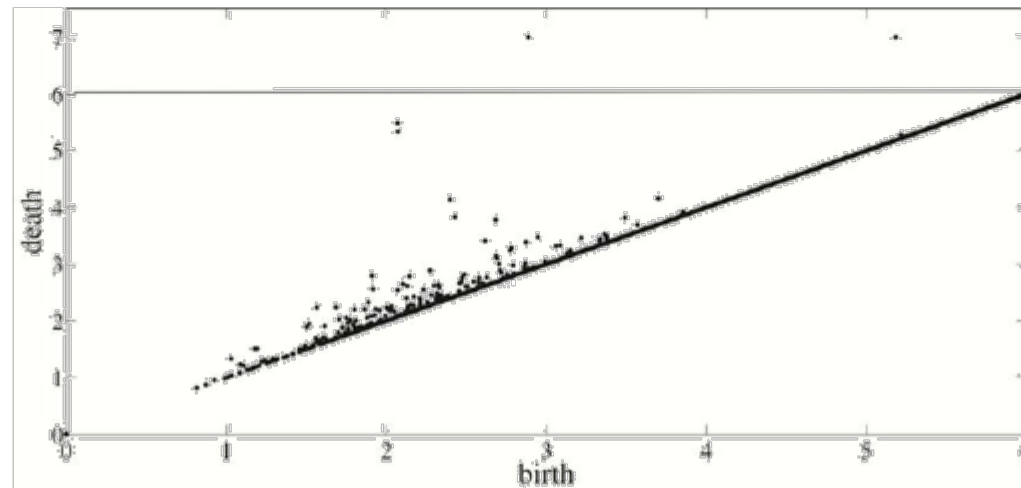
Visual summary: a **persistence diagram** or **persistence barcode**  
• Postprocessing available for ML-friendly output



# Another View: Persistence Diagrams



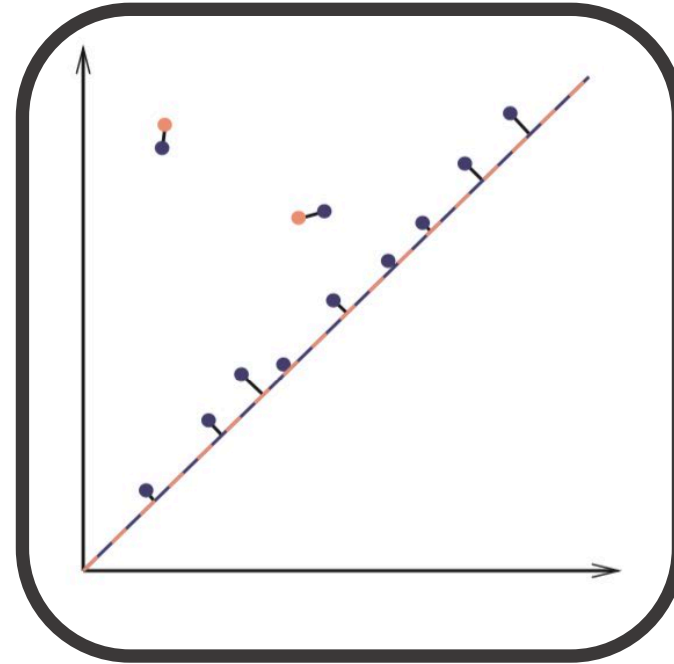
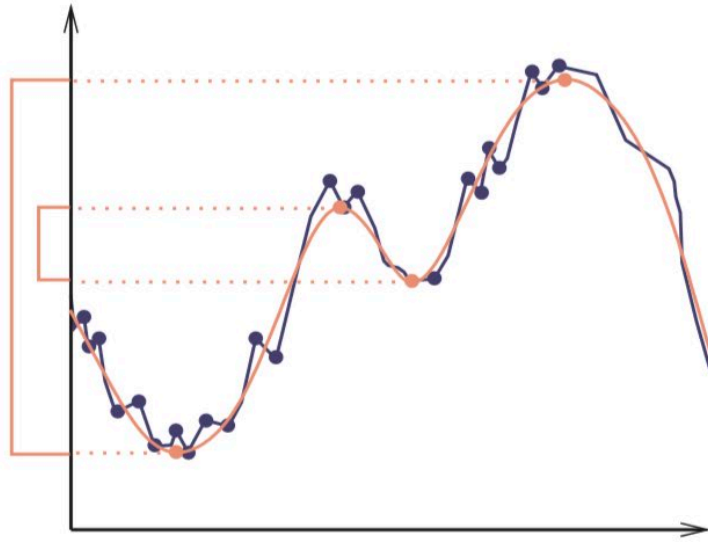
long barcodes =  
points away from  
the diagonal =  
robust features



Short barcodes =  
points near  
the diagonal =  
noise

Map 1-D intervals to points in 2-D

# The Bottleneck Distance



$A, B$  two persistence diagrams.

$$d_B(A, B) := \inf \left\{ \sup_{x \in A \cup \Delta} \|x - \phi(x)\|_\infty : \phi : A \cup \Delta \rightarrow B \cup \Delta \text{ a bijection} \right\}$$

# Stability of Persistent Homology

- ◆ Persistent homology outputs are robust to small perturbations of data

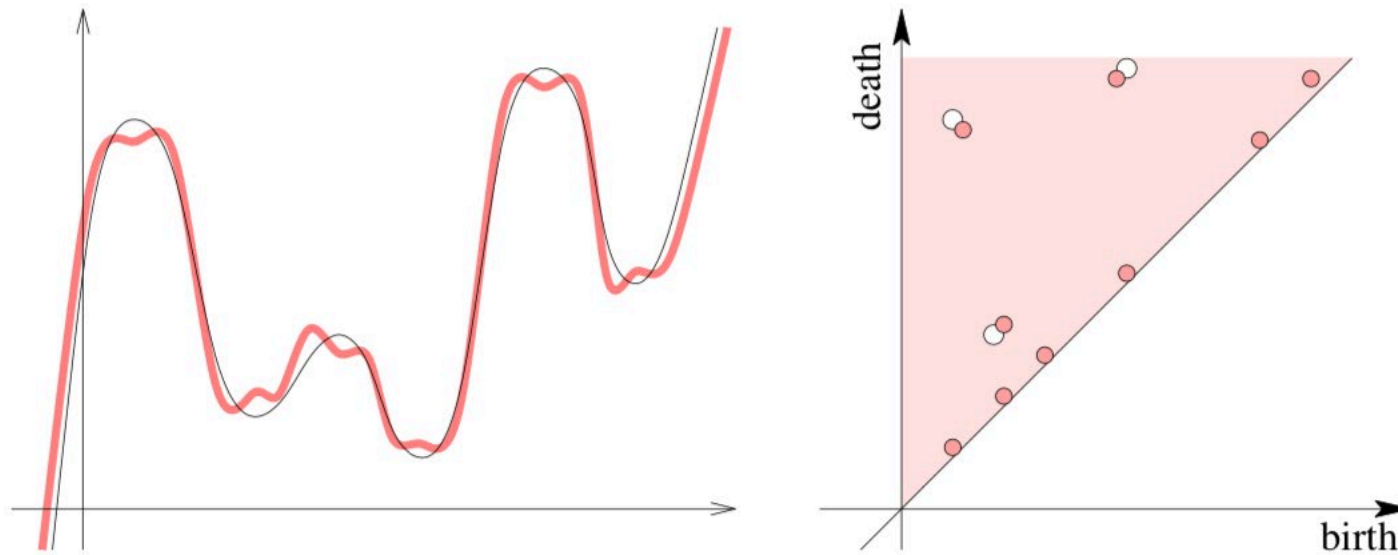


Figure from Edelsbrunner, Harer,  
*Computational Topology*

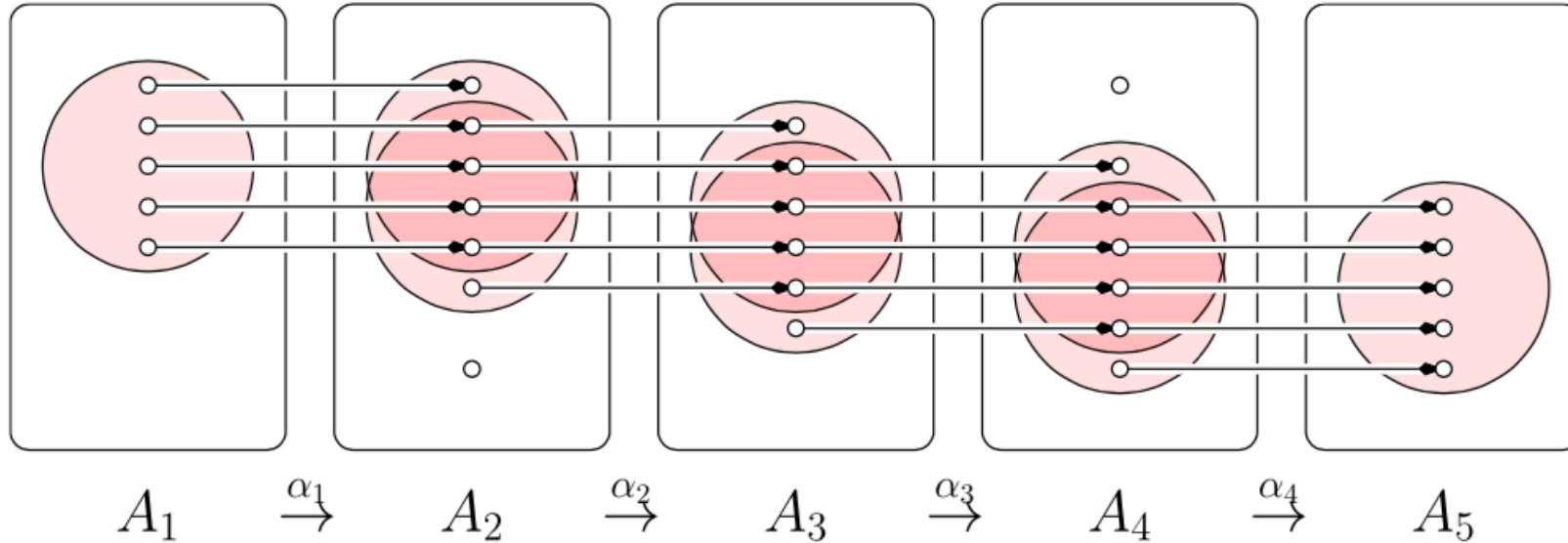
- ◆ Theorem (Cohen-Steiner, Edelsbrunner, Harer):  $\mathcal{M}$  a triangulable space,  $f, g: \mathcal{M} \rightarrow \mathbb{R}$  “nice”. Then,  
$$d_B(D_k(f), D_k(g)) \leq \|f - g\|_\infty.$$

Here  $D_k$  is the persistence diagram in dimension  $k$ ,  $\|\cdot\|_\infty$  is the max-norm, and  $d_B$  is the bottleneck distance

# Mathematical Theory

# A Deeper Look at the Algorithm

Figure from Edelsbrunner, Jablonski, Mrozek, *The persistent homology of a self-map*

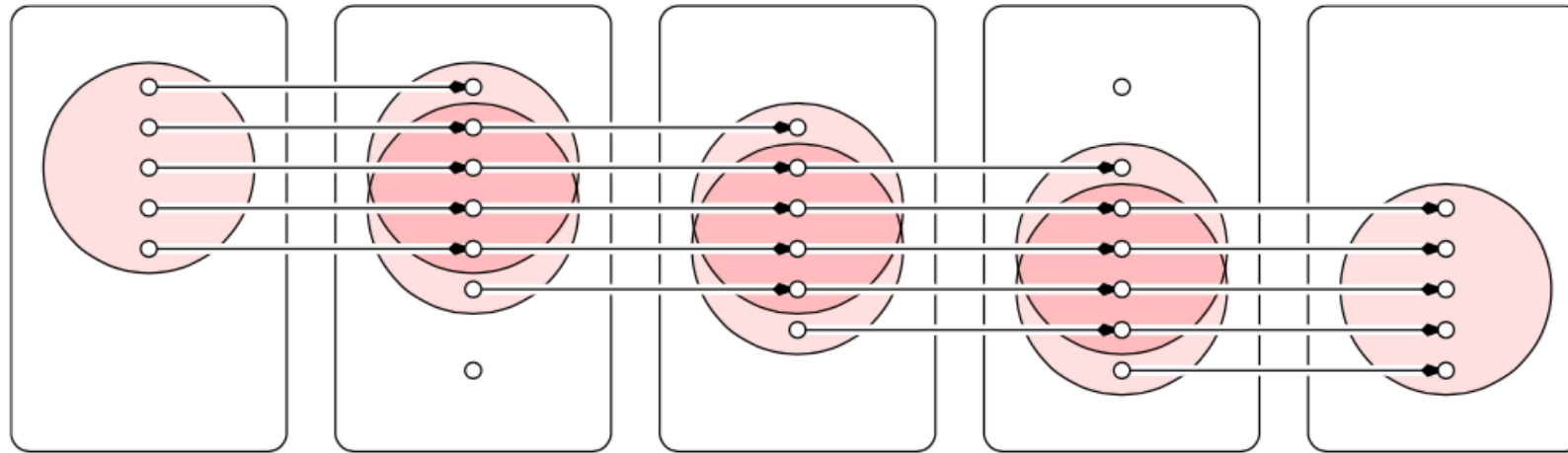


- ◆ Left-to-right ordering of chain complex basis vectors + left-to-right column operations gets us to compatible bases for homology
- ◆ “Existence and uniqueness” comes from commutative algebra



# PH as Applied Representation Theory

Carlsson, de Silva, *Zigzag Persistence*



Theorem (Gabriel '72): Any  $\mathbb{V}$  as above can be written  $\mathbb{V} = \bigoplus_i \mathbb{I}(b_i, d_i)$ , where

$$\mathbb{I}(b_i, d_i) = 0 \rightarrow 0 \rightarrow \cdots 0 \rightarrow \mathbb{F} \rightarrow \mathbb{F} \rightarrow \mathbb{F} \rightarrow \cdots \rightarrow \mathbb{F} \rightarrow 0 \rightarrow 0 \rightarrow 0$$

↑
↑
↑
↑

Index 1
Index  $b_i$ 
Index  $d_i$ 
Index  $n$

Theorem (Krull-Remak-Schmidt): Such a decomposition is unique up to relabeling.

◆ Gabriel's theorem does not need all the arrows to point in the same direction!

- Standard persistence:

$$V_1 \rightarrow V_2 \rightarrow \cdots \rightarrow V_n$$

- Zigzag persistence:

$$V_1 \xleftrightarrow{p_1} V_2 \xleftrightarrow{p_2} \cdots \xleftrightarrow{p_{n-1}} V_n.$$

- Each  $p_i$  can be forward or backward

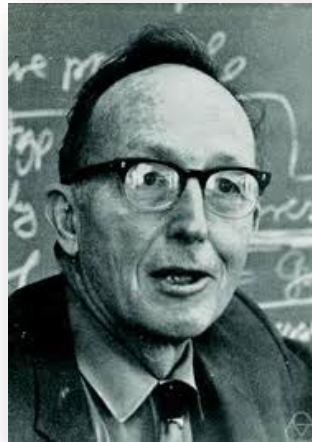
◆ Application: inclusion not needed, can compare/correlate features across incomparable spaces

- E.g. Union sequence:  $X_1 \rightarrow X_1 \cup X_2 \leftarrow X_2 \rightarrow \cdots \leftarrow X_n$

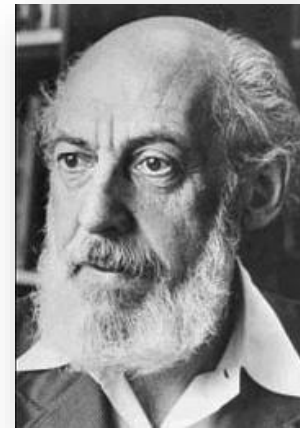
# Homological Algebra: Functors and Categories



Henri Cartan  
1904-2008



Saunders MacLane  
1909-2005



Samuel Eilenberg  
1913 -1998

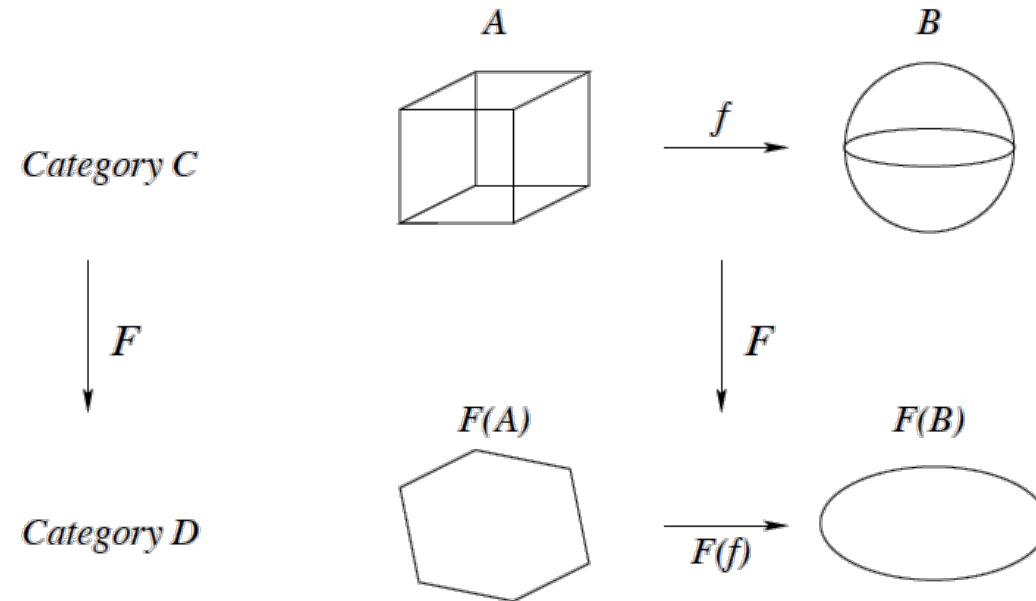
# Categories

- A collection  $\text{Ob}(\mathcal{C})$  of **objects**
- Sets  $\text{Mor}(X, Y)$  of **morphisms** for each pair  $X, Y \in \text{Ob}(\mathcal{C})$
- An identity morphism  $1 = 1_X \in \text{Mor}(X, X)$  for each  $X$ .
- a composition of morphisms function
  - $\circ : \text{Mor}(X, Y) \times \text{Mor}(Y, Z) \rightarrow \text{Mor}(X, Z)$  for each triple  $X, Y, Z \in \text{Ob}(\mathcal{C})$ , satisfying  $f \circ 1 = 1 \circ f = f$ , and  $(f \circ g) \circ h = f \circ (g \circ h)$ .
- A **category**  $\mathcal{C}$

# Example Categories

category	morphisms
sets	arbitrary functions
groups	homomorphisms
topological spaces	continuous maps
topological spaces	homotopy classes of maps

# Functors



- $X \in \mathcal{C}, F(X) \in \mathcal{D},$
- $f \in \text{Mor}(X, Y), F(f) \in \text{Mor}(F(X), F(Y))$
- $F(1) = 1$  and  $F(f \circ g) = F(f) \circ F(g)$
- $F$  is a **(covariant) functor**

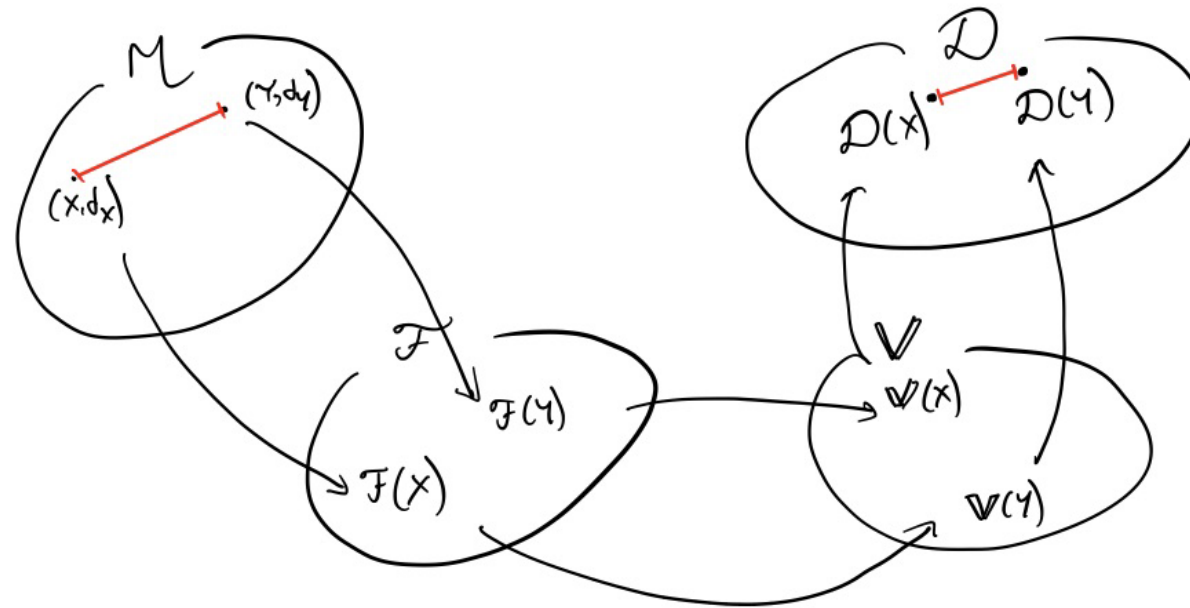
# Functoriality, Commutative Diagrams

transformation of input  $\Rightarrow$  transformation of output  
Specifically, this is a commutative diagram:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ H_* \downarrow & & \downarrow H_* \\ H_*(X) & \xrightarrow{H_*(f)} & H_*(Y) \end{array}$$

**Moral:** Invariants are not artifacts of arbitrary choices!

# Gromov-Hausdorff Stability



$\mathcal{M}$  = finite metric spaces  
 $\mathcal{F}$  = filtered simplicial complexes  
 $\mathbb{V}$  = persistent vector spaces  
 $\mathbb{D}$  = persistence diagrams

- Theorem (Chazal, Cohen-Steiner, Guibas, Mémoli, Oudot):  $(X, d_X), (Y, d_Y)$  finite metric spaces. Then,

$$d_B(D_k(VR(X)), D_k(VR(Y))) \leq 2d_{GH}(X, Y).$$

$O(n^3)$  with Hungarian algorithm,

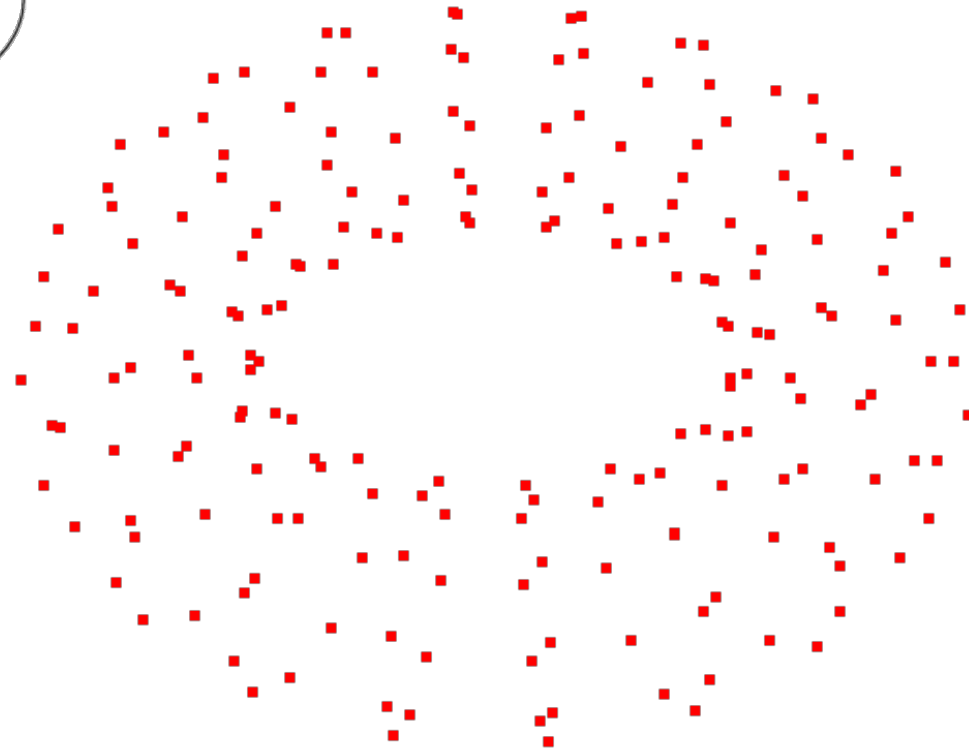
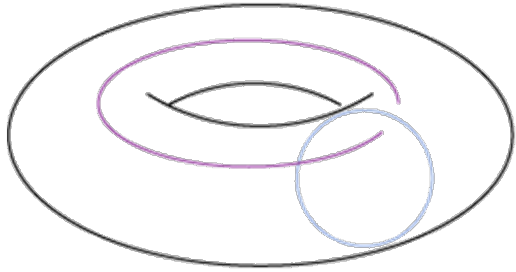
$O(n^{3/2})$  state-of-the-art (Kerber, Morozov, Nigmatov 2017)

NP-hard

# Today: Examples and Applications of Persistence

# Detecting a Torus from Samples

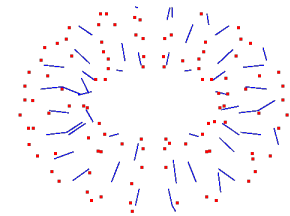
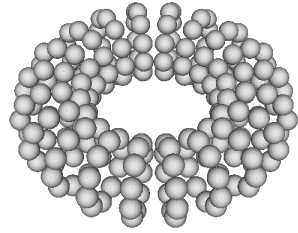
# Detecting a Torus from Samples



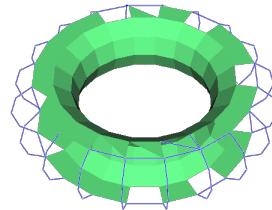
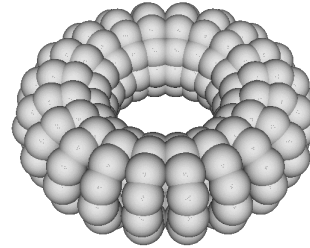
Point Cloud Data  
(PCD)

# Question of Scale: A Rips Filtration

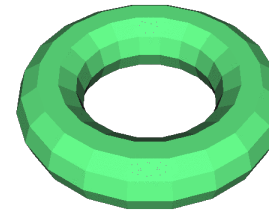
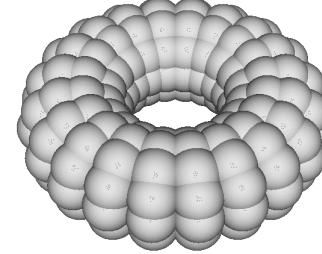
- $\beta_0$  counts the number of connected components
- $\beta_1$  counts the number of independent loops
- $\beta_2$  counts the number of independent voids



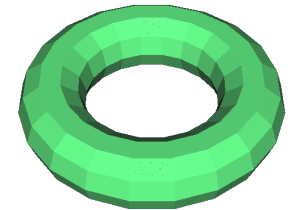
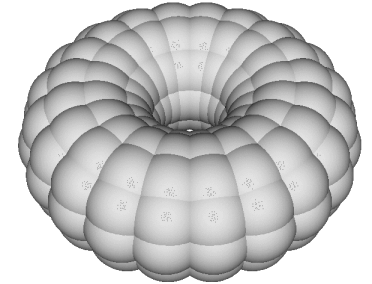
$$\begin{aligned}\beta_0 &= 150 \\ \beta_1 &= 0 \\ \beta_2 &= 0\end{aligned}$$



$$\begin{aligned}\beta_0 &= 1 \\ \beta_1 &= 37 \\ \beta_2 &= 0\end{aligned}$$



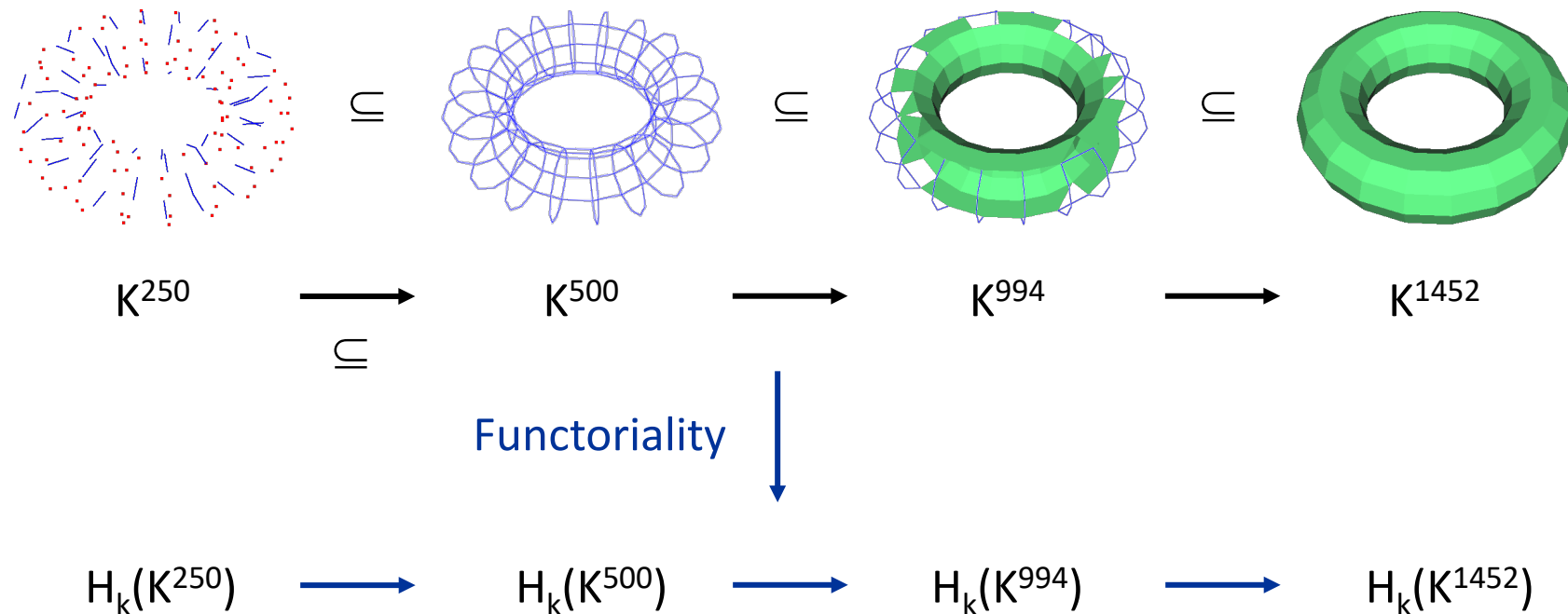
$$\begin{aligned}\beta_0 &= 1 \\ \beta_1 &= 2 \\ \beta_2 &= 1\end{aligned}$$



$$\begin{aligned}\beta_0 &= 1 \\ \beta_1 &= 1 \\ \beta_2 &= 22\end{aligned}$$

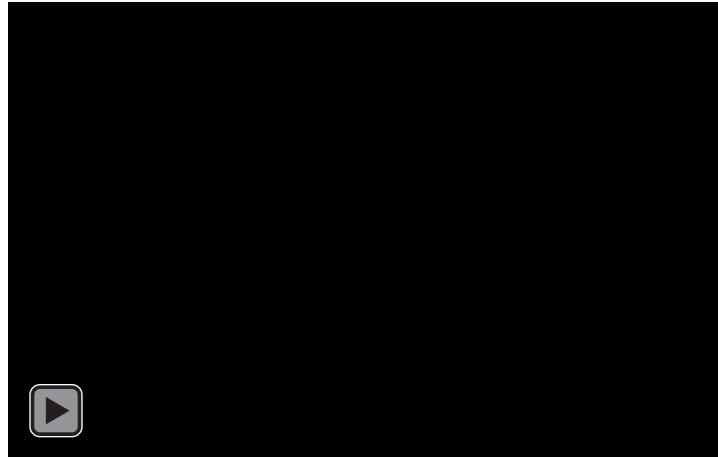


# Complex Inclusions to Homology Homomorphisms



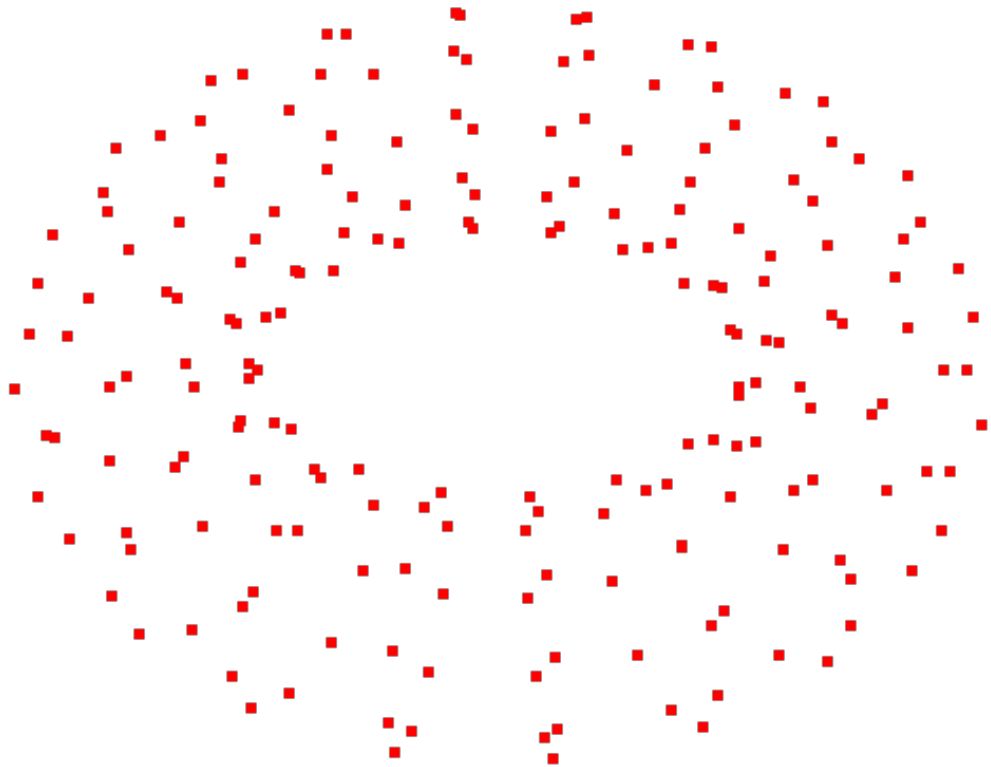
Idea: Follow homology basis elements from **birth** to **death** while maintaining **compatible bases**

# Consistent Bases Exist

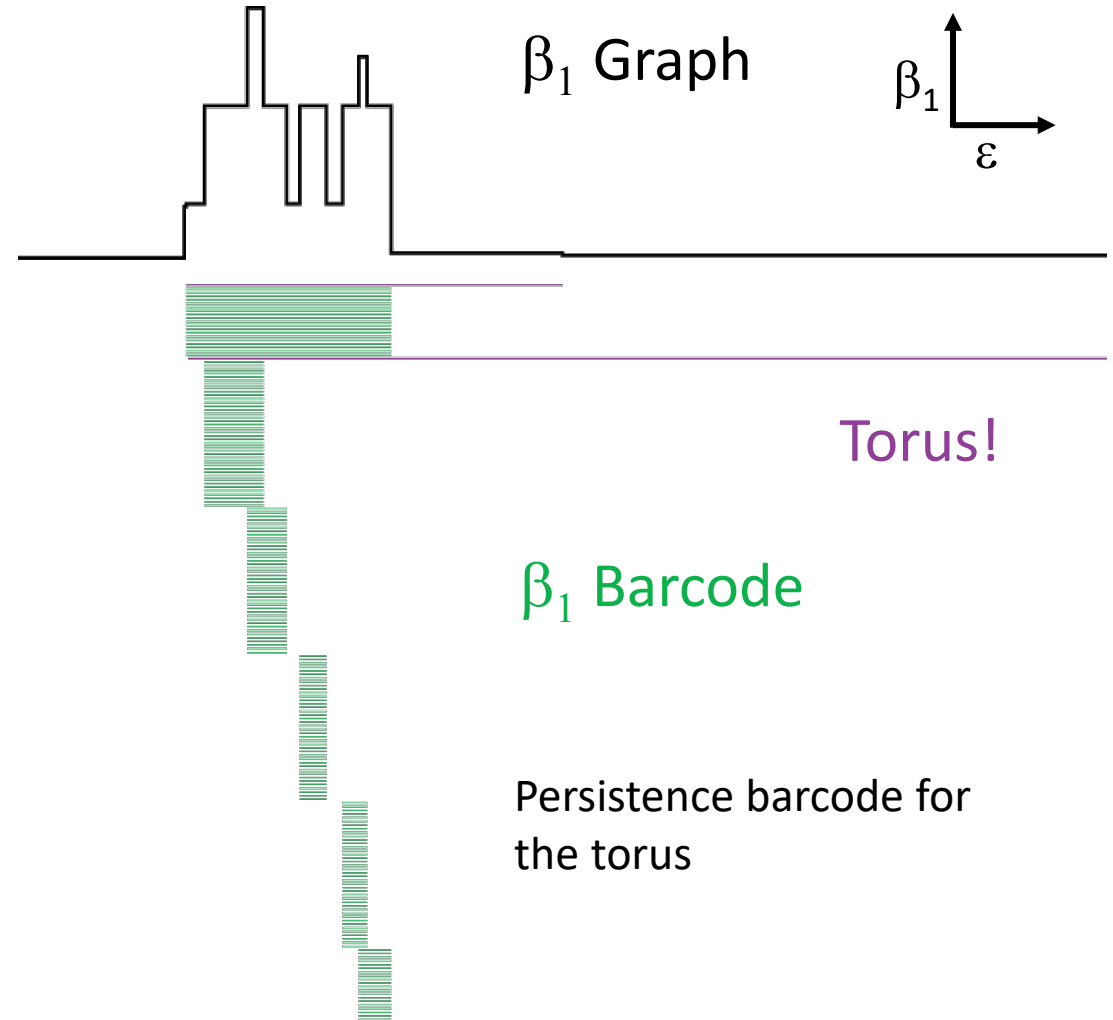


Basis elements for 1-homology

# Deconstructing the Barcode



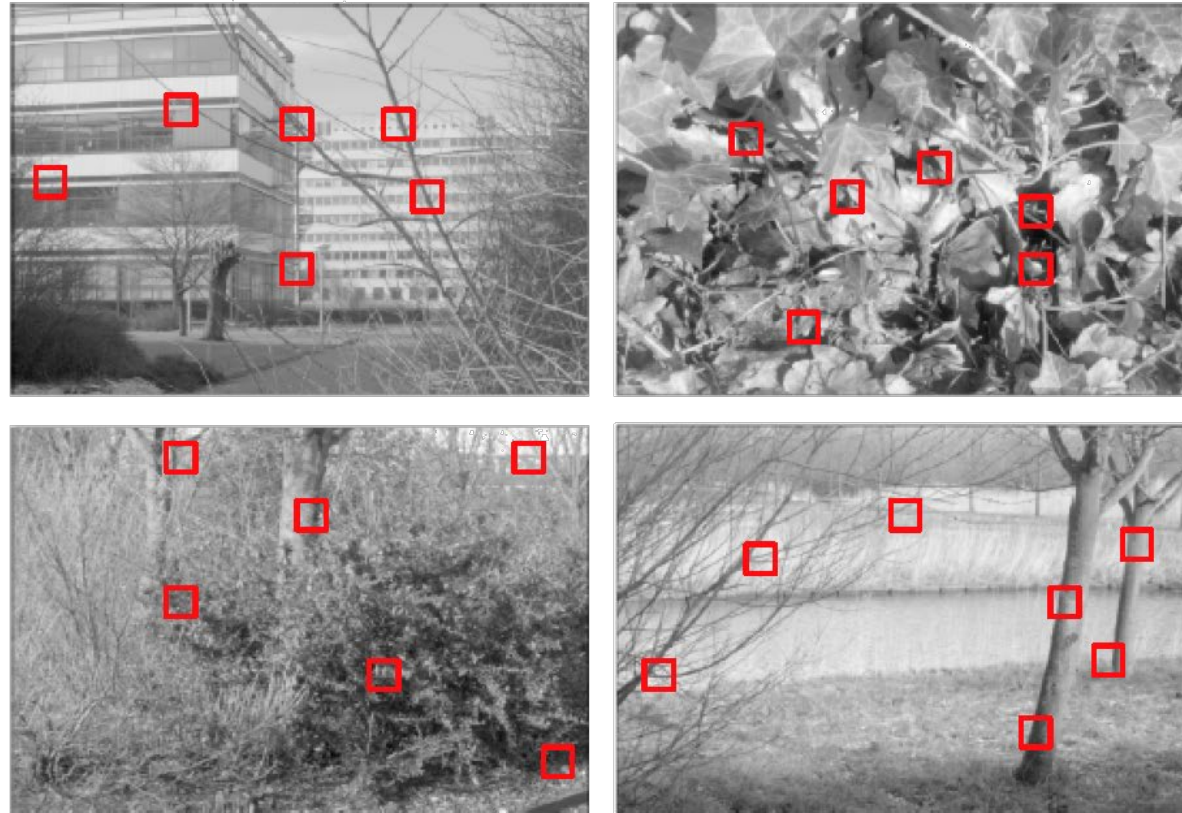
PCD



# The Space of Natural Images

# The Space of Natural Images

**Input:** 4 million data points on  $\mathbb{S}^7$ , coming from high-contrast  $3 \times 3$  image patches



(source: [Lee, Pederson, Mumford 03])

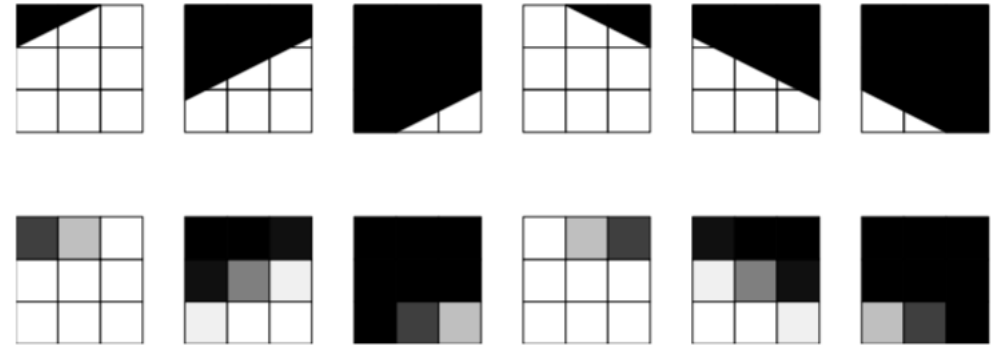
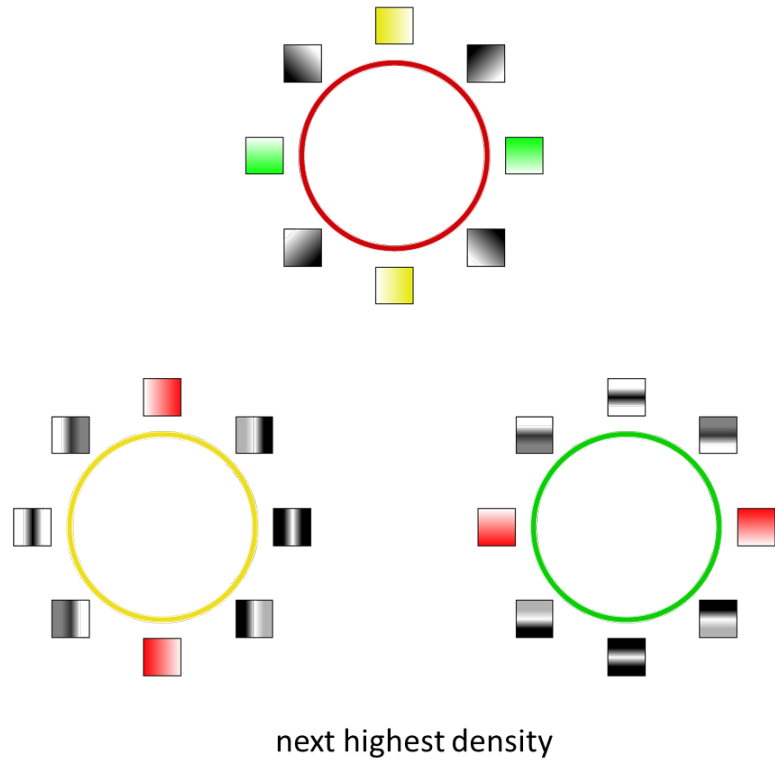
# Natural 3 x 3 Patches

- Instead of studying entire images, we consider the distribution of 3 x 3 pixel patches
- Most of these are roughly constant in natural images -- they drown out structure
- L.M.P. chose 8,500,000 patches with high contrast
- Each 3 x 3-patch is considered a vector in  $\mathbb{R}^9$
- Normalize brightness:  $\mathbb{R}^9 \rightarrow \mathbb{R}^8$
- Normalize contrast:  $\mathbb{R}^8 \rightarrow S^7$

# Natural Image Statistics

Expect to see high concentration of edge features

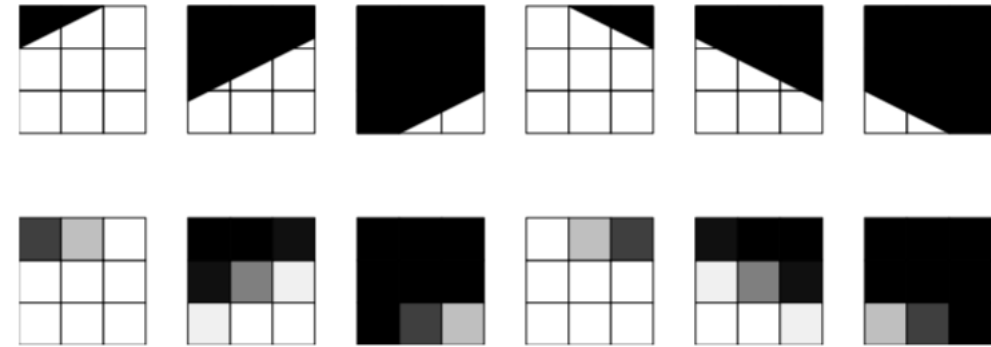
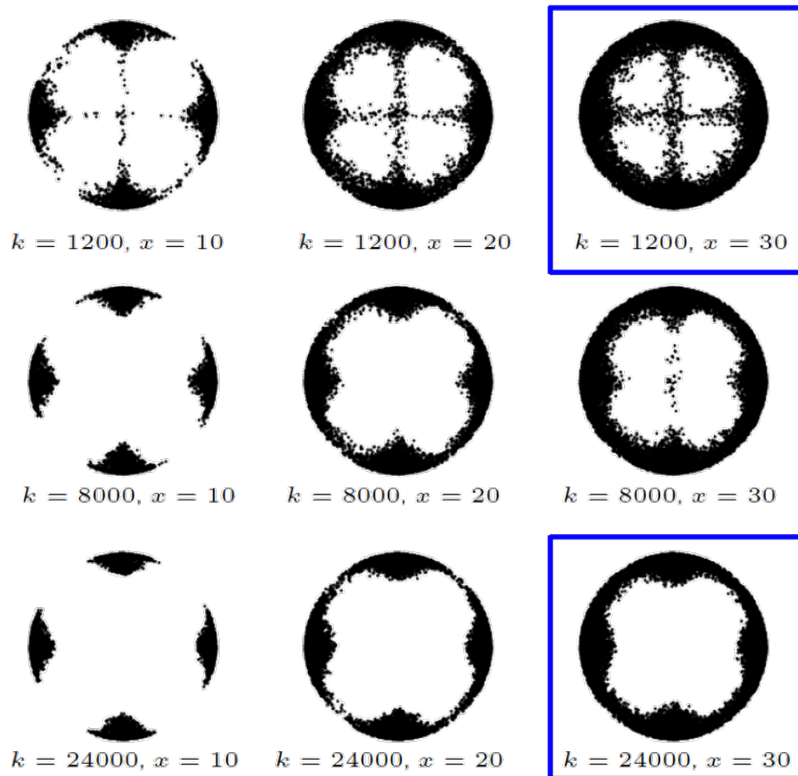
- Parametrized by angle and distance from center, expected to live on an annulus



# Natural Image Statistics

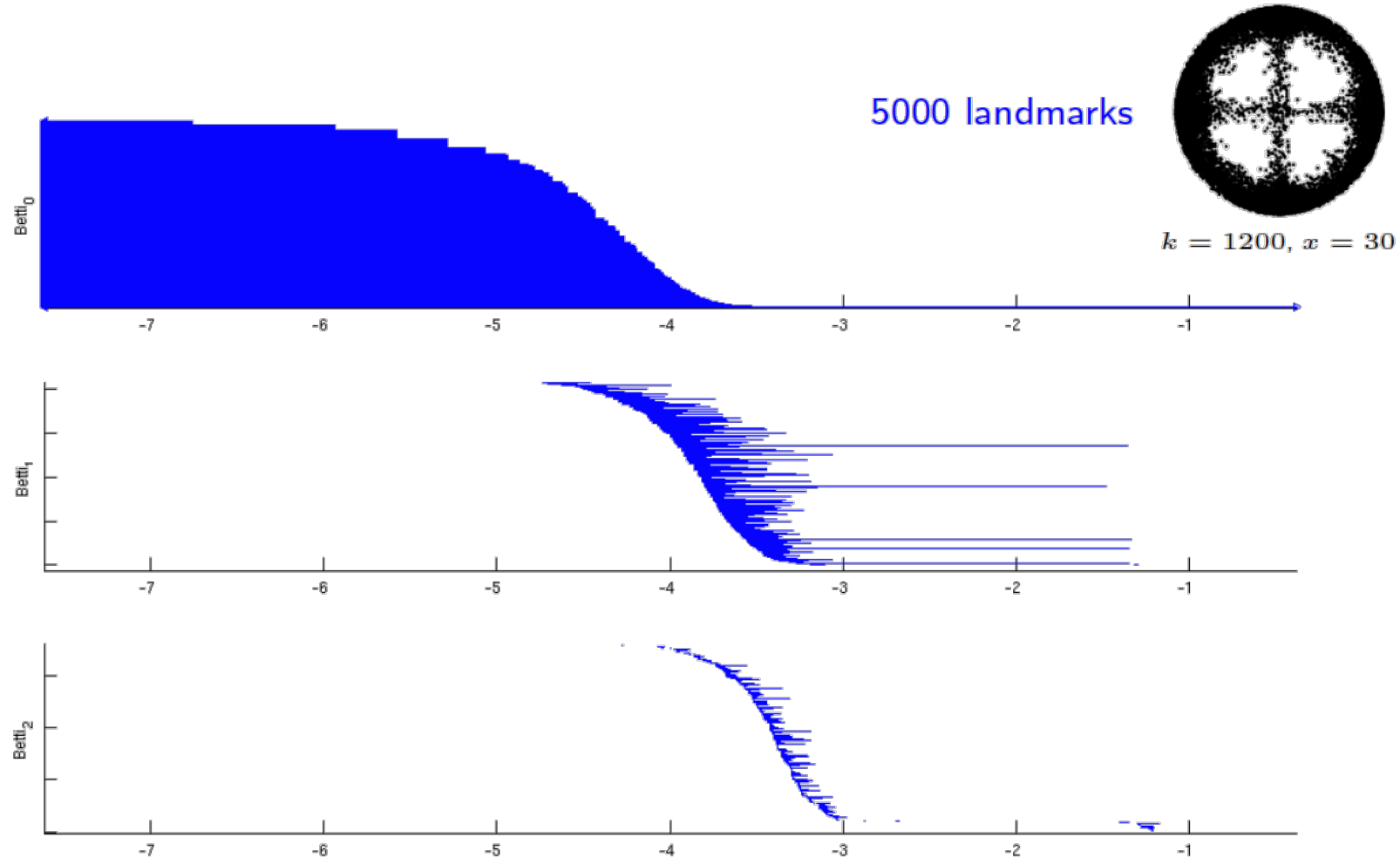
Expect to see high concentration of edge features

- Parametrized by angle and distance from center, expected to live on an annulus
- 2D view below---why does the cross appear?



# The Space of Natural Images

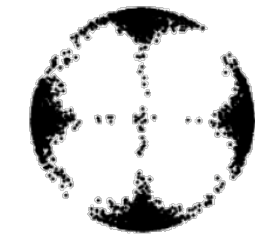
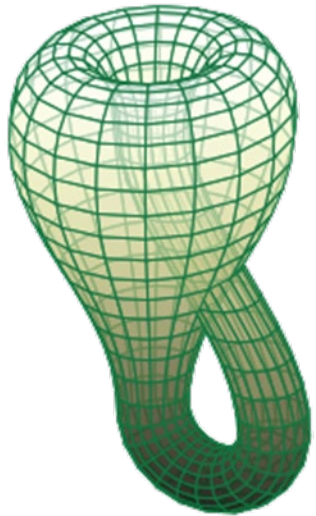
- Preprocessing:**
- select bottom  $x\%$  of data points according to  $k$ -NN distance
  - sample 5000 points uniformly at random from filtered point set



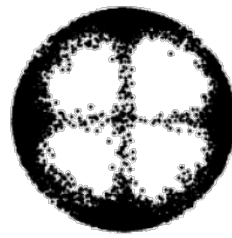
(source: [O., Sheehy 13])

# The Space of Natural Images

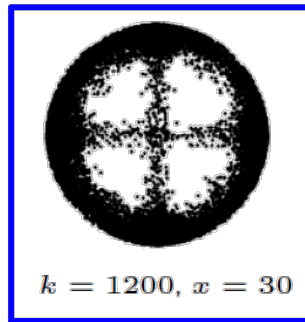
- Preprocessing:**
- select bottom  $x\%$  of data points according to  $k$ -NN distance
  - sample 5000 points uniformly at random from filtered point set



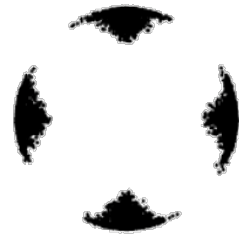
$k = 1200, x = 10$



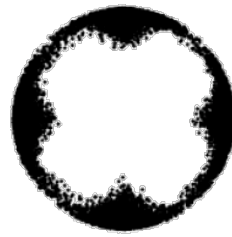
$k = 1200, x = 20$



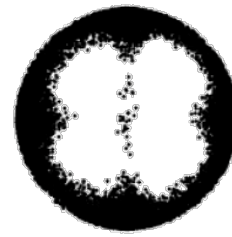
$k = 1200, x = 30$



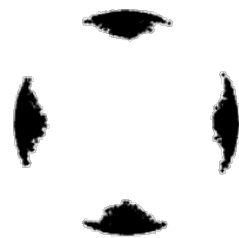
$k = 8000, x = 10$



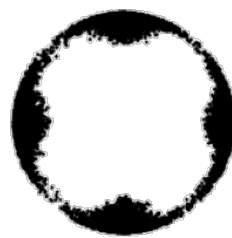
$k = 8000, x = 20$



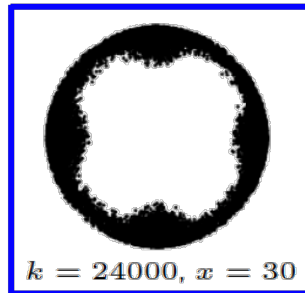
$k = 8000, x = 30$



$k = 24000, x = 10$

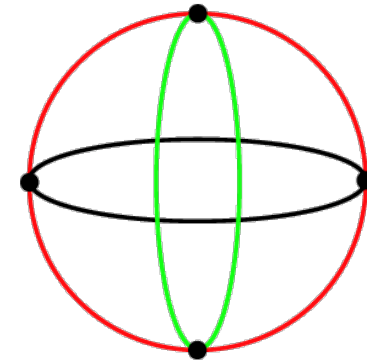


$k = 24000, x = 20$



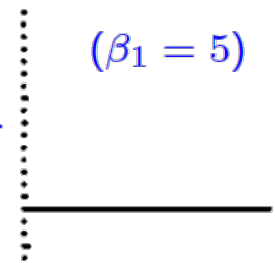
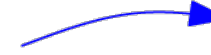
$k = 24000, x = 30$

50 landmarks



$(\beta_1 = 5)$

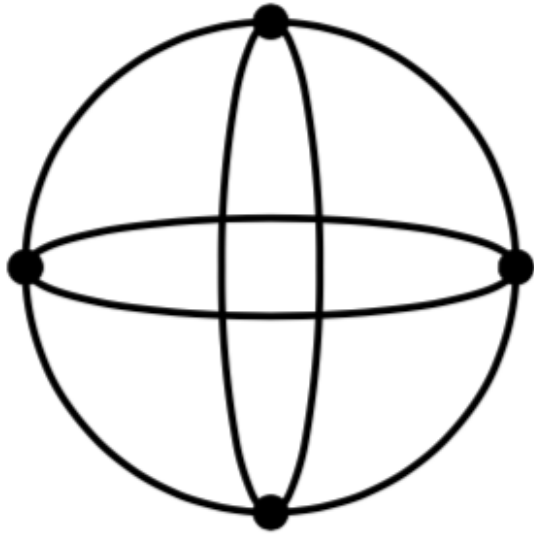
50 landmarks



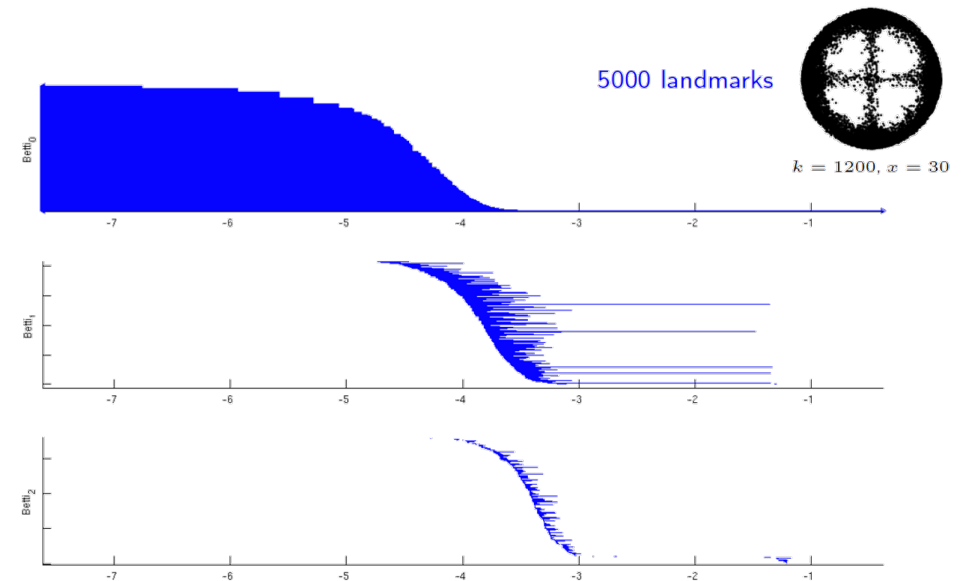
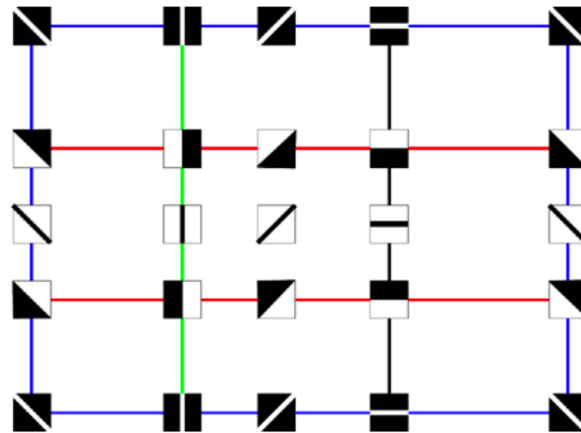
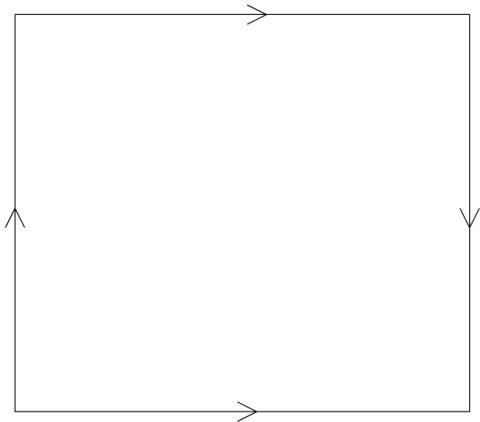
(source: [de Silva, Carlsson 04])

# Three Circle Model

Carlsson, Ishkanov, de Silva, Zomorodian, *On the local behavior of spaces of natural images*

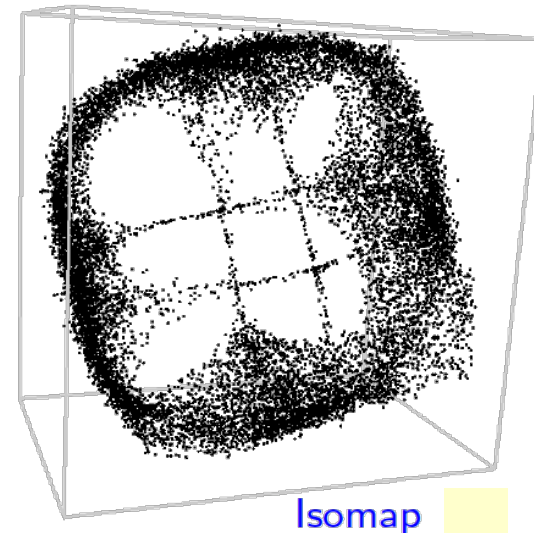
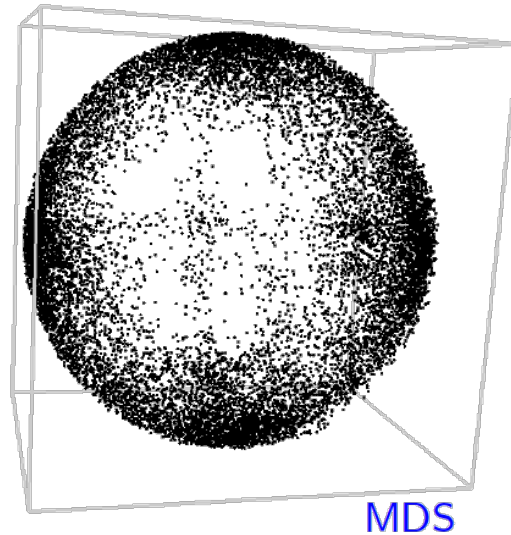
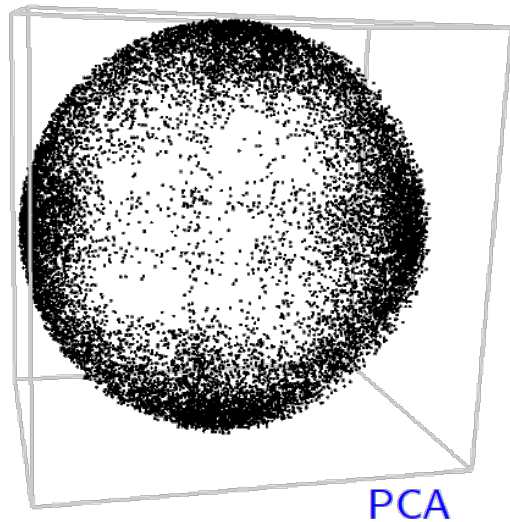
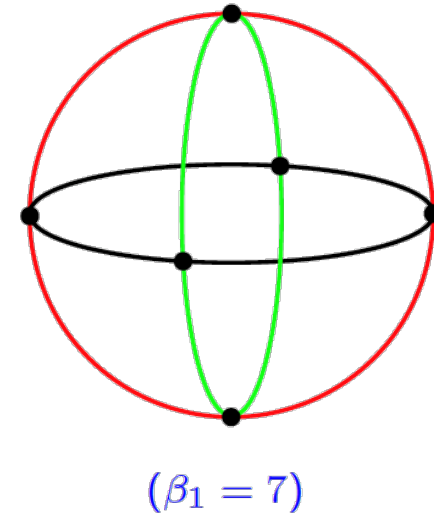
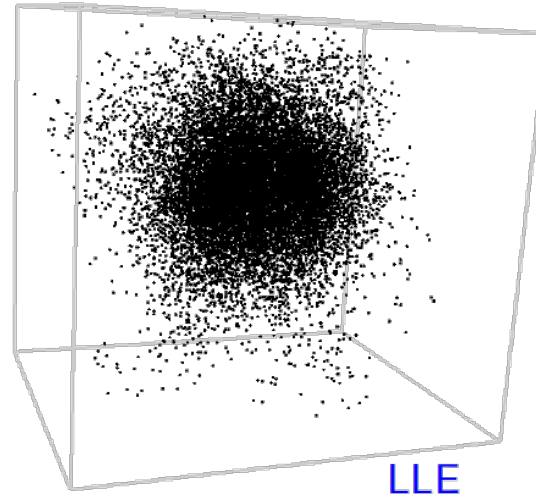
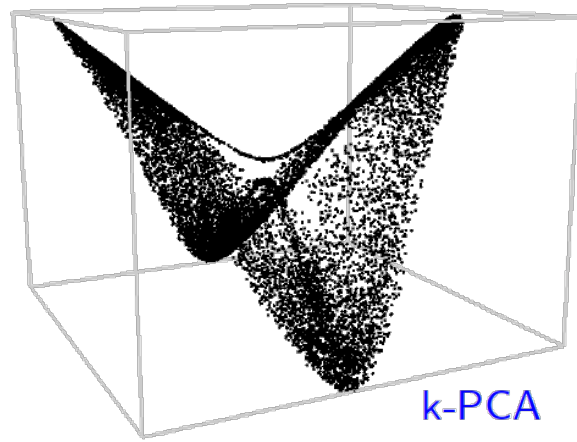


- Vertical and horizontal circles do not intersect
- Euler characteristic computation:
- $\beta_1 = \# \text{ arcs} - \# \text{ vertices} + \# \text{ connected components} = 8 - 4 + 1 = 5$
- PLEX results suggested 2-manifold.
- All homology comes from the Klein bottle.



(source: [O., Sheehy 13])

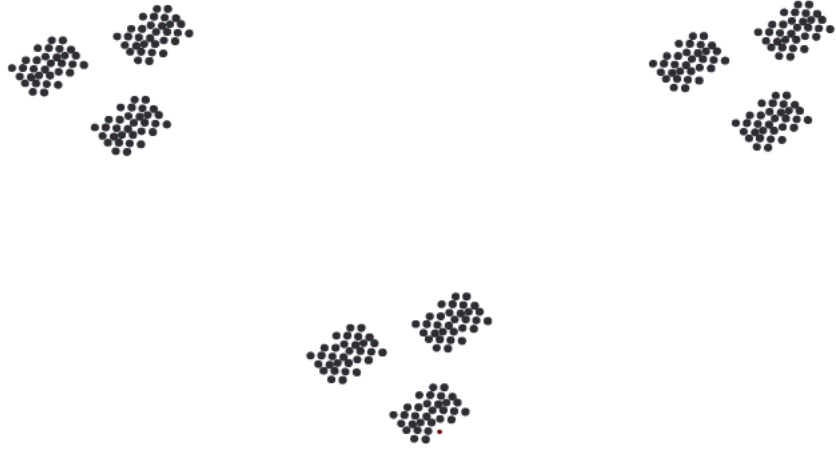
# FYI, Other Methods



# 0-Dimensional Persistence and Single Linkage Clustering

# Hierarchical Clustering

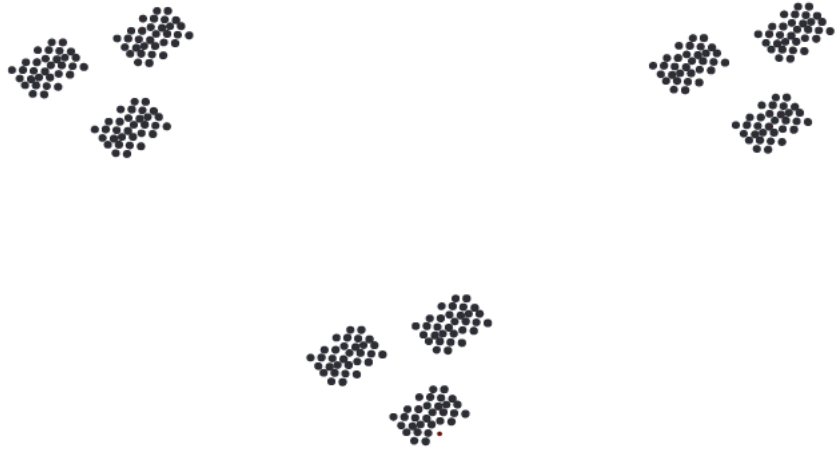
Carlsson, Mémoli, *Characterization, Stability, and Convergence of Hierarchical Clustering Methods*



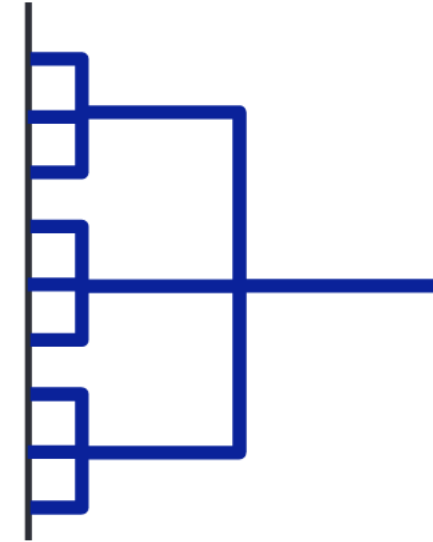
Are there three clusters? Or nine?

# Hierarchical Clustering

Carlsson, Mémoli, *Characterization, Stability, and Convergence of Hierarchical Clustering Methods*



Are there three clusters? Or six?

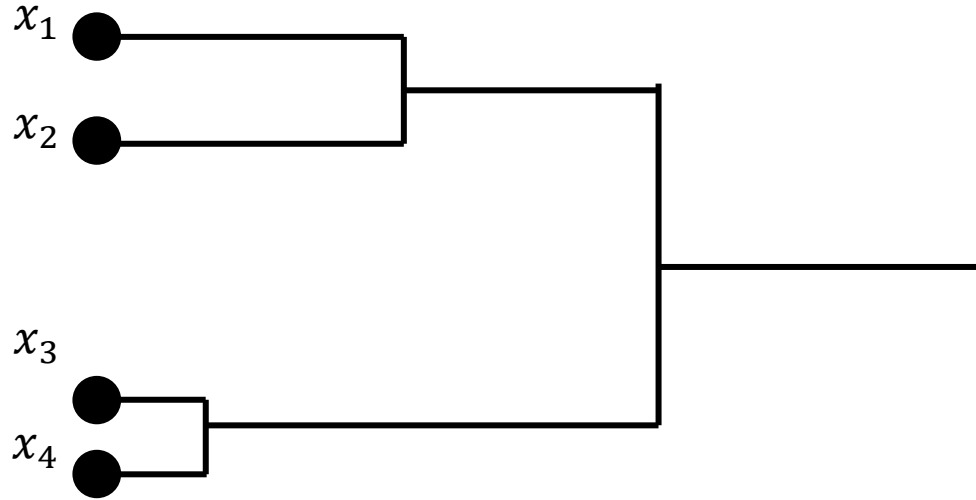


Represent clusters at **all** scales via dendrogram (= “tree” + “drawing”)



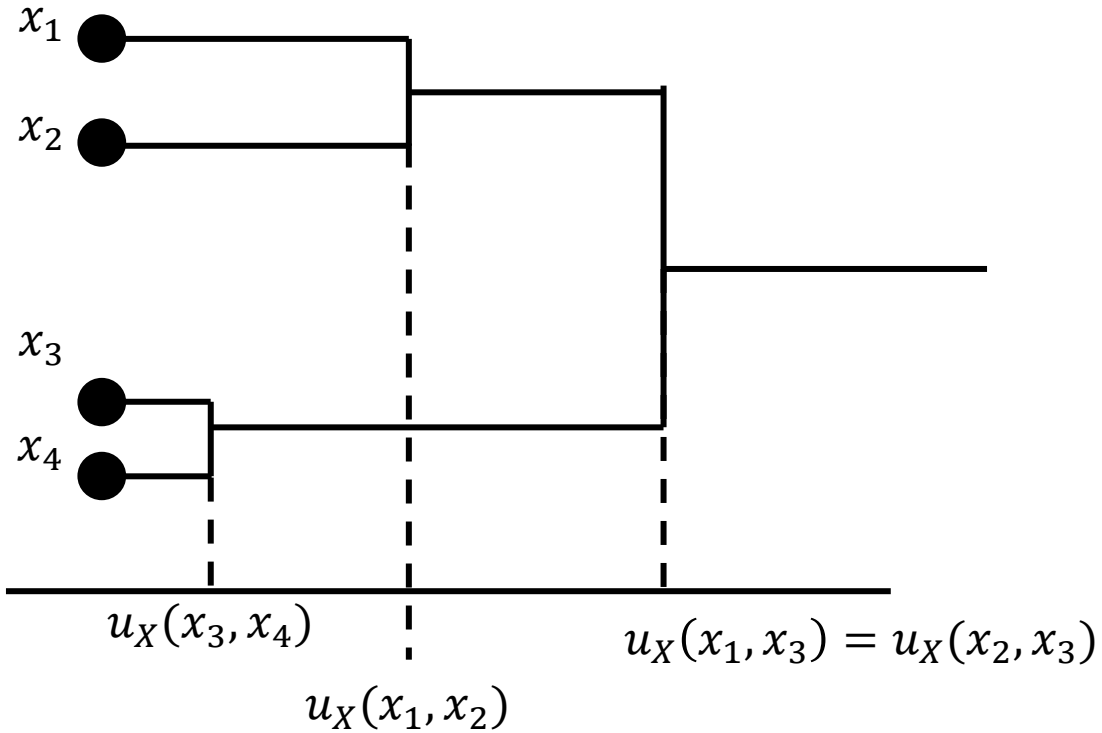
# Single Linkage Clustering

Carlsson, Mémoli, *Classifying clustering schemes*



# Single Linkage Clustering

Carlsson, Mémoli, *Classifying clustering schemes*

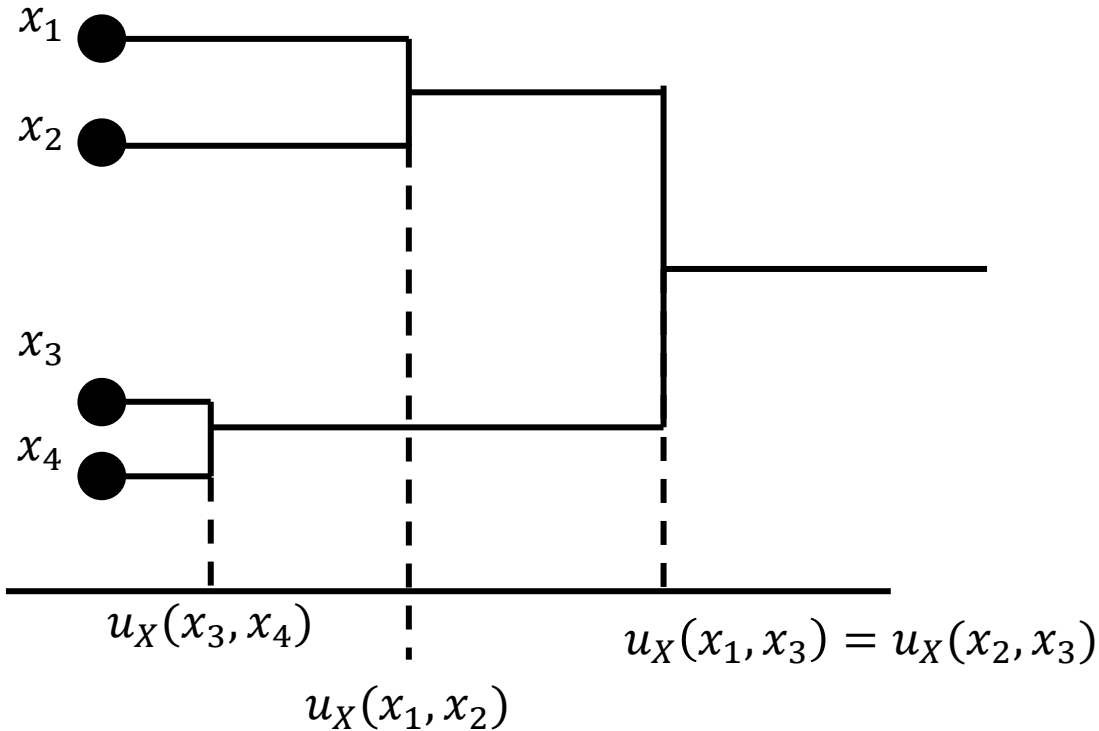


Fact: { dendrograms }  $\leftrightarrow$  { ultrametric spaces }

Exercise: Given an ultrametric space, draw the corresponding dendrogram

# Single Linkage Clustering

Carlsson, Mémoli, *Classifying clustering schemes*

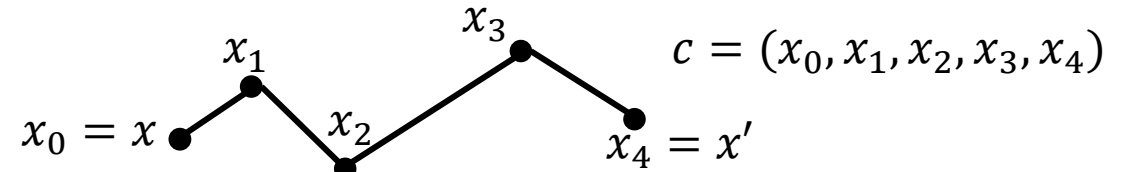


Fact: { dendrograms }  $\leftrightarrow$  { ultrametric spaces }

Exercise: Given an ultrametric space, draw the corresponding dendrogram

## Example: Single linkage clustering

- Input:  $(X, d_X)$  finite metric space
- $\mathcal{C}(x, x') =$  paths starting at  $x$  and ending at  $x'$



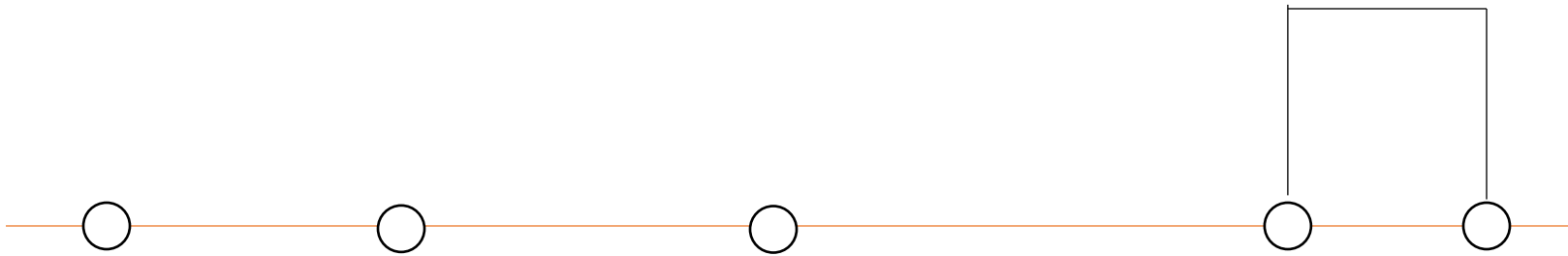
- For  $c \in \mathcal{C}(x, x')$ ,  $cost(c) = \max_{x_i, x_{i+1} \in c} d_X(x_i, x_{i+1})$
- $u_X(x, x') := \min_{c \in \mathcal{C}(x, x')} cost(c)$
- Output:  $(X, u_X)$  **ultrametric** space  $\leftrightarrow$  dendrogram

- Hierarchical clustering methods can be viewed as functors  $\mathcal{H}: \mathcal{M} \rightarrow \mathcal{M}^{ult}$  (metric  $\rightarrow$  ultrametric)
- Vietoris-Rips filtration + 0-dimensional persistence  $\approx$  single linkage clustering!

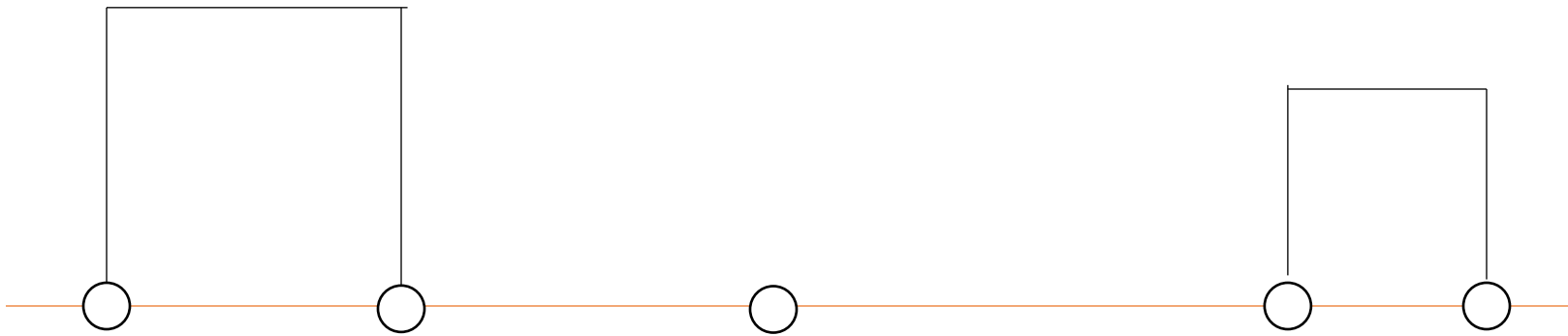
# 0-Dimensional VR Persistence and Single Linkage Clustering



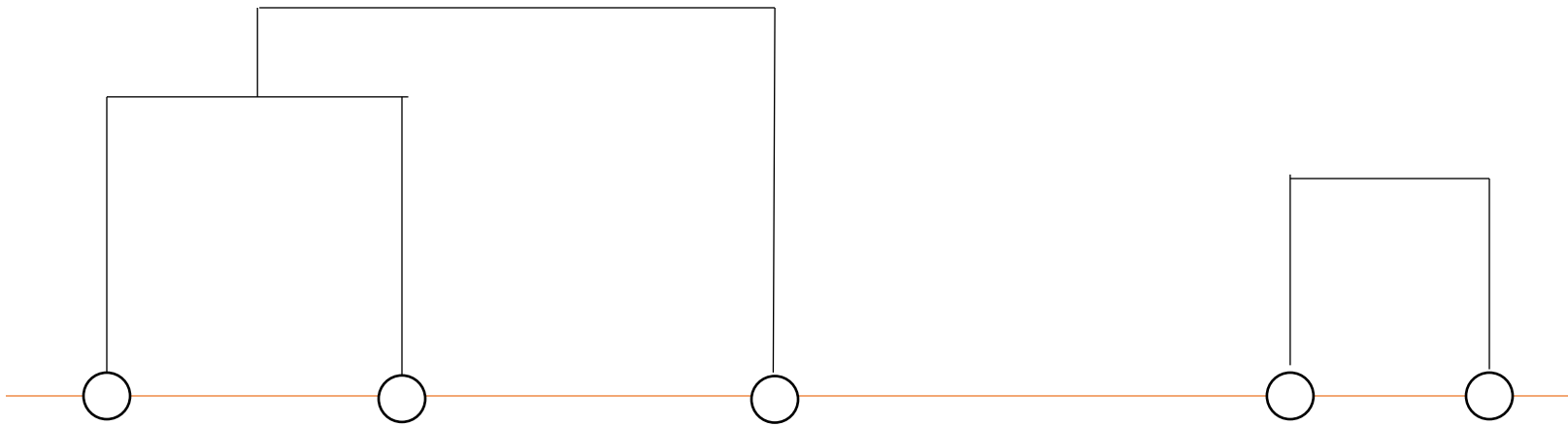
# 0-Dimensional VR Persistence and Single Linkage Clustering



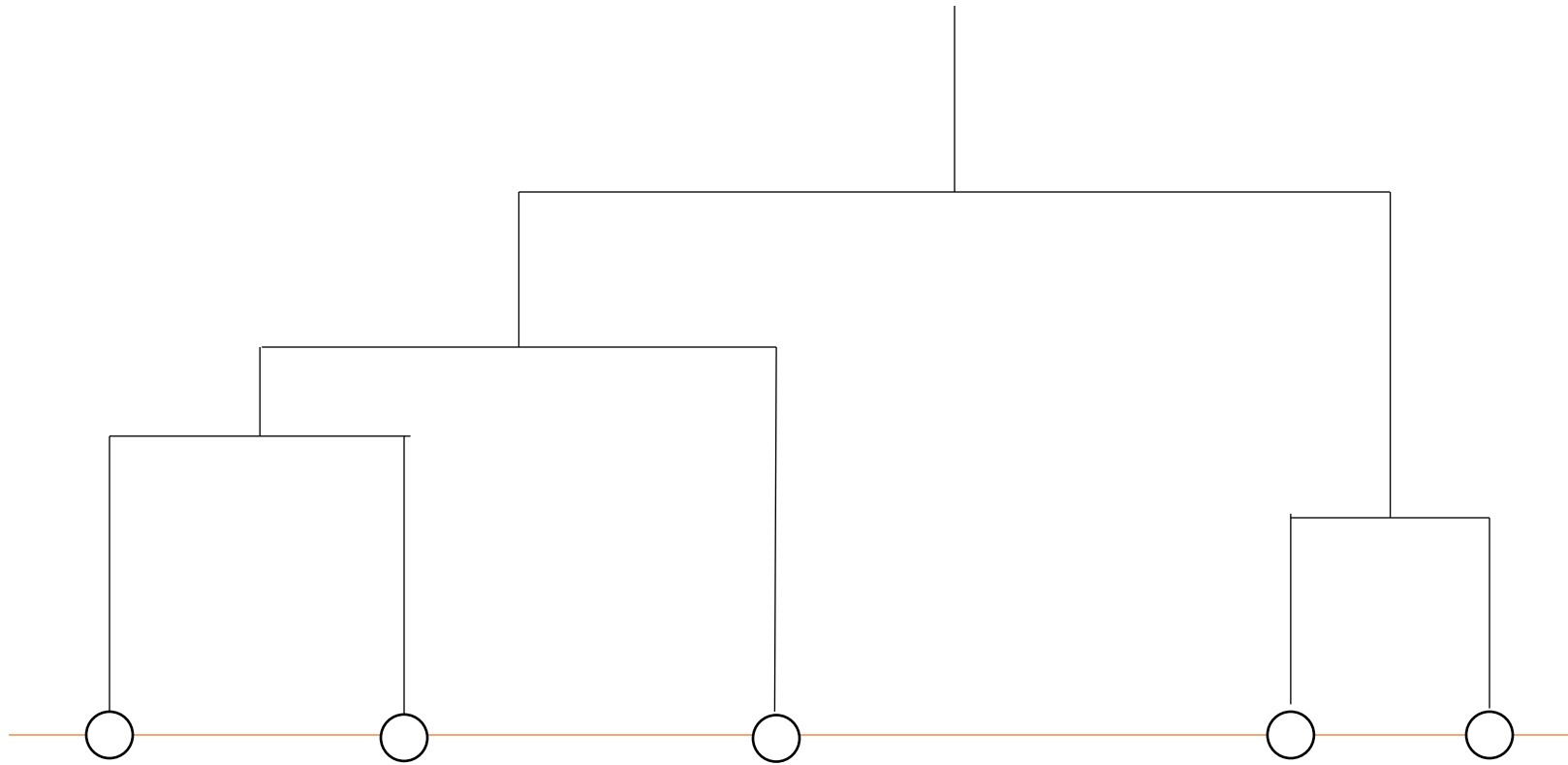
# 0-Dimensional VR Persistence and Single Linkage Clustering



# 0-Dimensional VR Persistence and Single Linkage Clustering

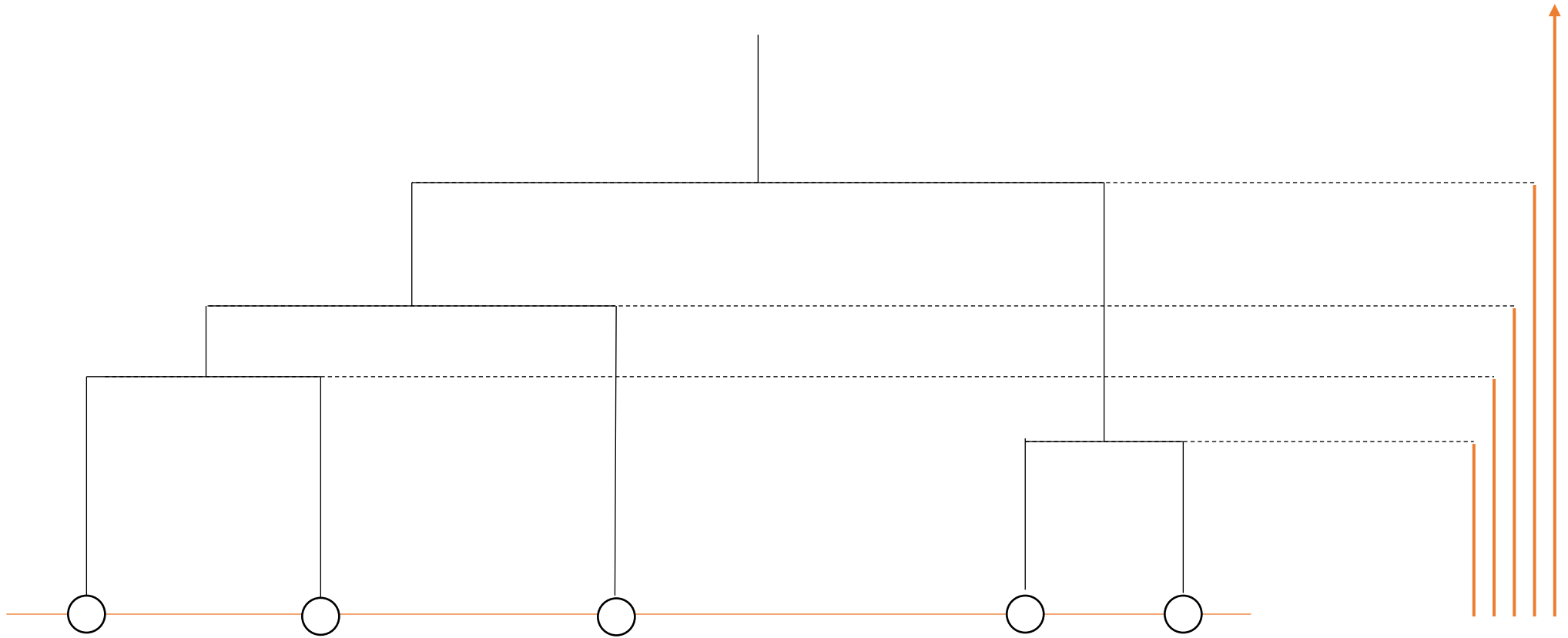


# 0-Dimensional VR Persistence and Single Linkage Clustering



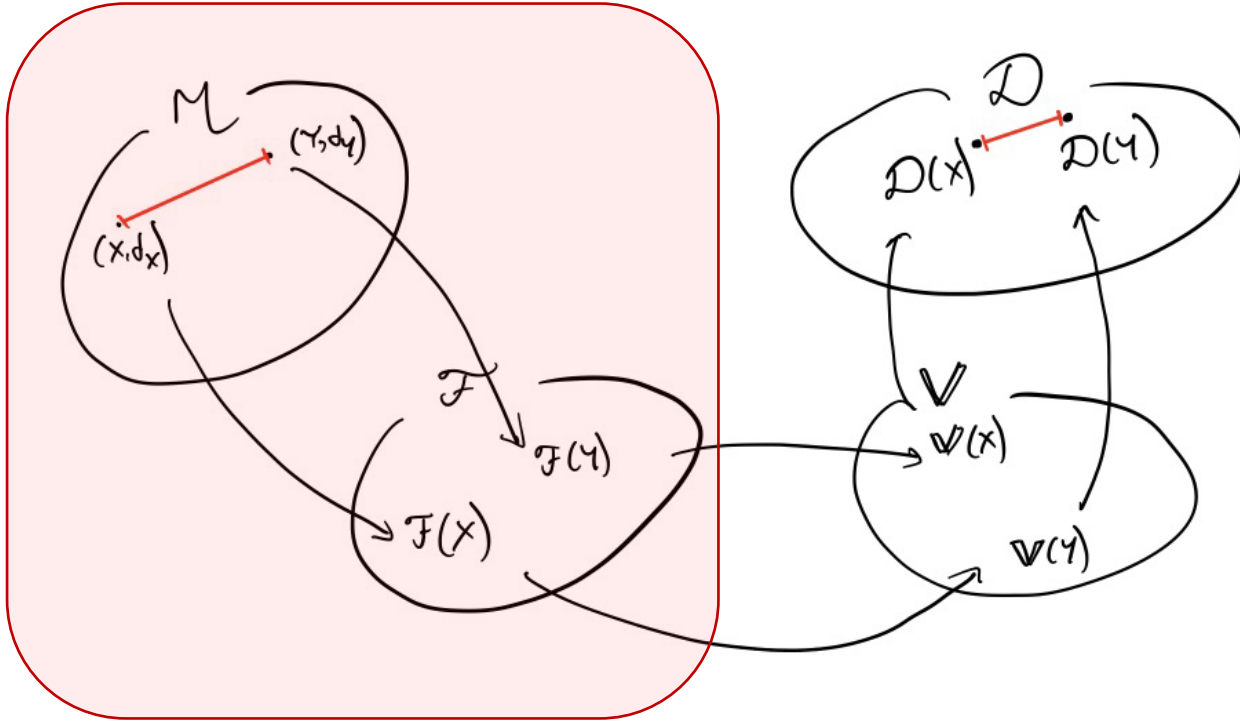
# 0-Dimensional VR Persistence and Single Linkage Clustering

Dendrogram → 0-Dim Persistence barcode



# Getting More Out of Topology

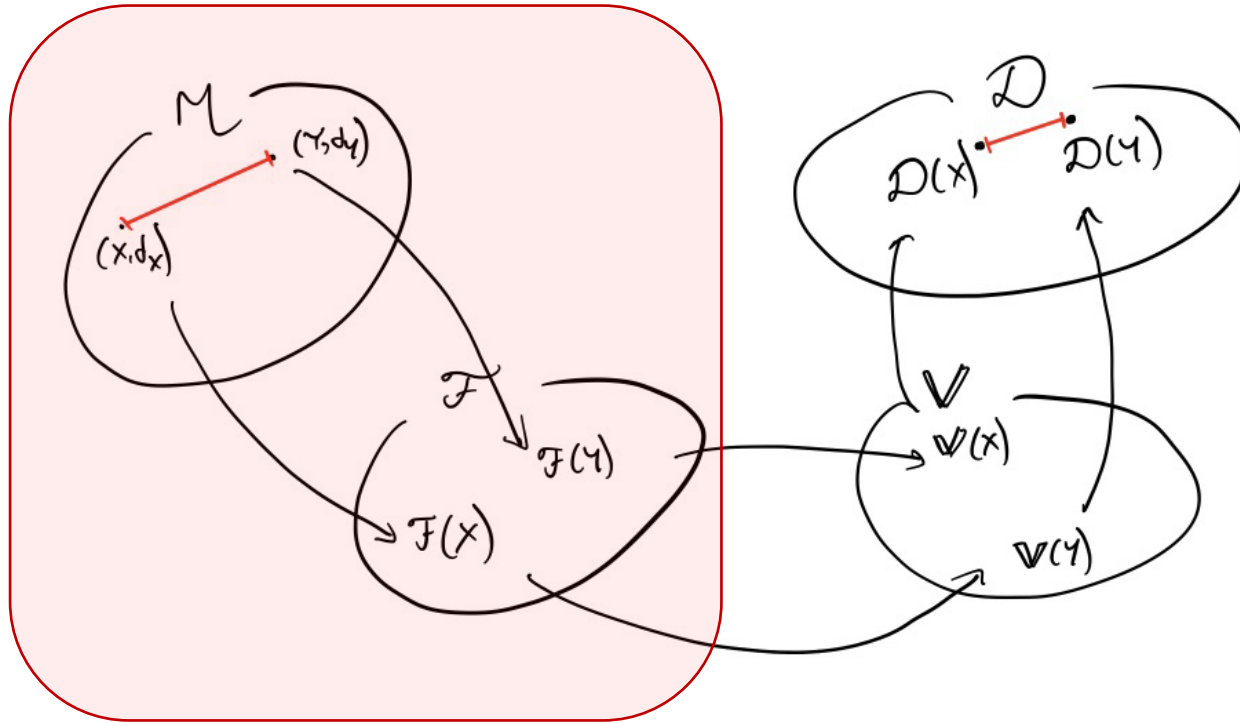
# Synthetic Filtrations are Possible, and Informative



Why only build complexes capturing proximity?

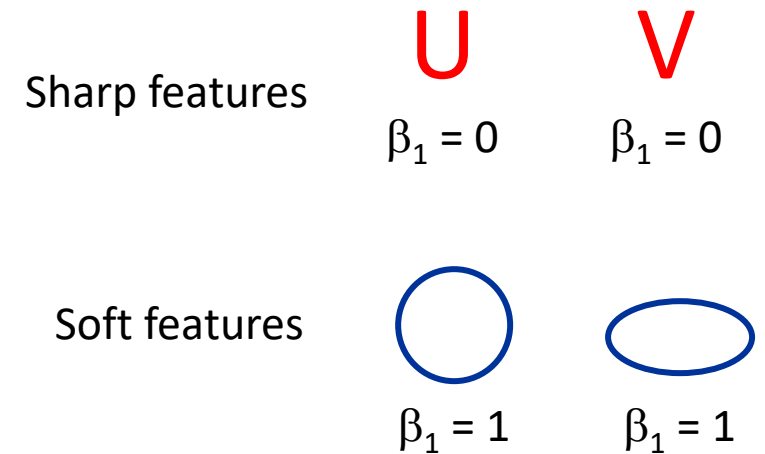
- Possible to create “derived complexes” that capture different information
- Natural to consider geometry-dependent filtrations

# Synthetic Filtrations are Possible, and Informative



Why only build complexes capturing proximity?

- Possible to create “derived complexes” that capture different information
- Natural to consider geometry-dependent filtrations



Geometry  
discriminating

Topology  
classifying

# Making Topology a Finer Tool

Geometry  
discriminating

Topology  
classifying

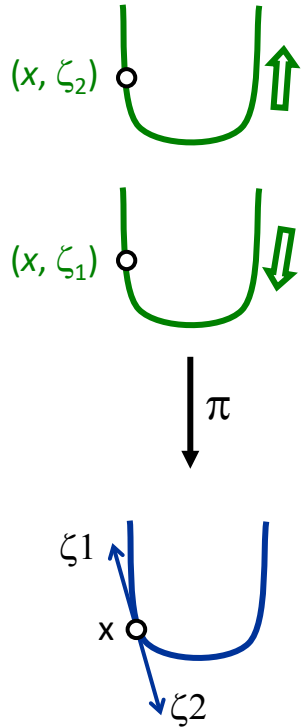
- Topology: connectivity of a space
- Key Idea: no reason to look at the original space only
  - Add geometry  $\Rightarrow$  look at **derived space(s)**
  - **Compute topology of derived space(s)**

1. Find filtration
2. Compute persistence

Our recipe

via the **tangent complex**

# 2-D Curve Tangent Complex



Covering space

$T(X)$  has **two** components:  
 $\beta_0(T(X)) = 2$

There are **two** points in its fiber  $\pi^{-1}(x)$

Every point  $x$  on a smooth curve  $X$  has **two** tangent directions.

A corner point has four tangent directions:  
 $\beta_0(T(X)) = 4$

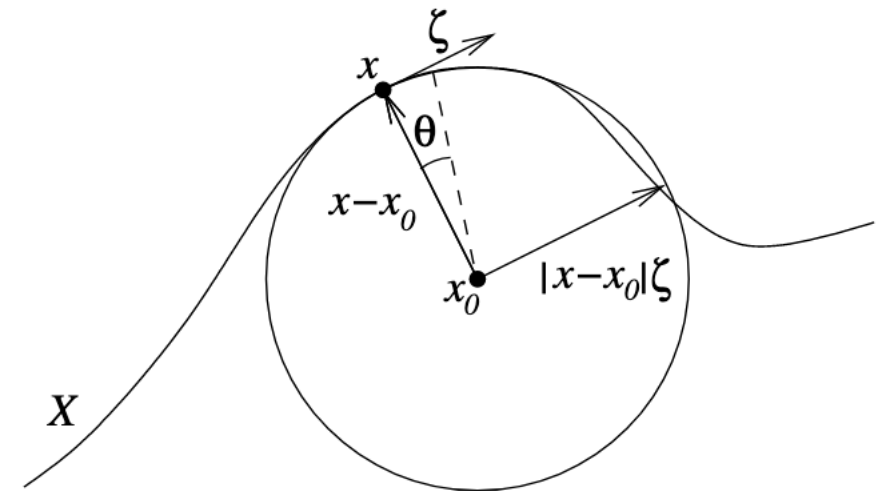
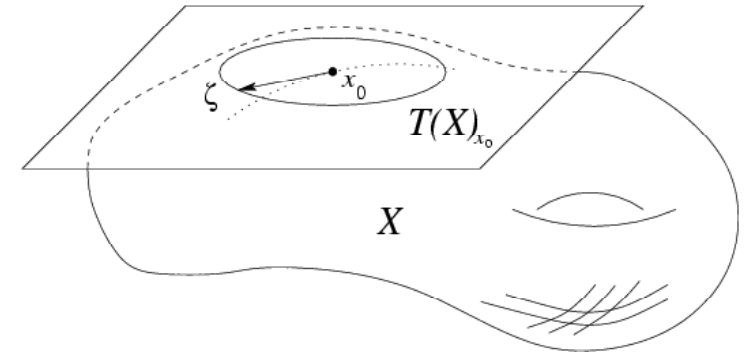


# The (Filtered) Tangent Complex

Collins, Zomorodian, Carlsson, Guibas, *A barcode shape descriptor for curve point cloud data*, CaG'04

Setup:  $X \subseteq \mathbb{R}^n$

- Tangent complex  $T(X) \subseteq X \times \mathbb{S}^{n-1}$ :
  - unit tangent vectors at all points of  $X$ , i.e. pairs  $(x, \zeta)$  where  $x \in X$  and  $\zeta$  is a tangent vector
- Filtered tangent complex: filter the points in the tangent complex by increasing curvature
  - Curvature at point  $(x, \zeta)$  is  $1/\rho$ , where  $\rho$  is radius of osculating circle
  - $T_\delta(X) =$  points with curvature  $< \delta$
  - Filtered tangent complex  $T^{filt}(X) := \{T_\delta(X)\}_{\delta>0}$



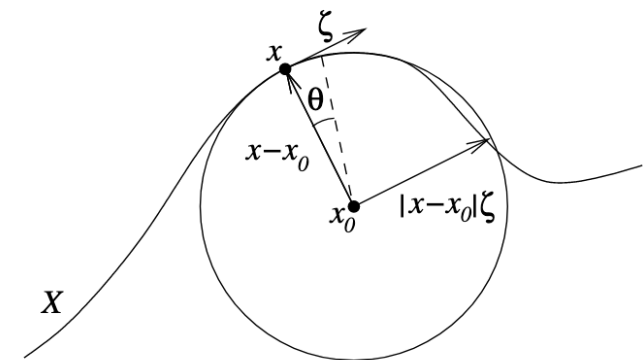
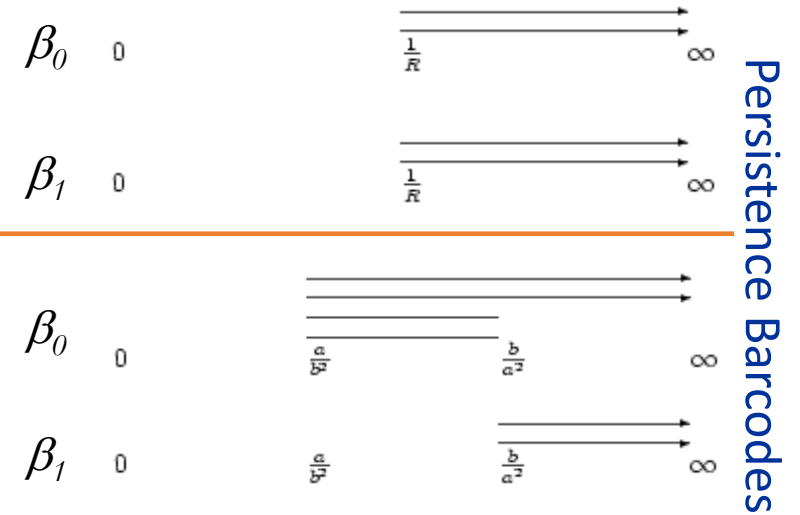
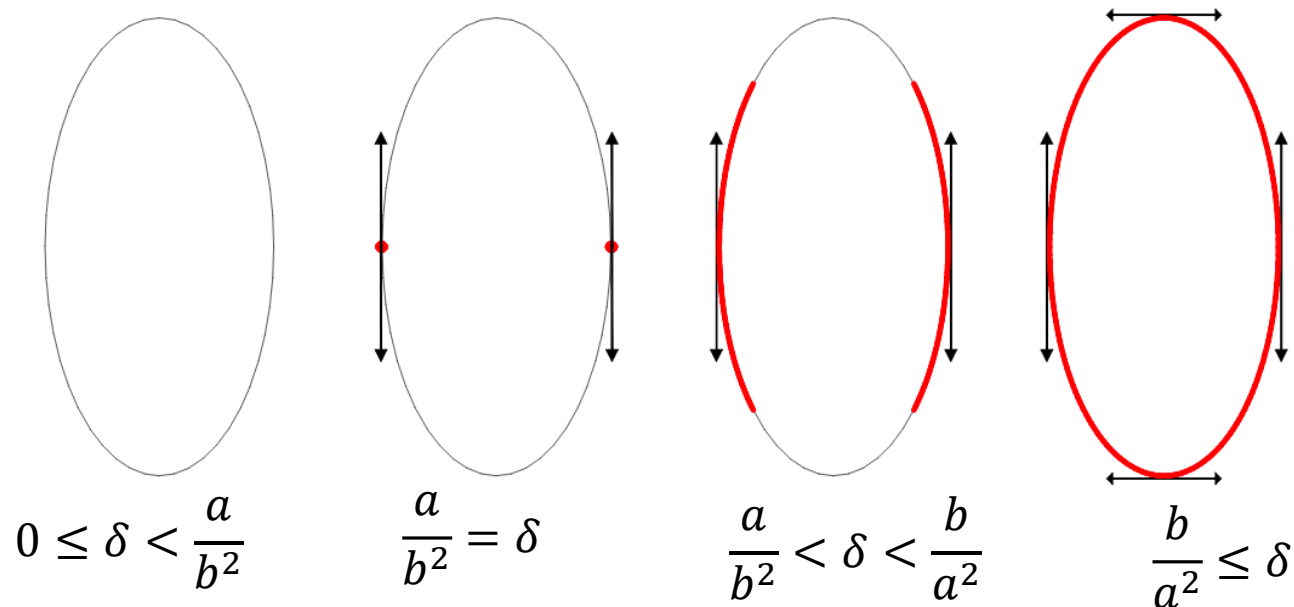
# Examples: Circle vs Ellipse

**Circle of radius R:** Tangent complex is  $S^1 \times S^0$ , as there are two tangent directions at each point.

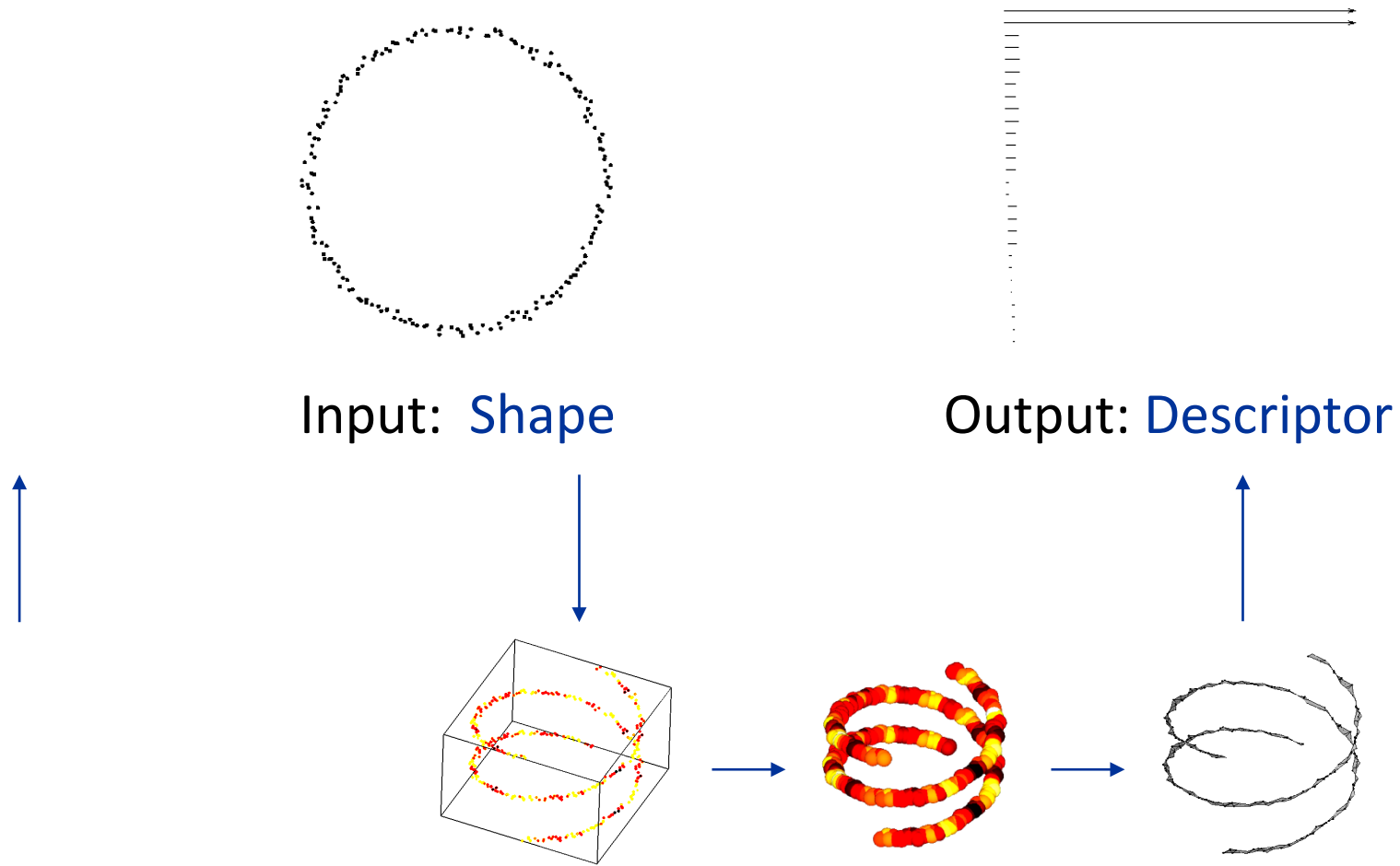
- $T^{filt}$  is empty until  $1/R$ , then full complex enters at once

**Ellipse**  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ :

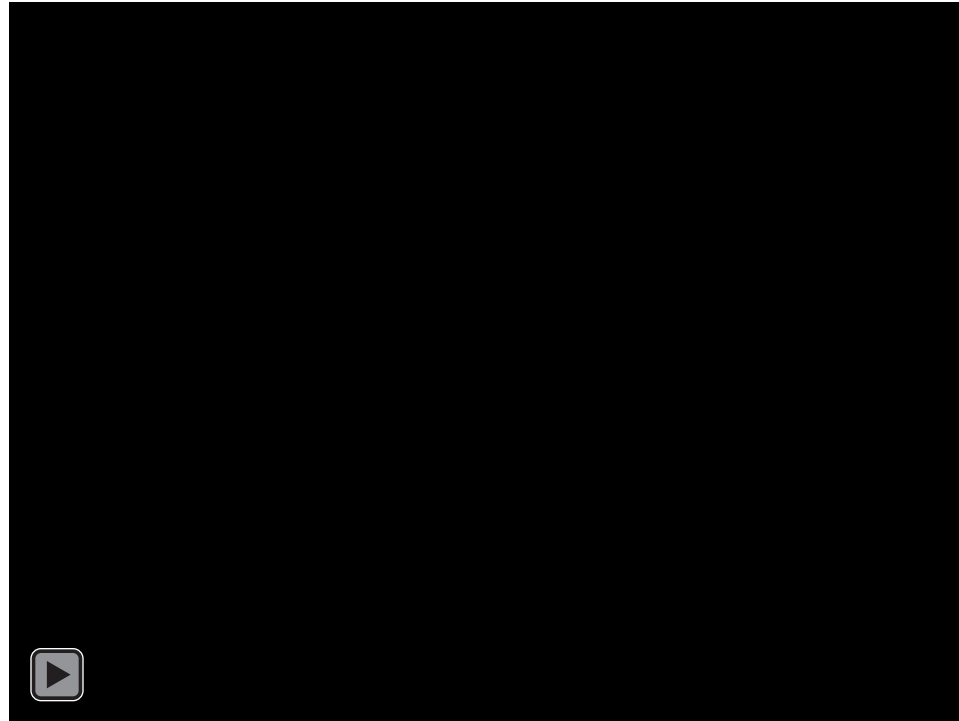
- evolves through **four stages**: points at *lower* curvature appear earlier.



# Applying Barcodes to 2D PCDs

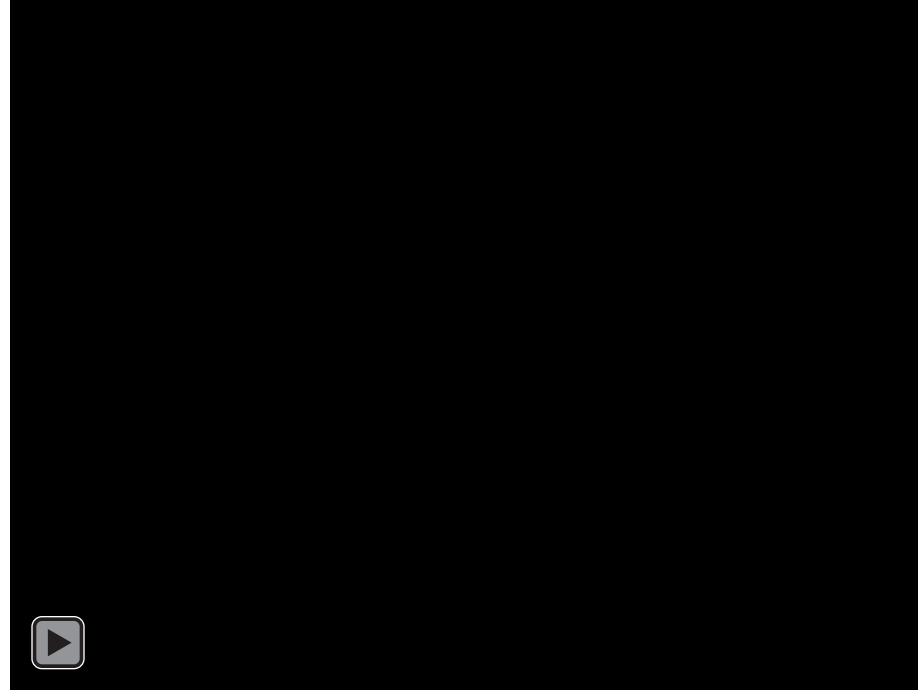


# Fibers



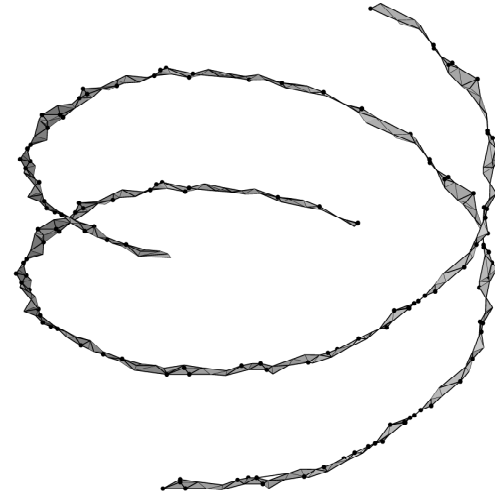
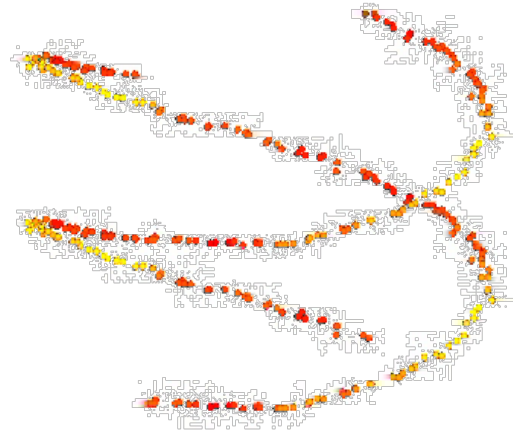
- PCD  $P \subseteq X$ , sampled from smooth closed 1-manifold
- We compute tangent fibers  $\pi^{-1}(P)$  by normal estimation at each point

# Filtering by Curvature



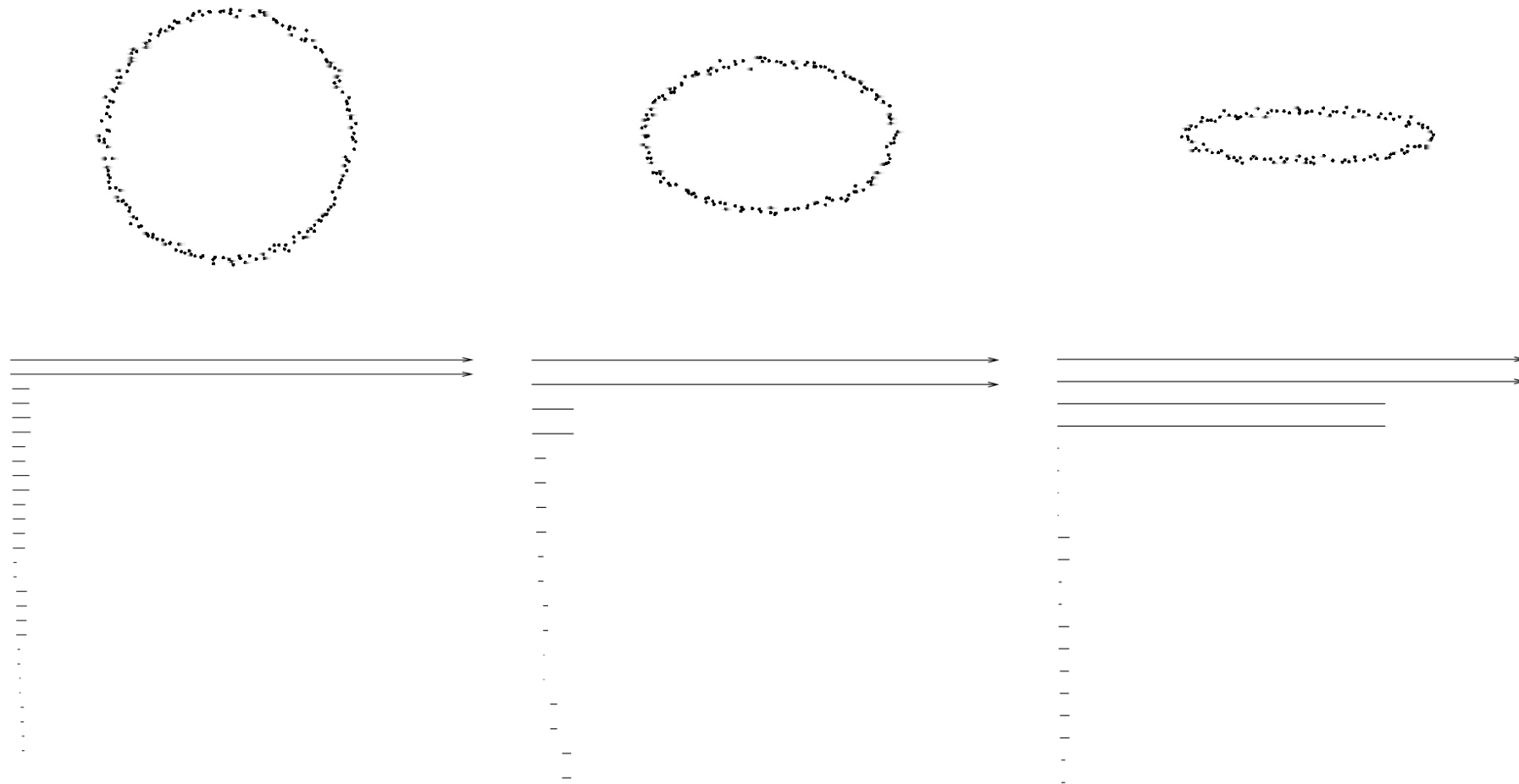
- Construct tangent complex incrementally
- Transform points to coordinate frame provided by tangent computation
- Fit osculating parabola to estimate curvature (more robust integral methods possible)

# Approximating $T(X)$

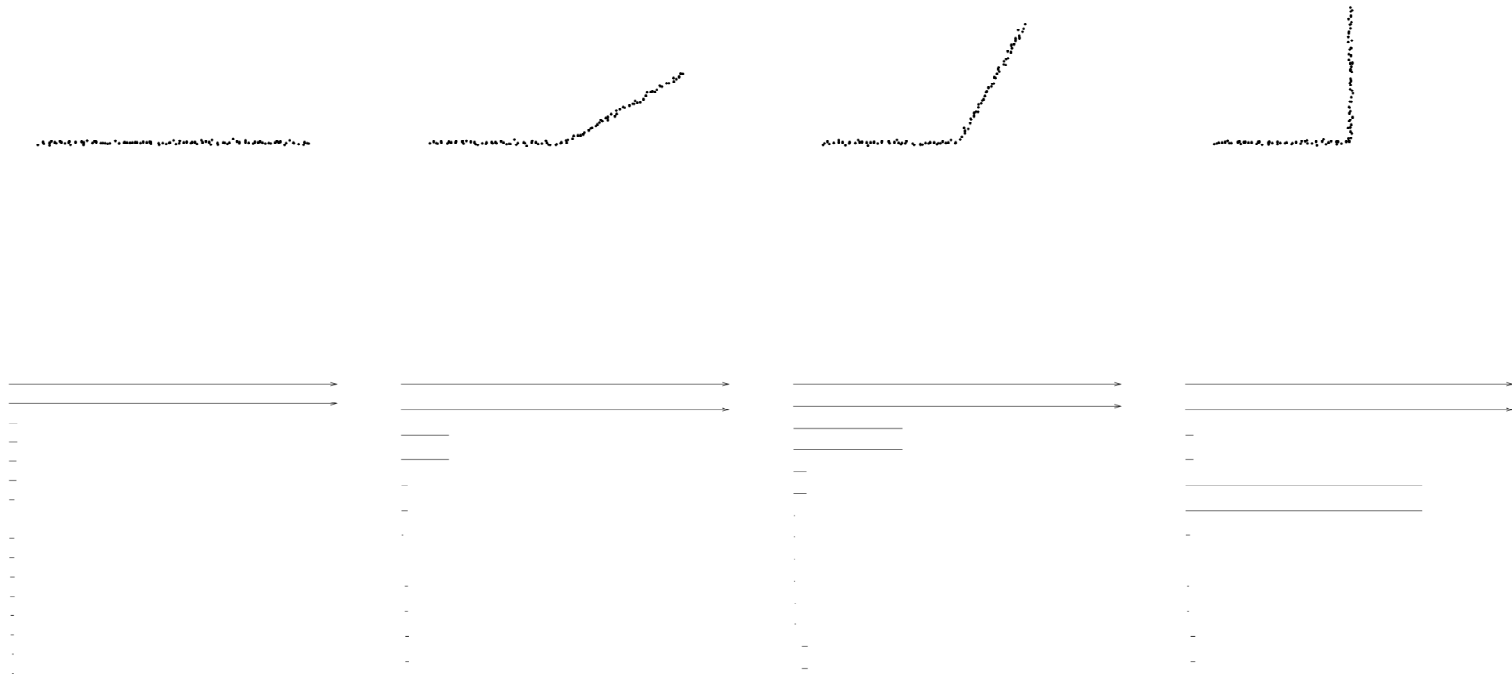


- $\mathbb{R}^n \otimes S^{n-1}$  with  $ds^2 = dx^2 + \omega^2 d\zeta^2$
- $T(X) \cong \bigcup_{p \in \pi^{-1}(p)} B_\varepsilon(p)$

# Family of Ellipses



# Articulated Arm Parametrization



# The Mapper Algorithm

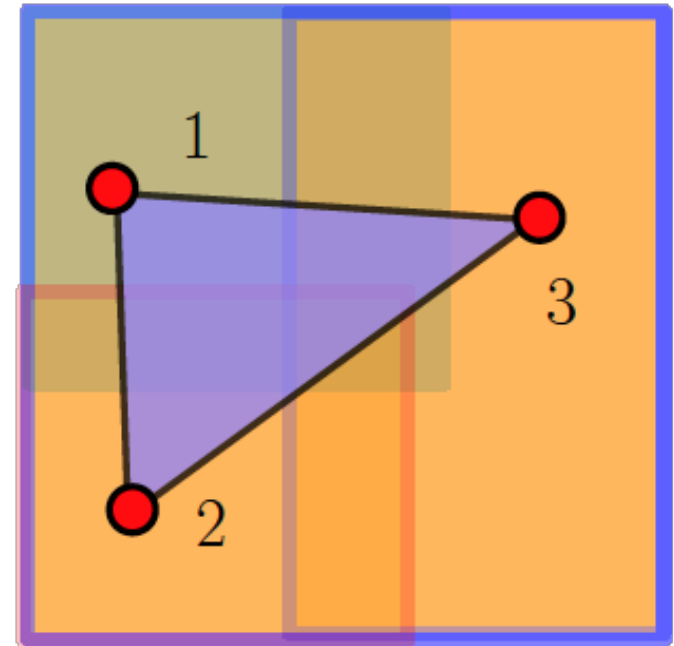
# Review: Covers and Nerves

## Finite cover of a topological space $X$

- ▶  $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$  for a finite index set  $A$ .
- ▶ each  $U_\alpha \subseteq X$  is open and  $X = \bigcup_{\alpha \in A} U_\alpha$

## Nerve of a cover

- ▶ Simplicial complex:  $N(\mathcal{U})$  with vertex set  $A$ .
- ▶ simplices:  $A \supseteq \sigma \in N(\mathcal{U}) \Leftrightarrow \bigcap_{\alpha \in \sigma} U_\alpha \neq \emptyset$ .



# Pullback Covers and Their Nerves

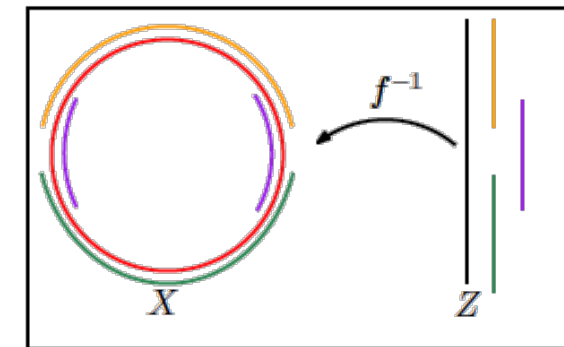
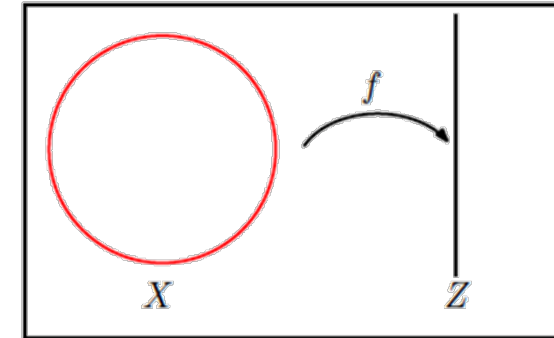
Studying data by looking at “lens” functions over the data

- ▶ Assume you have  $f : X \rightarrow Z$  well behaved continuous function and  $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$  finite open cover of  $Z$ .
- ▶ For each  $\alpha \in A$  consider the **connected components** of  $f^{-1}(U_\alpha) = \{V_{i,\alpha}, 1 \leq i \leq j_\alpha\}$ .
- ▶ Let  $f^*(\mathcal{U})$  be the (finite) open cover of  $X$  thus induced:

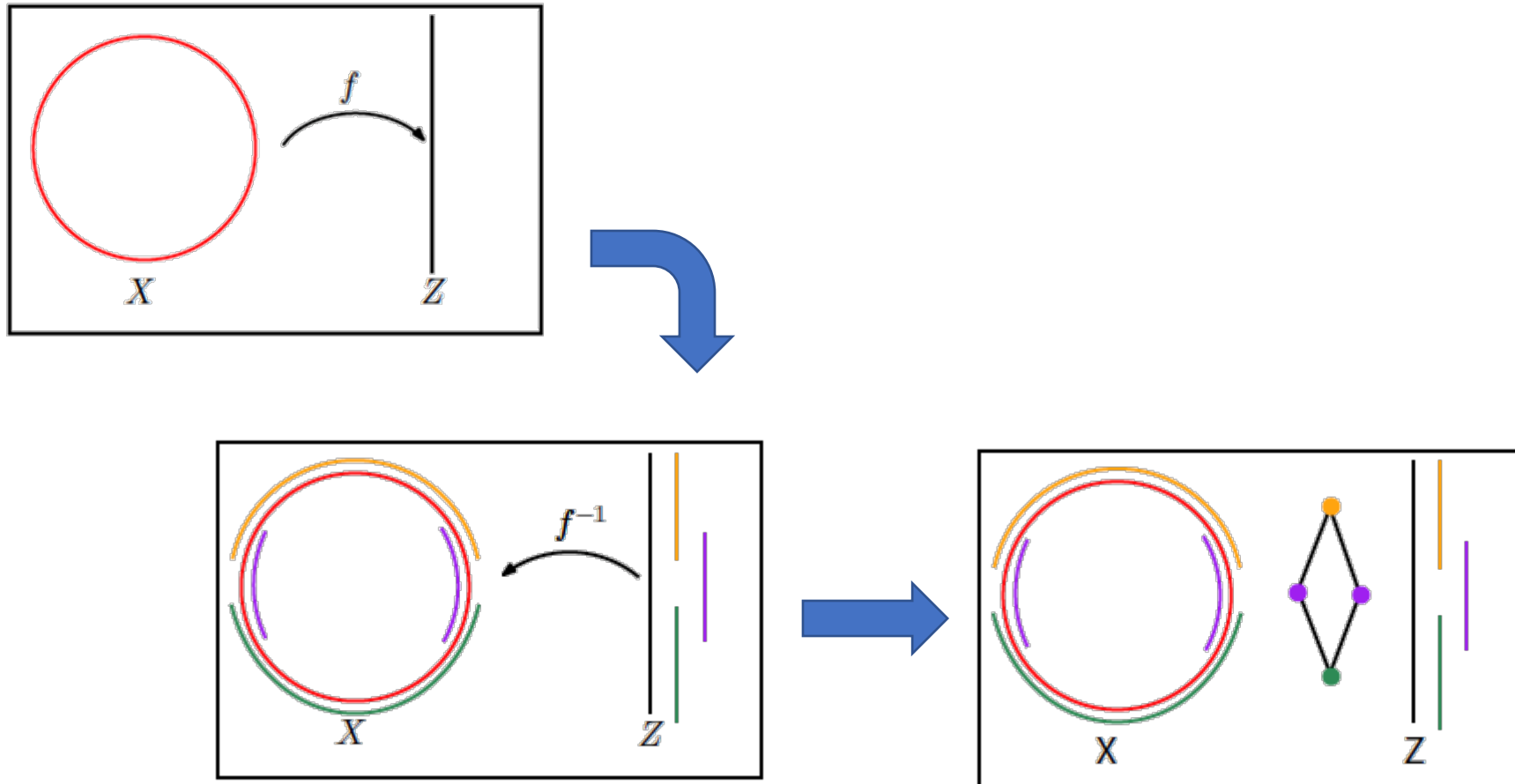
$$f^*(\mathcal{U}) := \{V_{i,\alpha}, 1 \leq i \leq j_\alpha, \alpha \in A\}.$$

This is the **pullback** of  $\mathcal{U}$  via  $f$ .

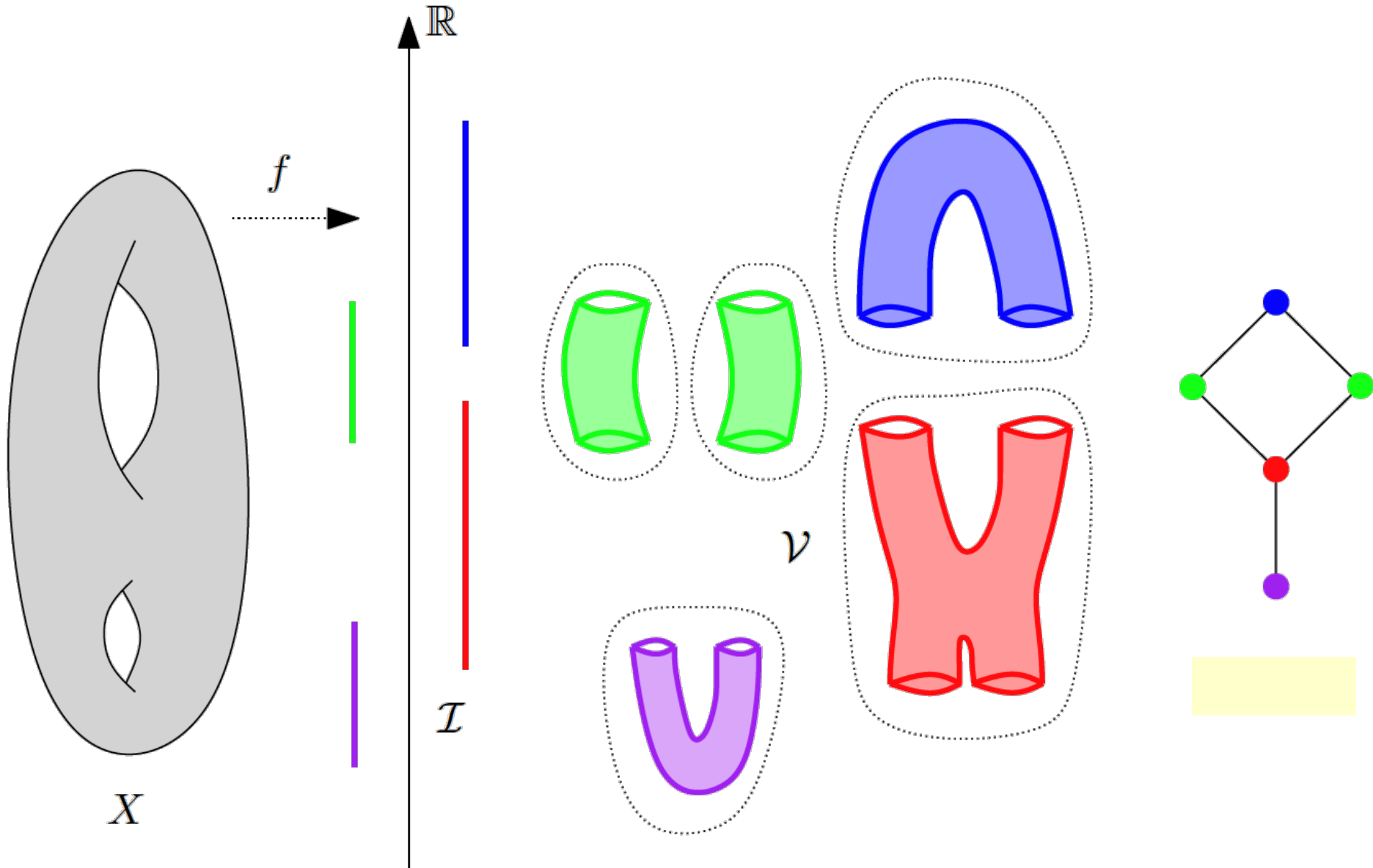
- ▶ Now consider the nerve of the pullback:  $N(f^*(\mathcal{U}))$ . This complex often retains structural information about underlying space  $X$ .



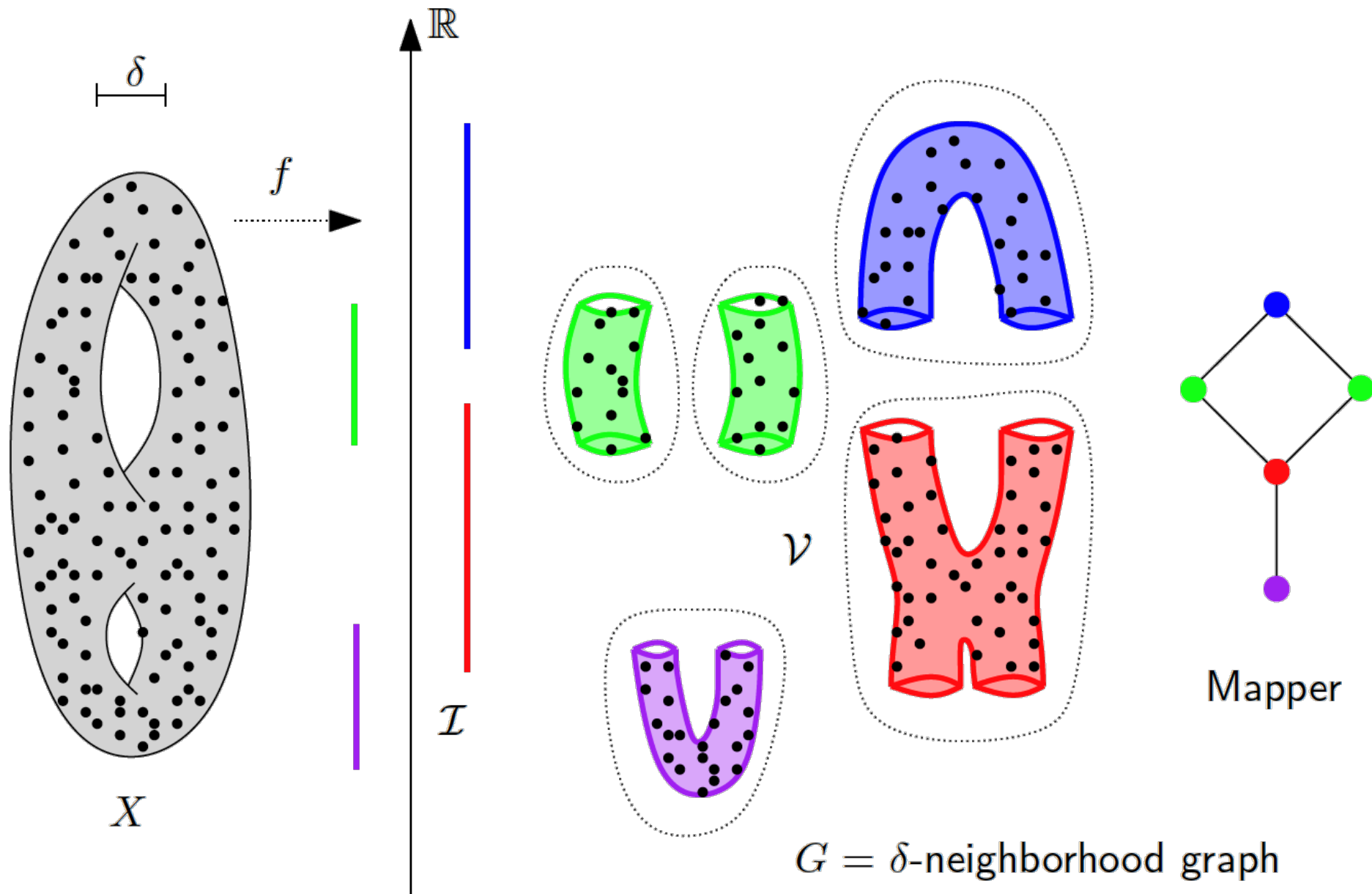
# Pullback Covers and Their Nerves



# Another Example



# In Practice

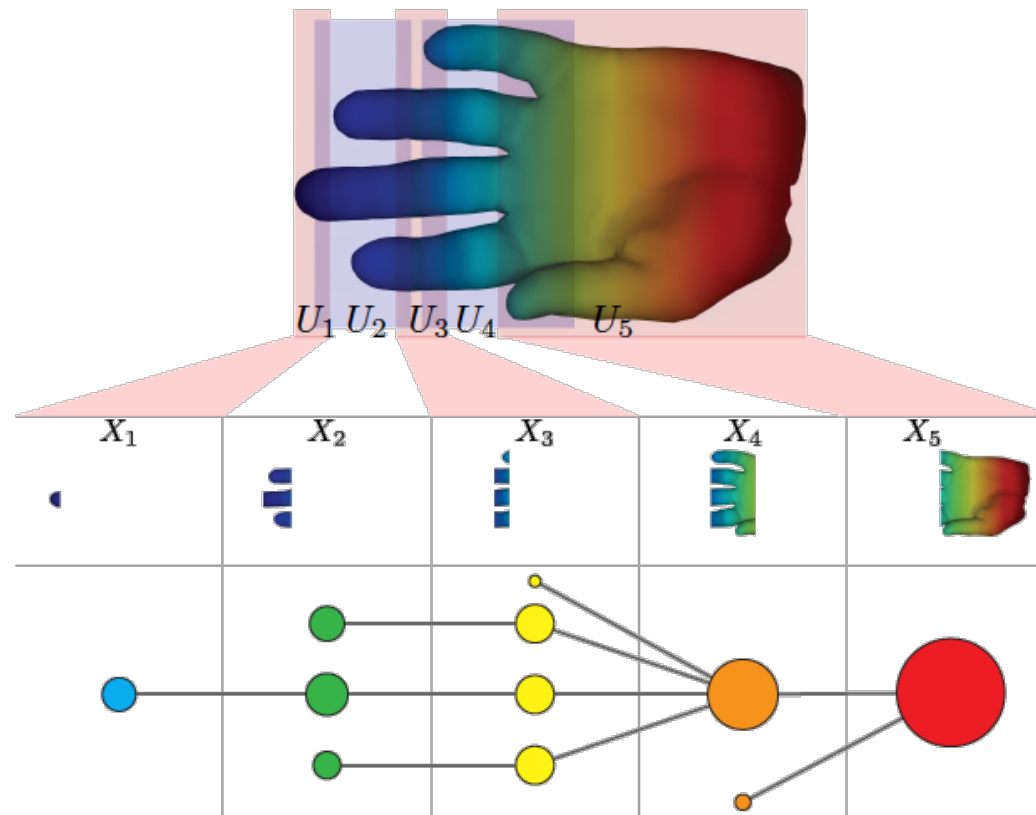


# The Mapper Algorithm

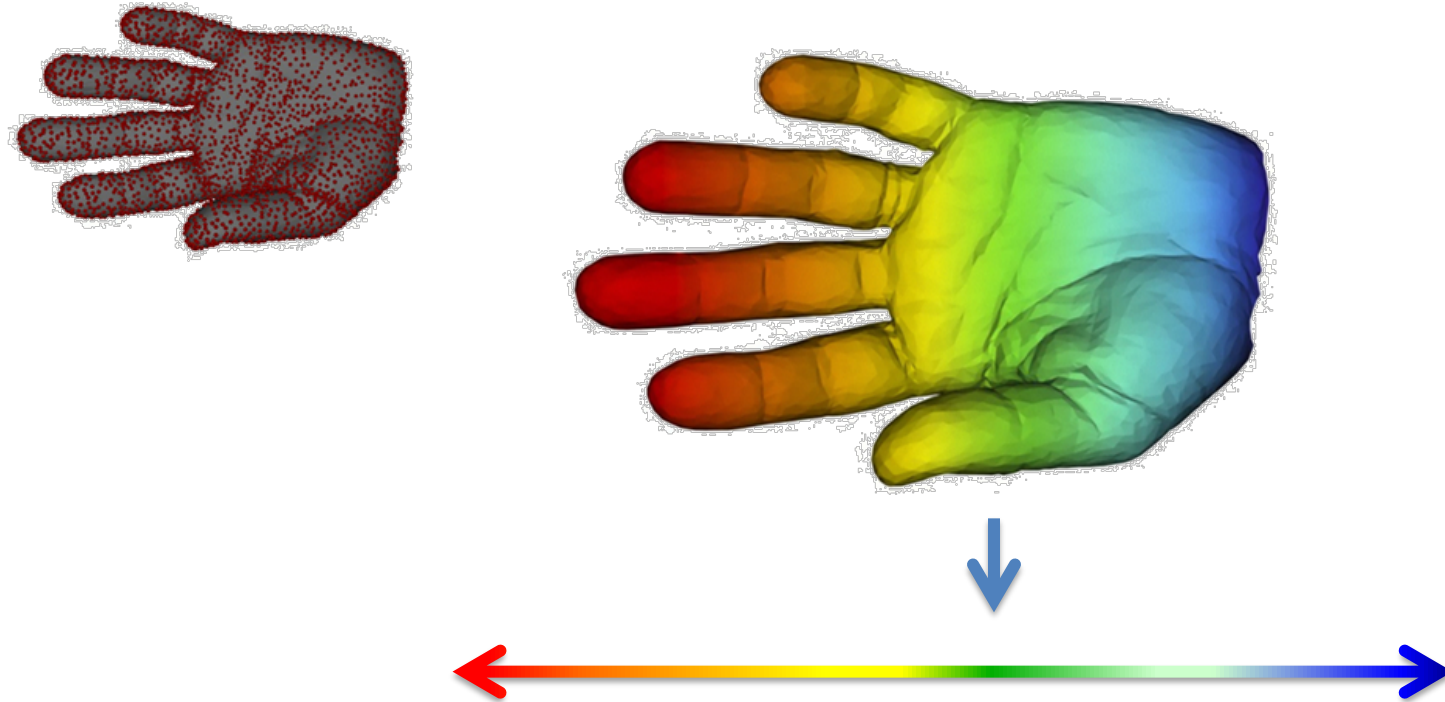
(Carlsson, Mémoli, Singh 2007)

Let  $f : X \rightarrow Z$  be well behaved and continuous and  $\mathcal{U}$  be finite open cover of  $Z$ , then the **Mapper** output corresponding to  $\mathcal{U}$  and  $f$  is

$$M(\mathcal{U}, f) := N(f^*(\mathcal{U})).$$



# Step 1: Choose a Lens / Filter Function



Function  $f$  : Data Set  $\rightarrow \mathbf{R}$

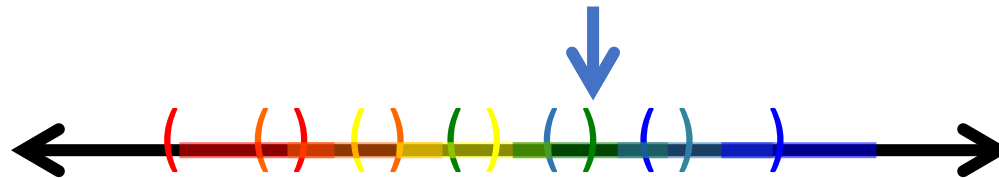
Ex 1:  $x$ -coordinate  $f : (x, y, z) \rightarrow x$

# Step 2: Partition into Overlapping Bins

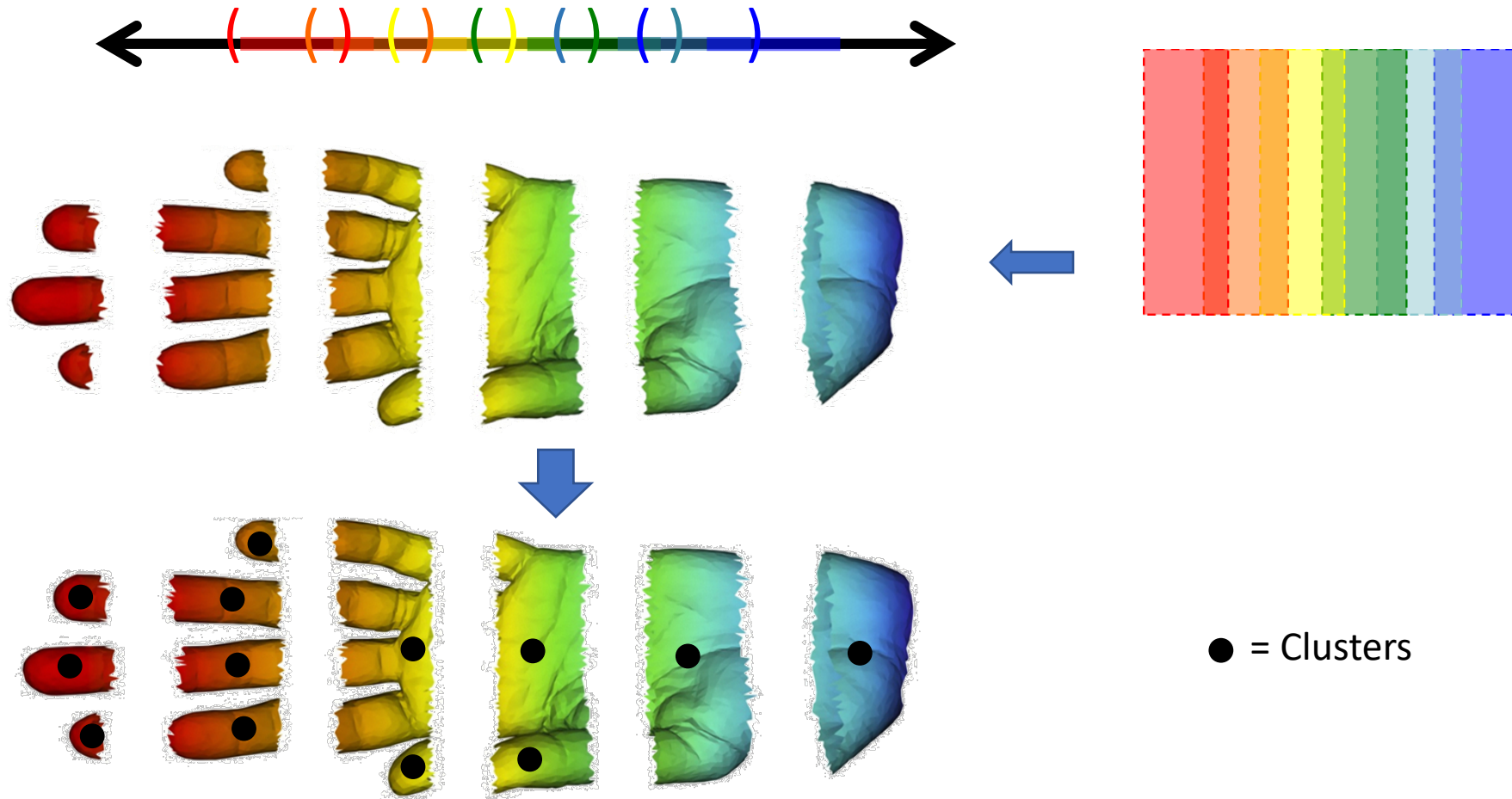


Cover data via overlapping bins.

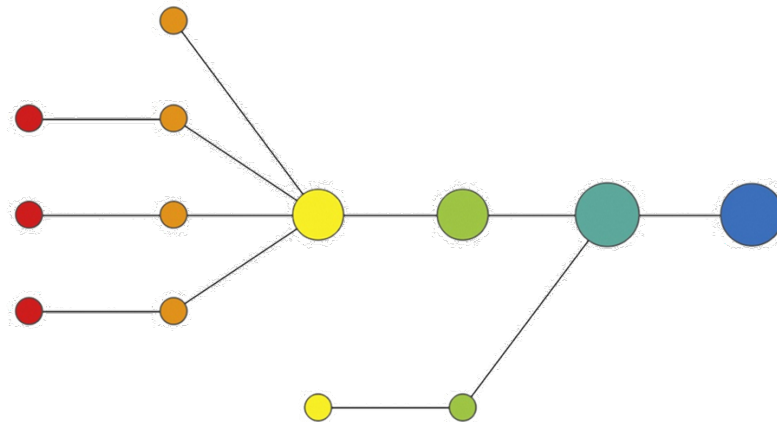
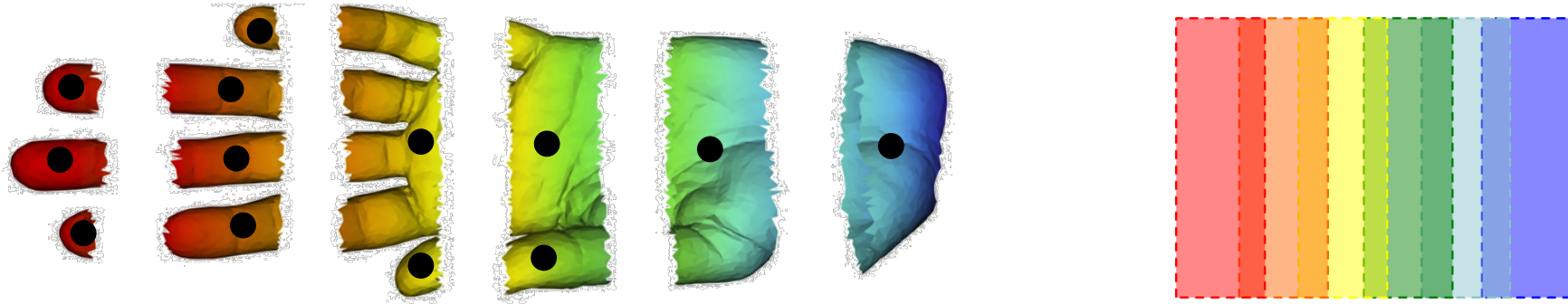
Example:  $f^{-1}(a_i, b_i)$



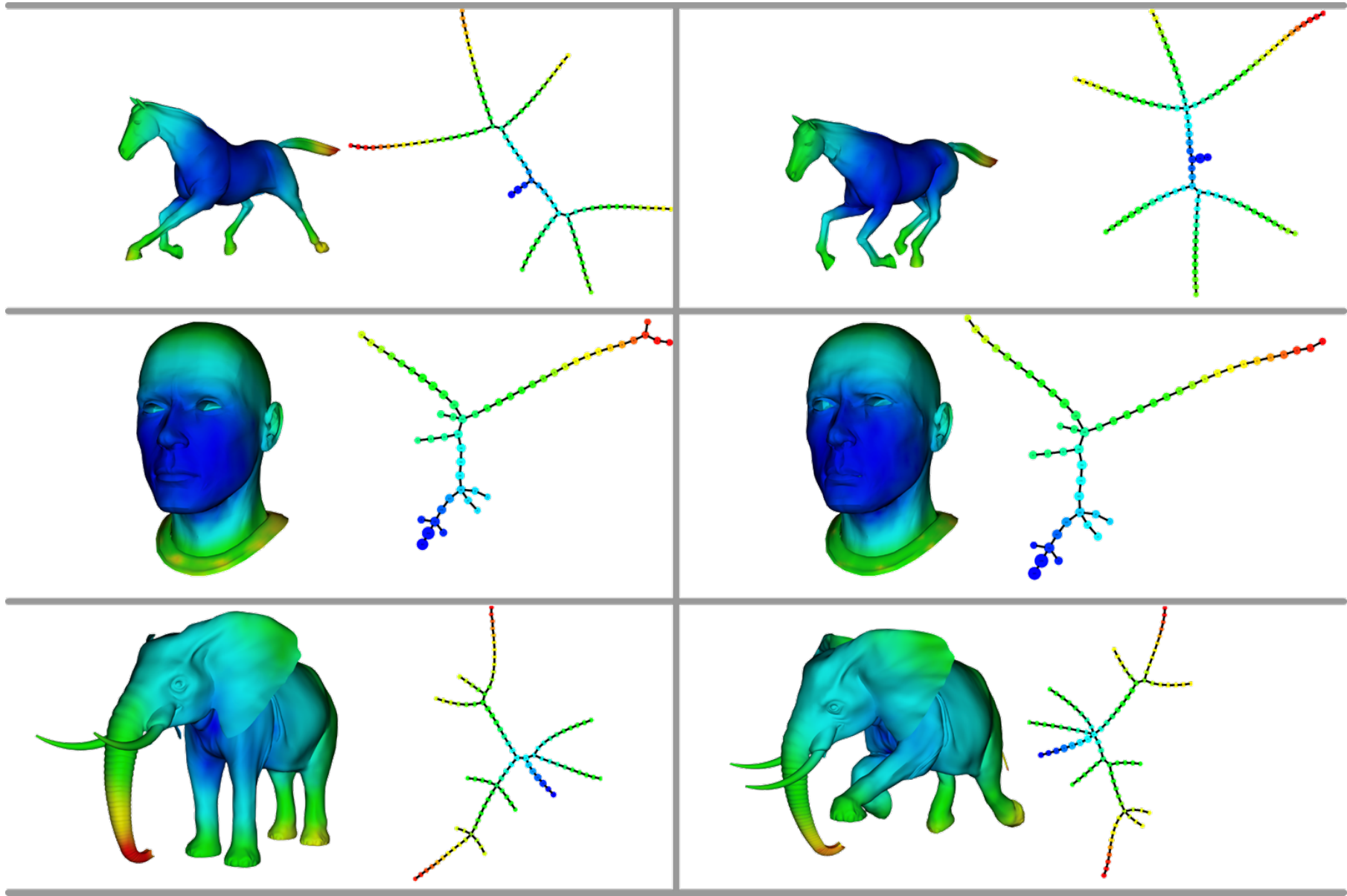
# Step 3: Form Connected Components in the Bins



# Step 4: Form a Network of Intersecting Clusters

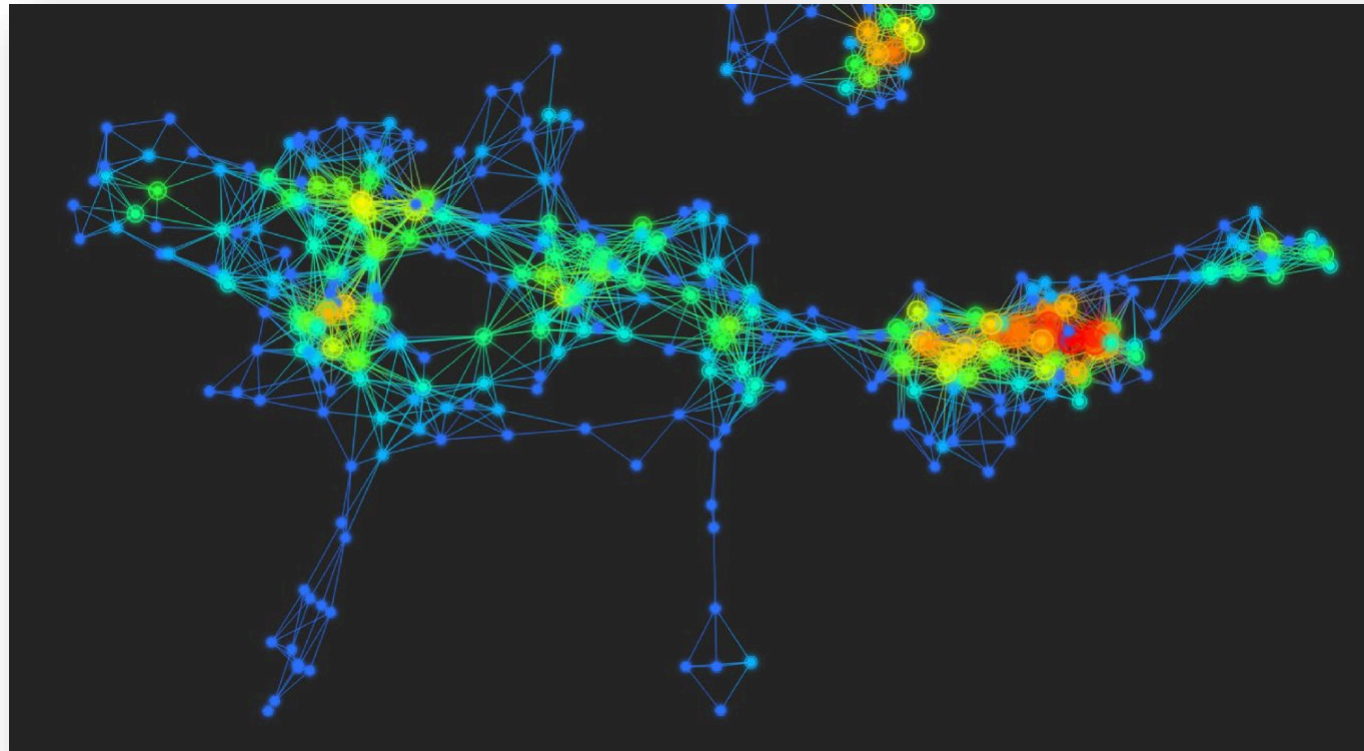


# Centrality Filter Under Deformation



# Many, Many Choices

“It is useful to think of Mapper as a camera, with lens adjustments and other settings. A different filter function may generate a network with a different shape, thus allowing one to explore the data from a different mathematical perspective.”



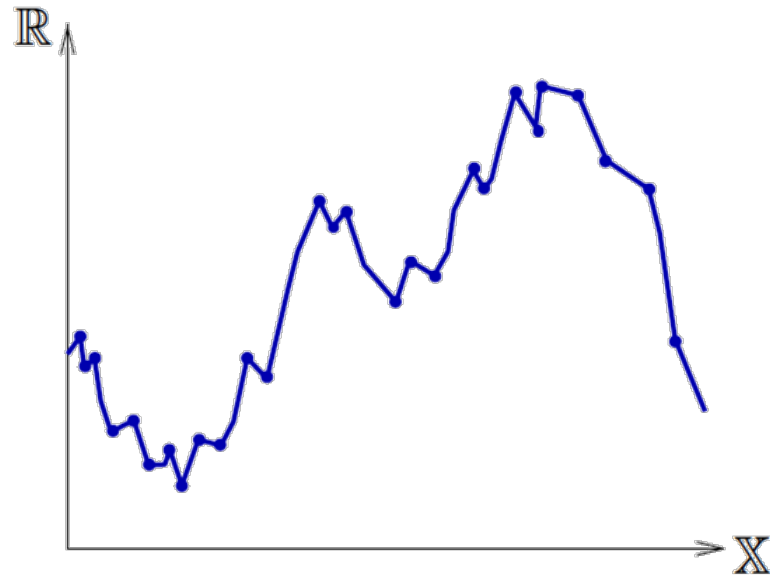
# Persistence-Based Segmentation

# 3D Shape Segmentation

Partition a 3D model into meaningful components



# Computing Segments



Use descriptor functions on shapes

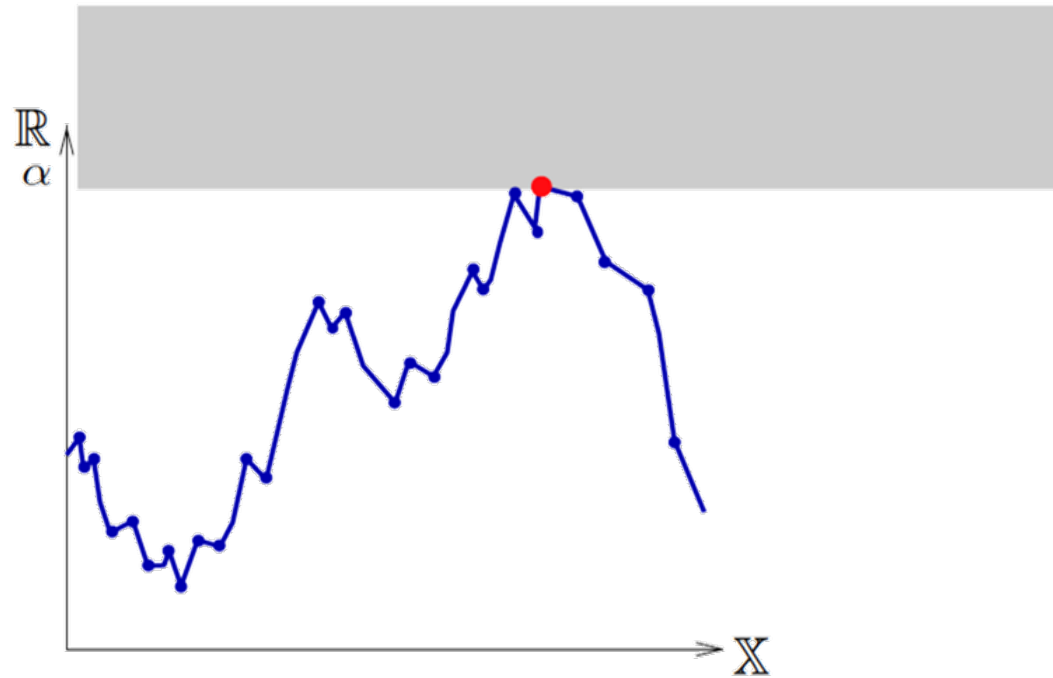


Super-level-set filtration

How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold  $\tau$ !

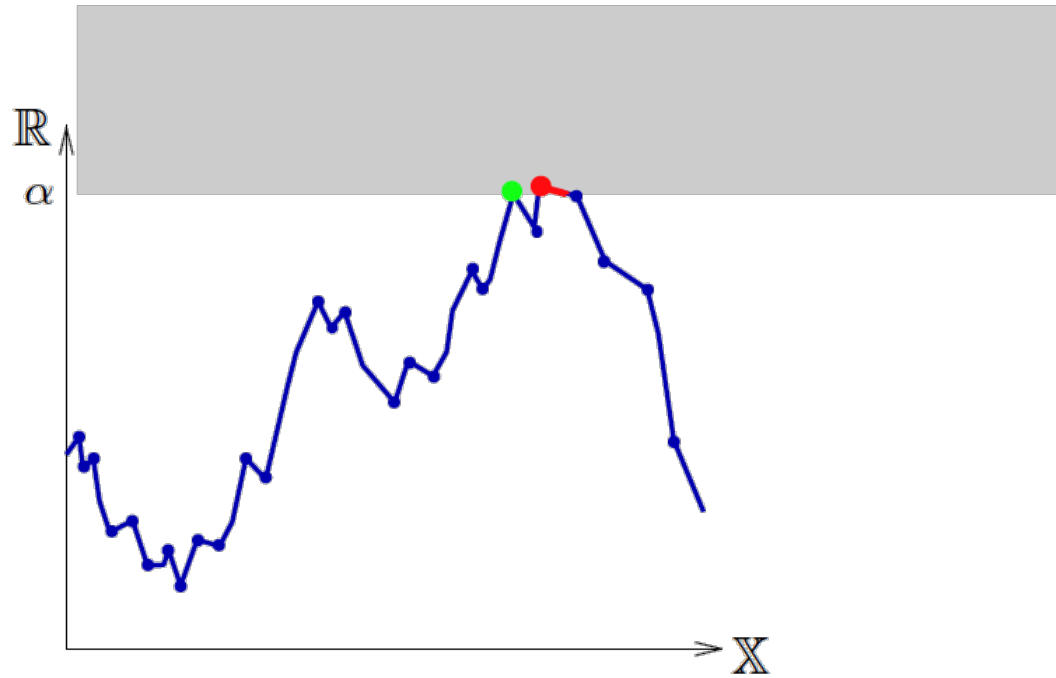
# Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold  $\tau$ !

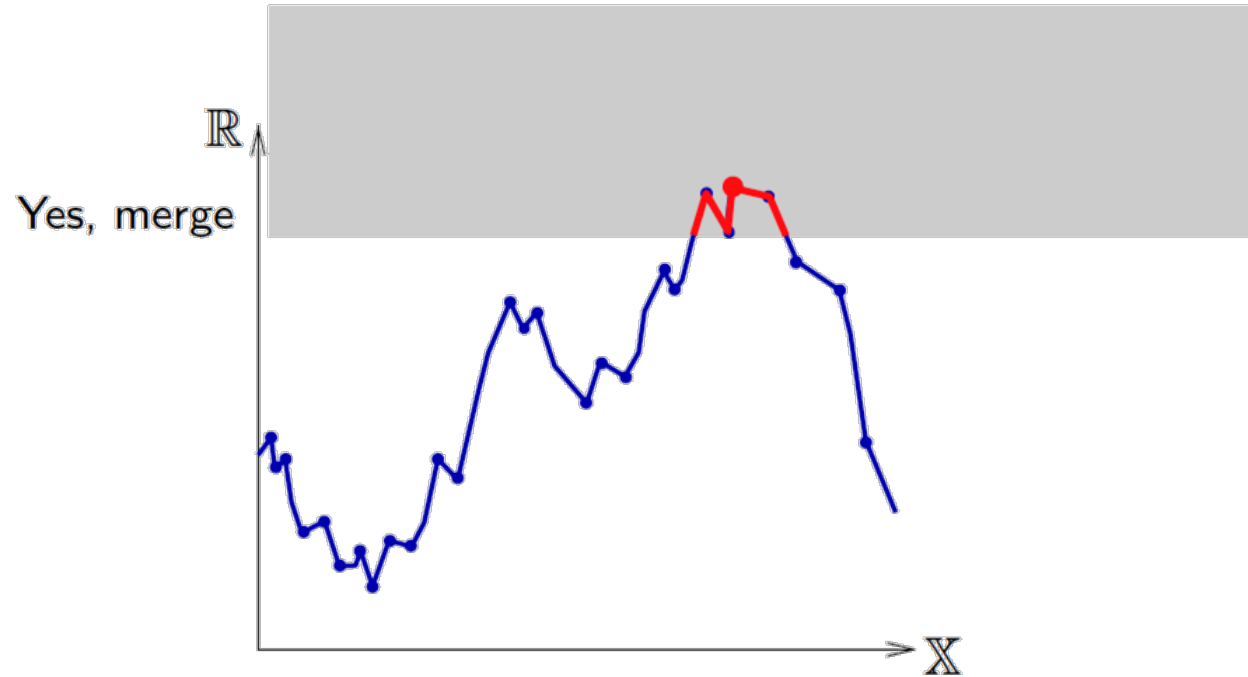
# Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence more than a threshold  $\tau$ !

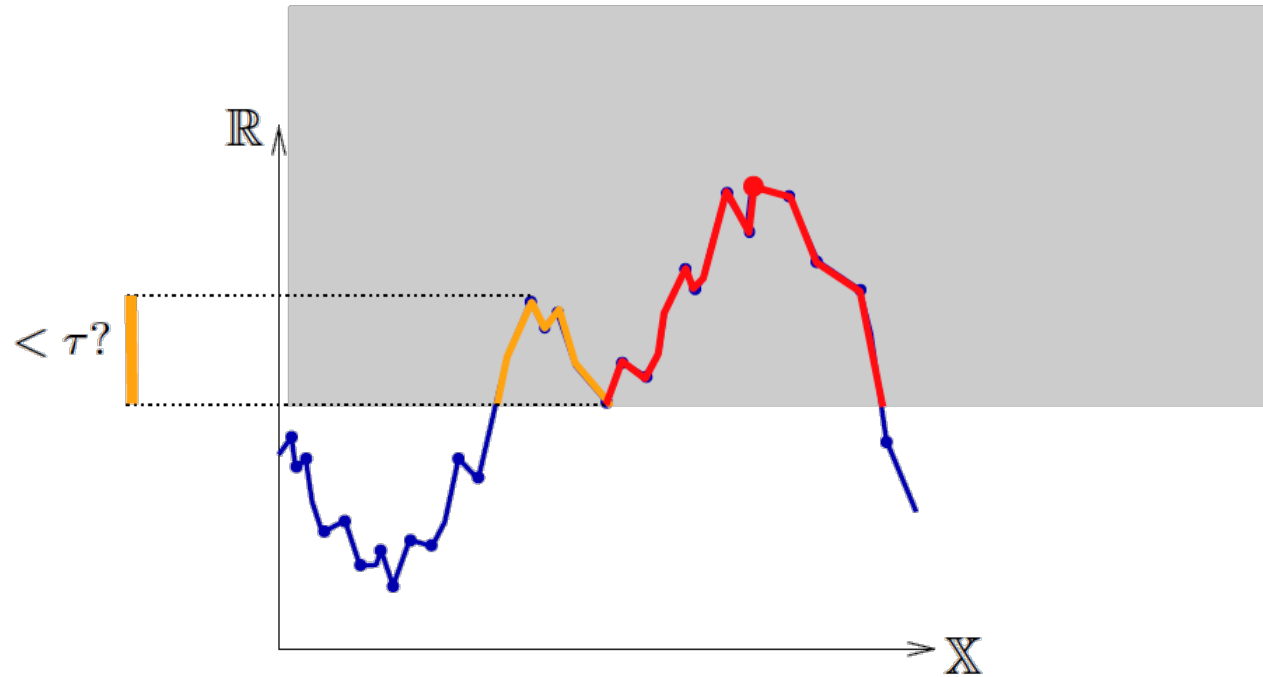
# Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold  $\tau$ !

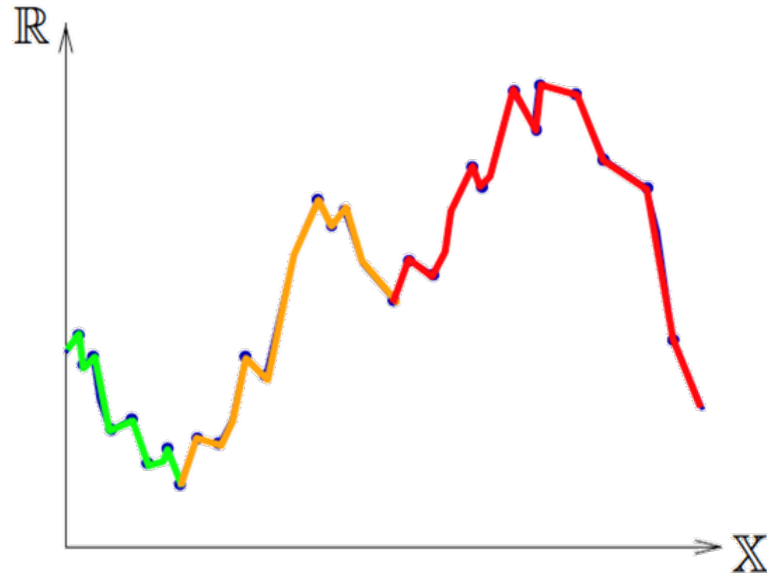
# Computing Segments



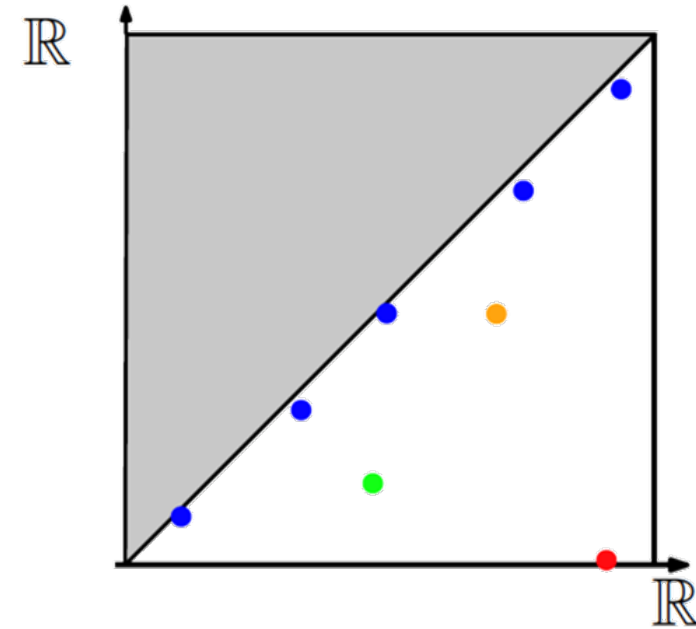
How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold  $\tau$ !

# Computing Segments



How do we compute segments from a PD?



Q: Why the flipped diagram?

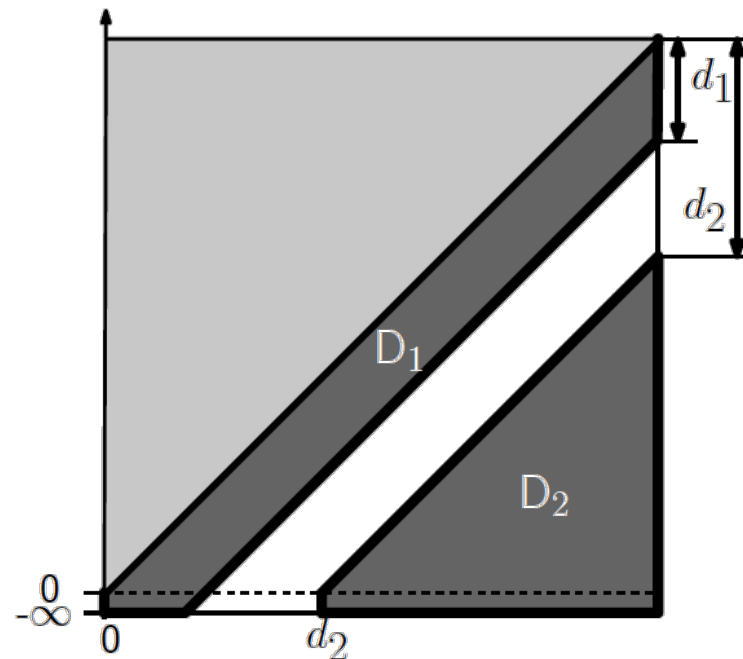
Do not merge segments with persistence greater than a threshold  $\tau$ !

# Algorithm

- Input:  $f(x), \mathcal{M}, \alpha$ 
  1. Sort  $x$  according to  $f$
  2. For  $x \in L$ 
    - 2a. For neighbors of  $x$  in  $\mathcal{M}$   
If no higher neighbors  $\Rightarrow$  new cluster  
else assign  $x$  to  $\nabla f$
    - 2b. For adjacent clusters  $y$  to  $x$   
if  $|f(y) - f(x)| \leq \alpha$   
merge into oldest adjacent cluster

# Interpreting Persistence Diagrams

- If peaks are prominent enough, number of segments is stable
- Theoretically,
  - The number of segments is stable
  - The finer the mesh, the smaller the noise

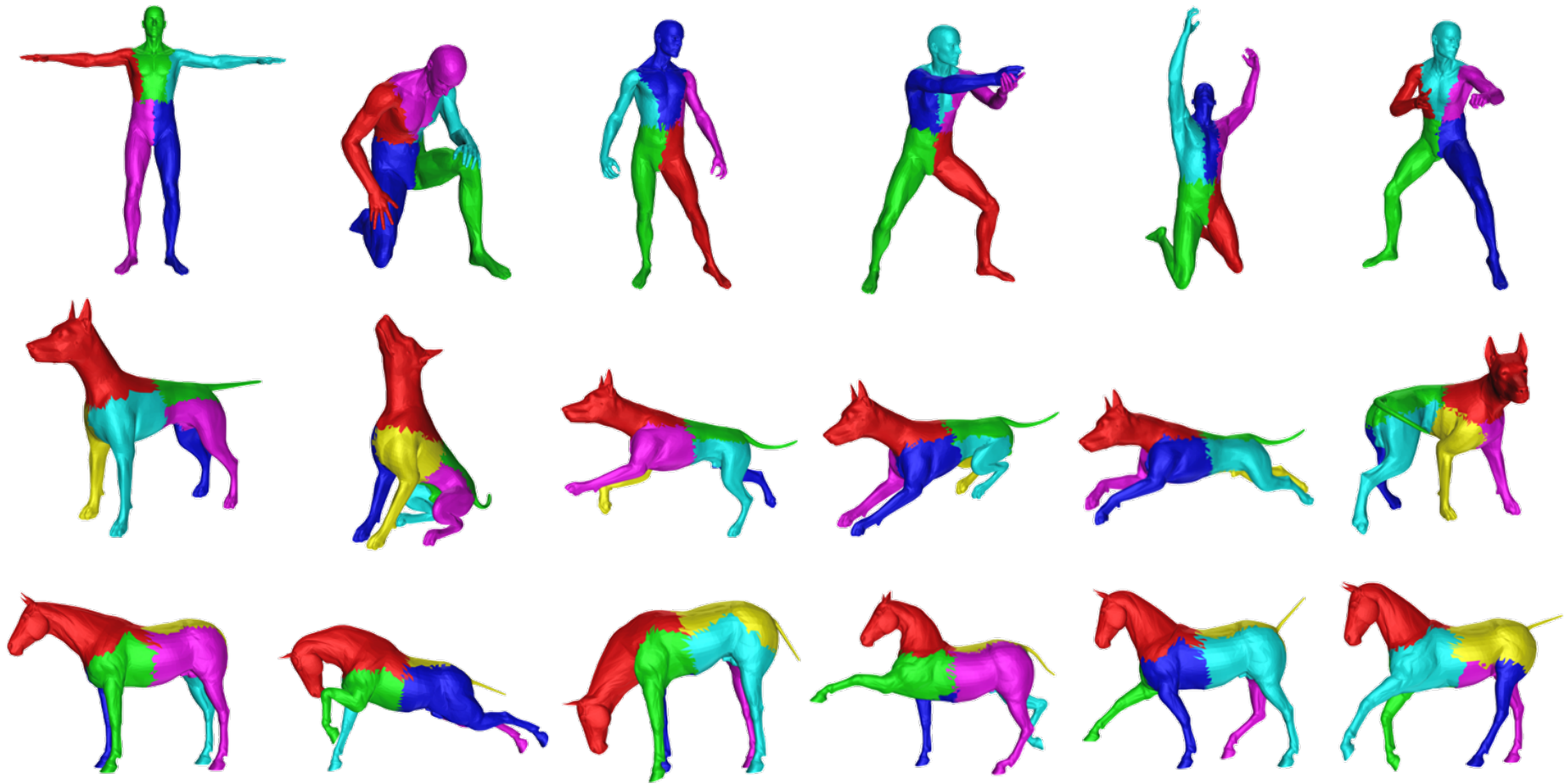


The PD itself can help us decide what the merging threshold should be

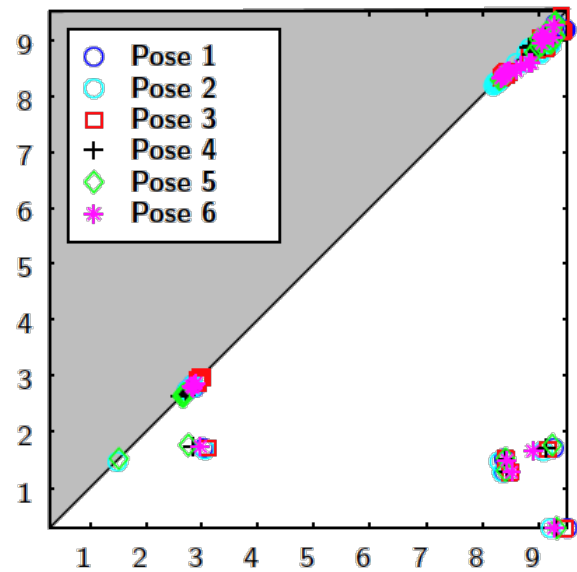
# Choice of Filter Function is Crucial

- Ideal function should be
  - Stable under perturbations
  - Invariant under rigid and isometric deformations
  - Informative: local maxima should correspond to segments
  - Efficiently computable
- Use heat kernel signature (HKS) or wave kernel signature
  - These are functions obtained from solving certain partial differential equations on the surface of a 3D shape
  - More later ...

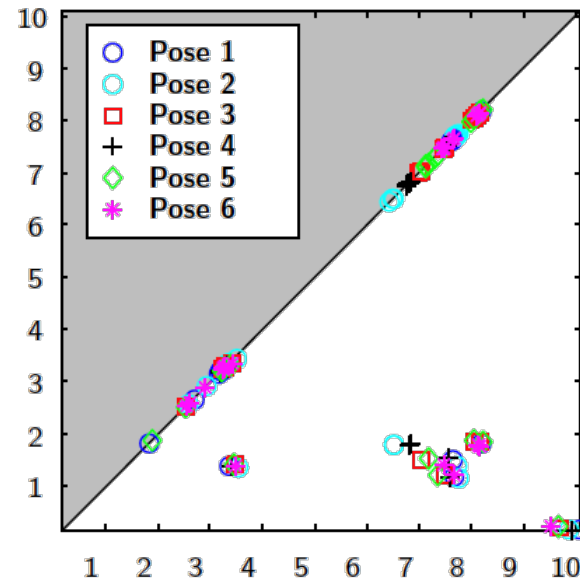
# Segmentations



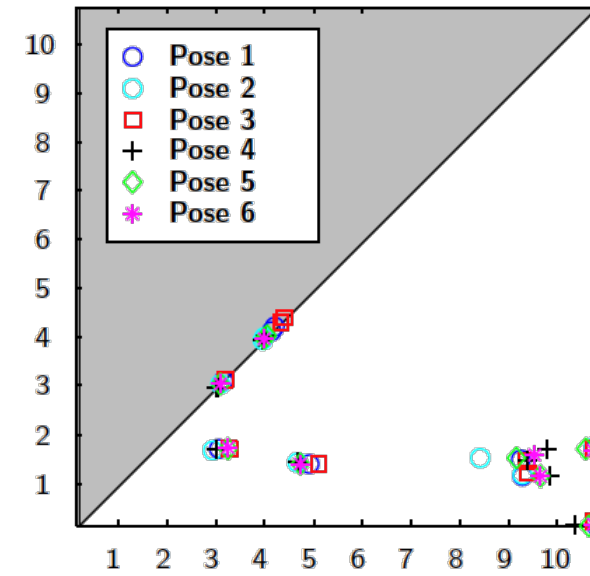
# Stable Diagrams Under Isometric Deformations



Human



Dog



Horse

# Scalar Field Analysis

# Underlying Problem: Characterizing a Landscape from Limited Data

Chazal, Guibas, Oudot, Skraba, *Scalar Field Analysis over Point Cloud Data*. DCG 2011

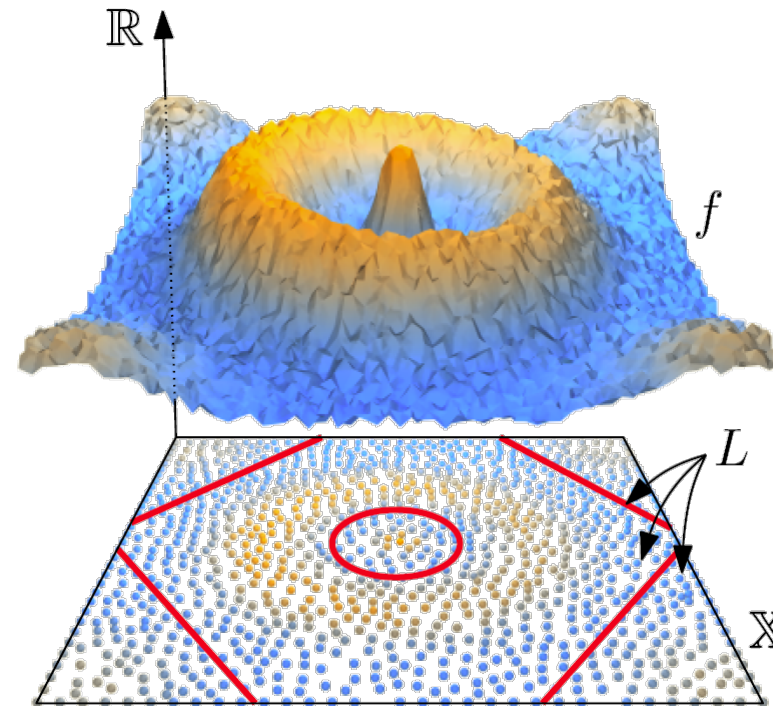
**Setting:** topological space  $\mathbb{X}$ ,  $f : \mathbb{X} \rightarrow \mathbb{R}$

**Input:** a finite sampling  $L$  of  $\mathbb{X}$ , the values of  $f$  at the sample points

- assuming  $f$  is smooth (Lipschitz condition)

**Goal:** Analyze landscape of  $\text{graph}(f)$ :

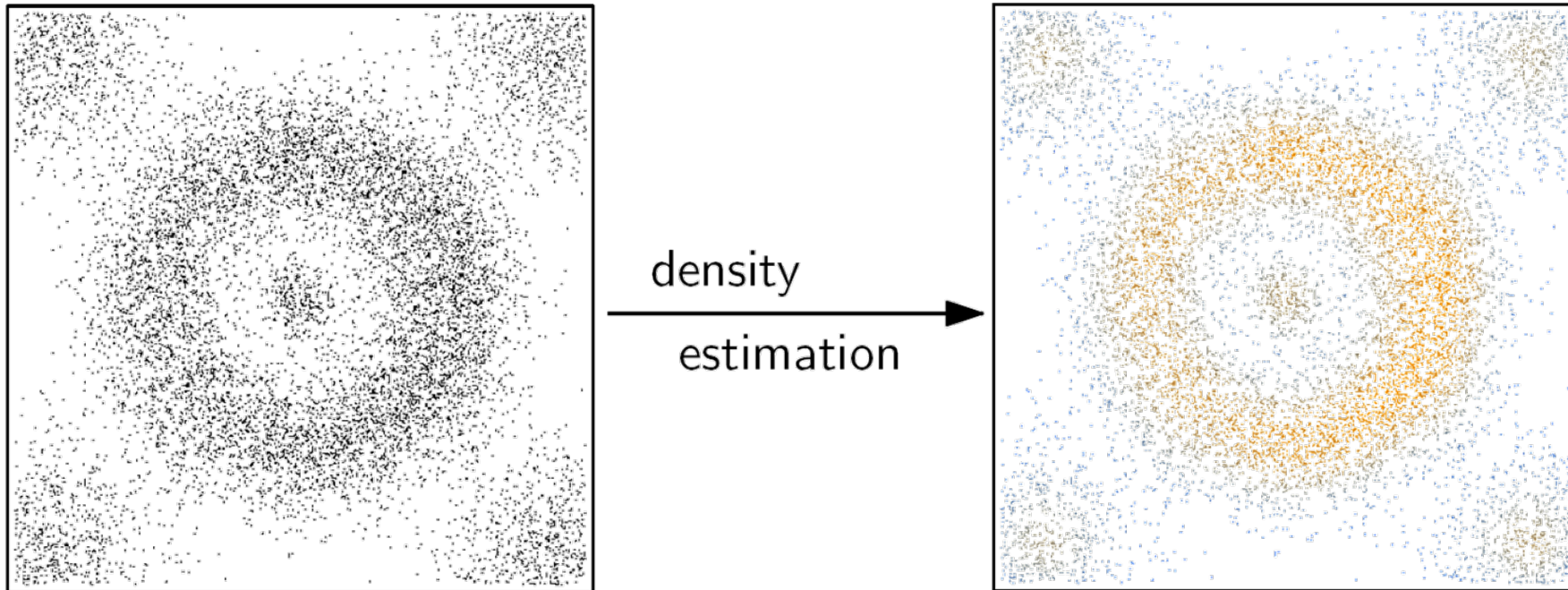
- prominent peaks/valleys
- basins of attraction
- in the presence of noise
- without explicit knowledge of the sample positions



# Motivating Applications

- **unsupervised learning:**

- data points drawn at random from some unknown density distribution  $f$
- approximate  $f$  through some density estimator  $\hat{f}$
- cluster data points according to prominent basins of attraction of  $\hat{f}$

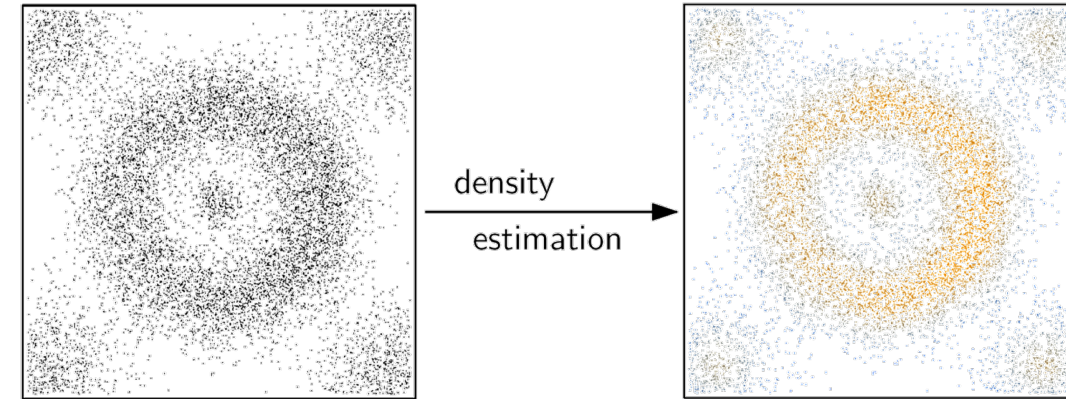


# Motivating Applications

Chazal, Guibas, Oudot, Skraba, *Scalar Field Analysis over Point Cloud Data*. DCG 2011

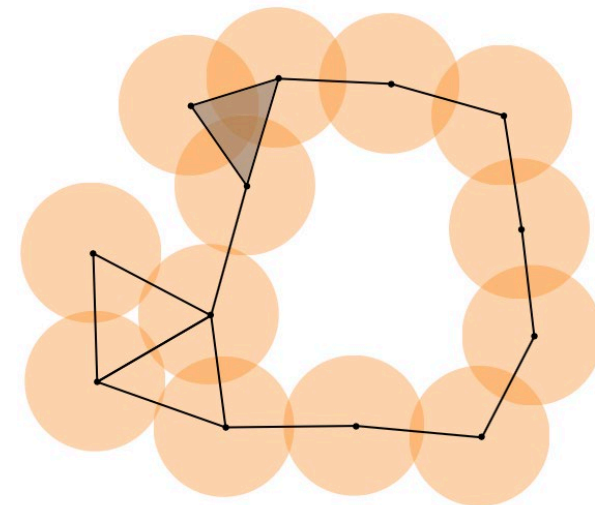
## Unsupervised learning

- Data (and function values) sampled from some unknown underlying distribution  $f$
- Obtain approximation through density estimator  $\hat{f}$
- Characterize peaks, valleys, basins of attraction



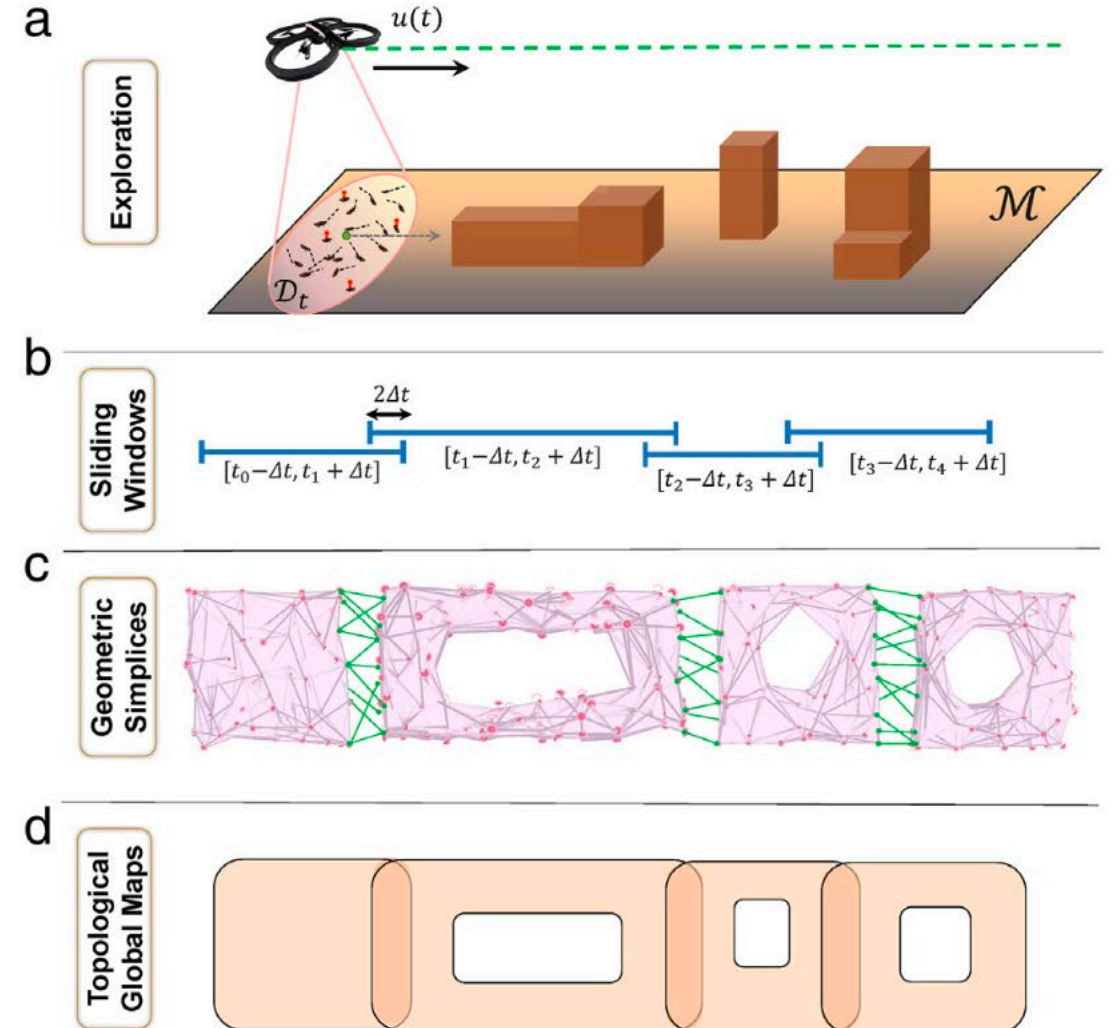
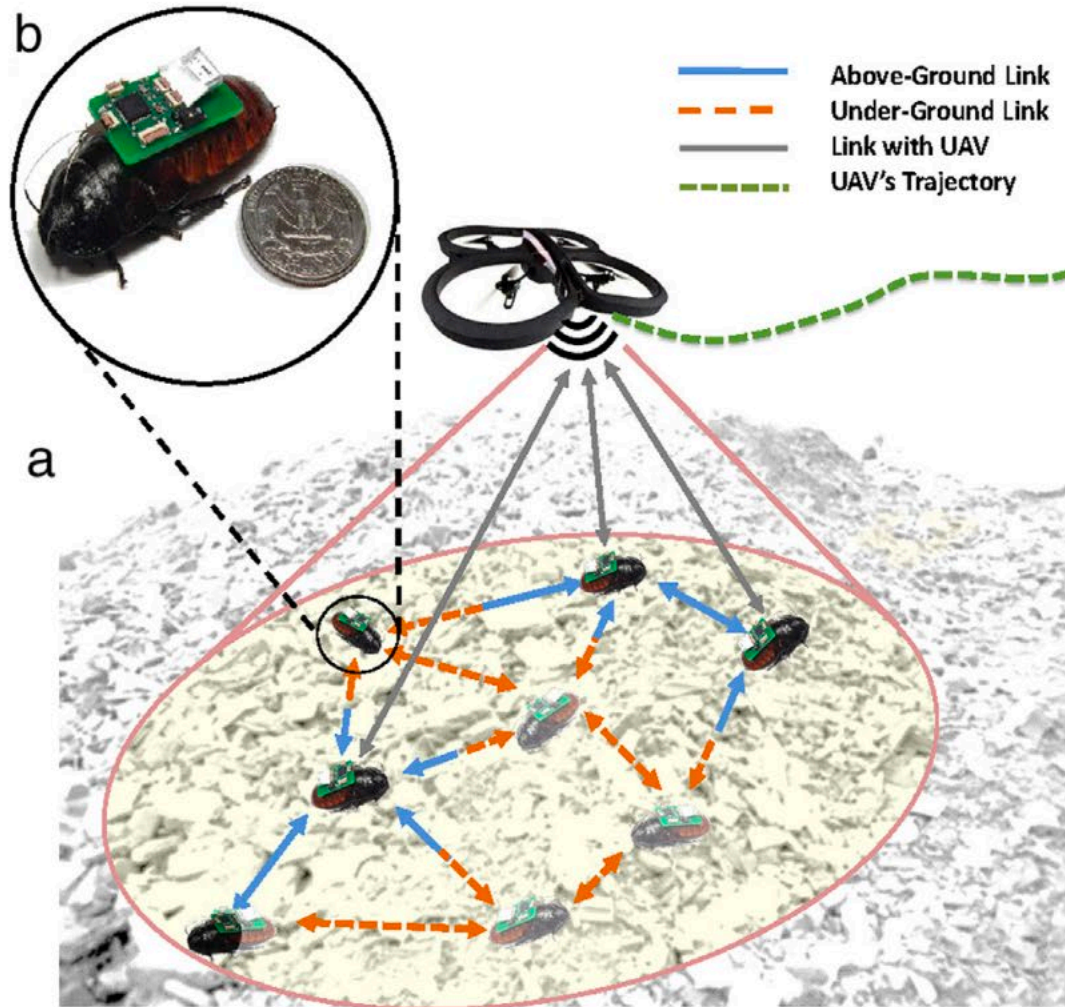
## Sensor networks

- Sensors distributed in some unknown area
- Measure some quantity  $f$
- Communicate with nearby sensors
- **Want:** to understand landscape of  $f$



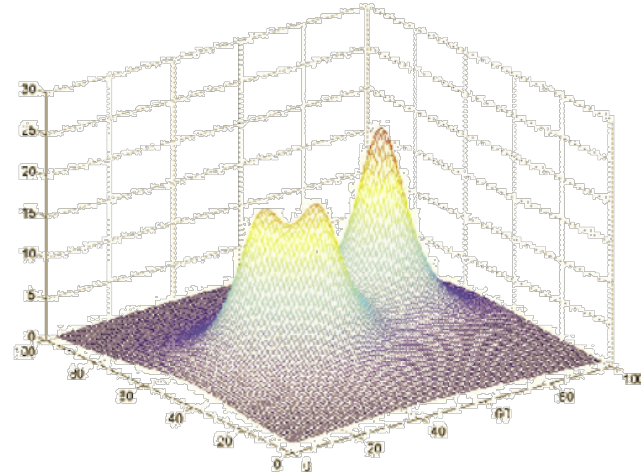
# Mapping Spaces with Cyborg Insects

Dirafzoon, Bozkurt, Lobaton, *A framework for mapping with biobotic insect networks*



# Extant Approaches

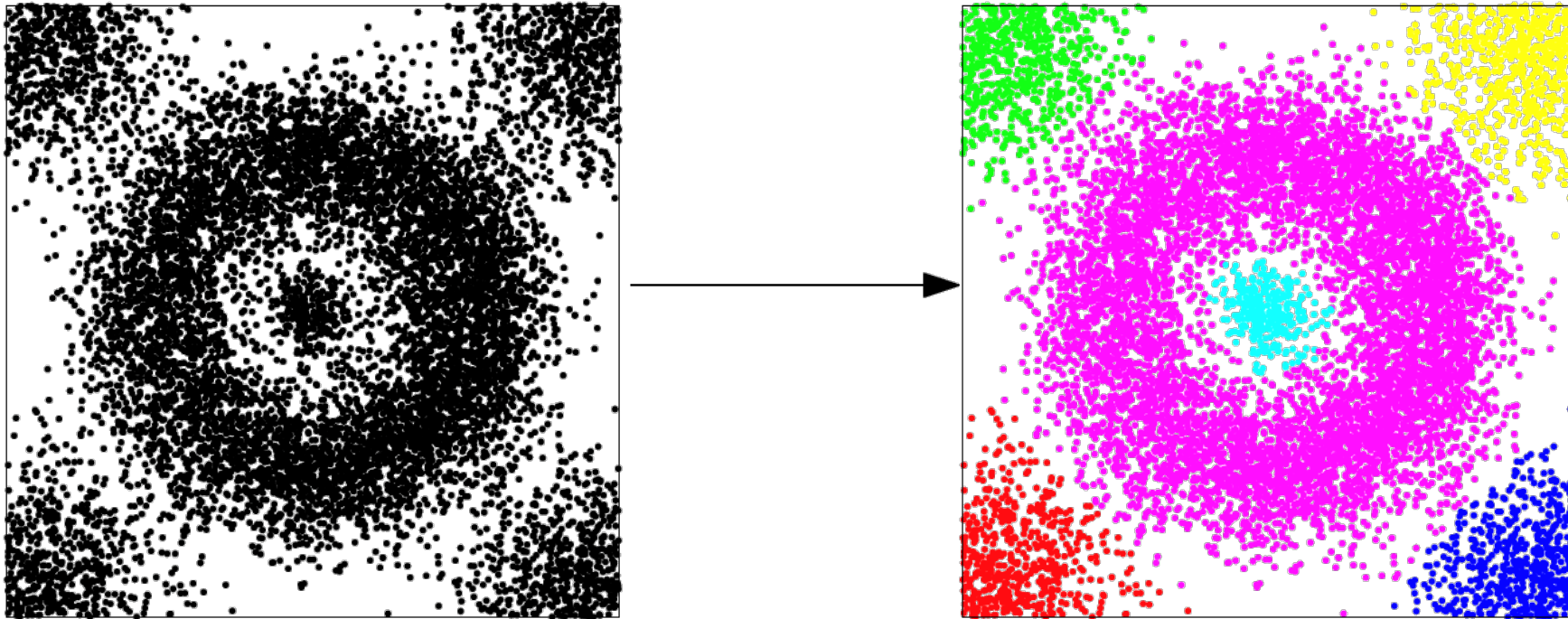
- Classical: when a parametrization of  $\mathbb{X}$  is available, this is a standard function interpolation or regression problem



- Persistence-based: using a triangulation of  $\mathbb{X}$  based on  $L$ , obtained from a parametrization or other means

# Cluster Analysis

**Input:** a finite set of observations: - point cloud with coordinates  
- distance / (dis-)similarity matrix



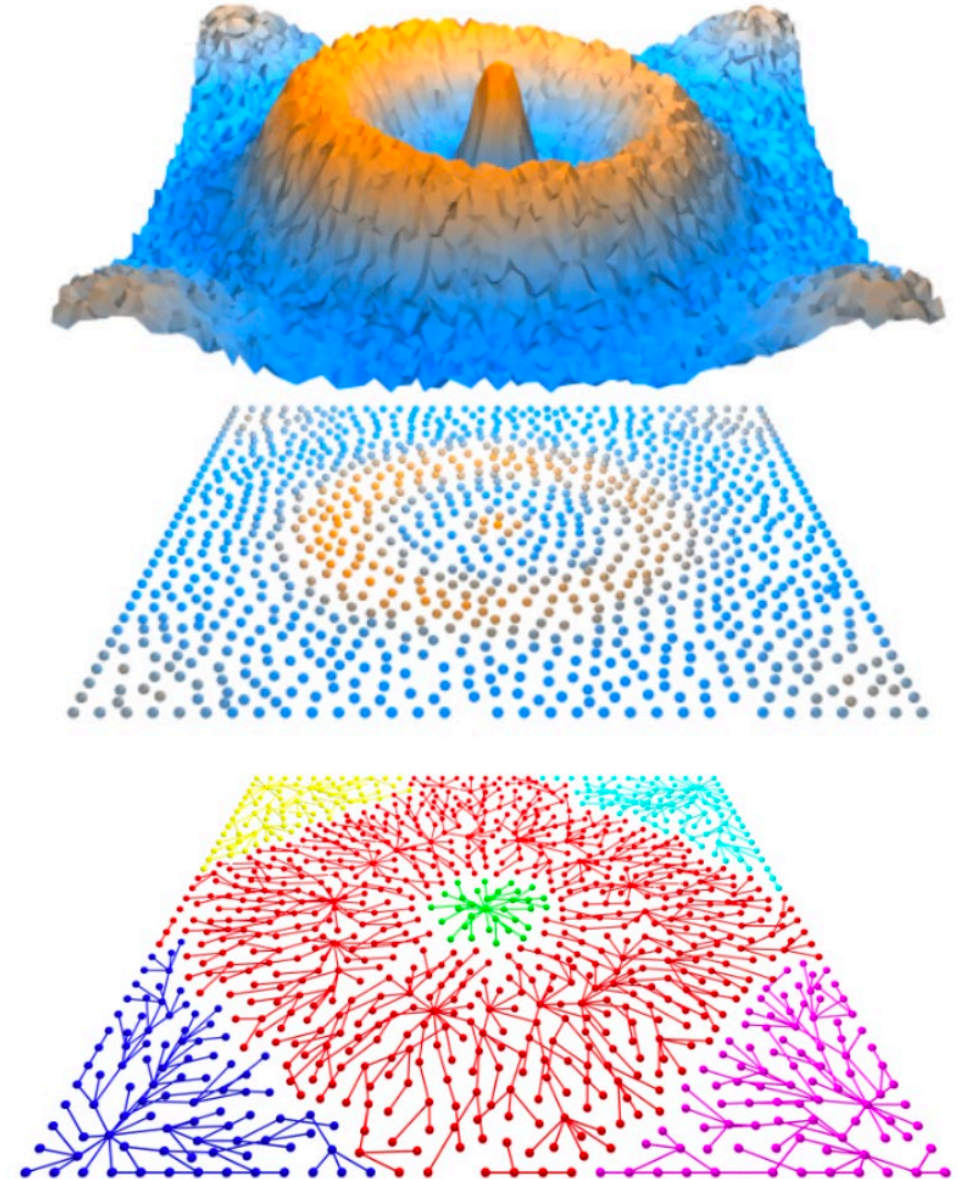
**Task:**

partition the data points into a collection of *relevant* subsets called clusters

# Overview

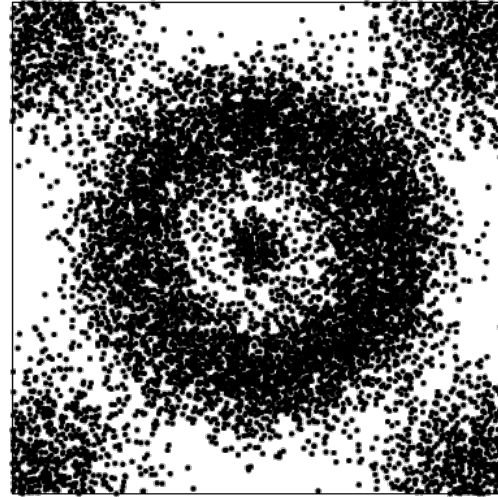
## Mode-seeking paradigm (Mean-Shift and variants)

- Set local peaks of  $\hat{f}$  to be cluster centers
- Partition data according to basins of attraction, i.e. subsets of data that reach a cluster center by some greedy hill-climbing procedure
- Problem: can be highly unstable/unpredictable
- Proactive solution: apply smoothing before hill-climbing (but unclear how this affects output, e.g. number of clusters)
- *Reactive* solution: Use **topological persistence** to detect and merge clusters *after* computation

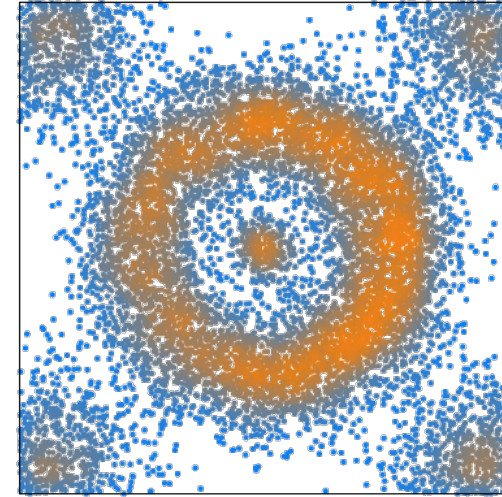


# Mean-Shift and Variants

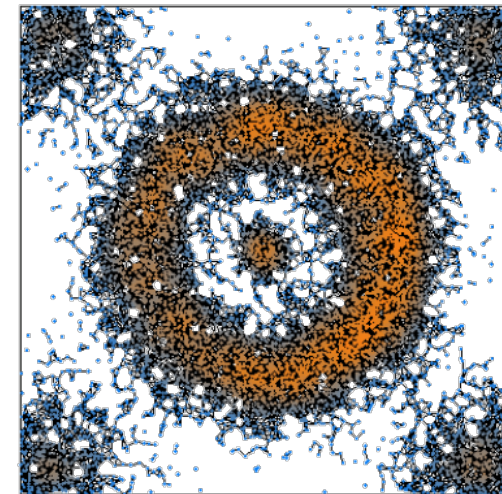
- Estimate density  $f$
- Build neighborhood graph (e.g. via Vietoris-Rips at scale  $\delta$ )
- Approximate gradient: at each vertex  $v$ , either connect  $v$  to neighbor with highest  $f$  value or declare  $v$  to be a peak
- Resulting collection of edges forms a spanning forest of the neighborhood graph



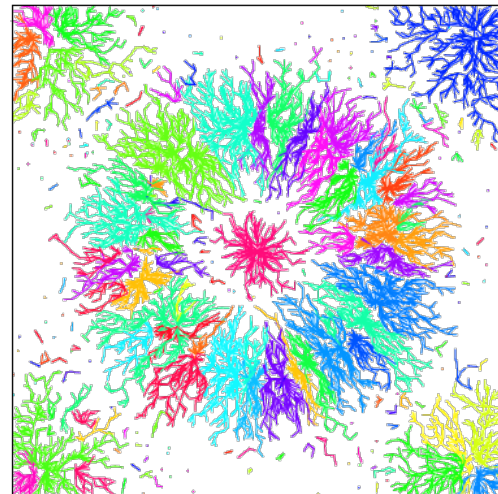
estimate density  
at the data points



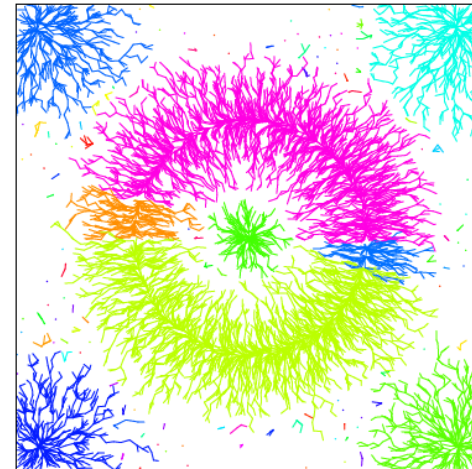
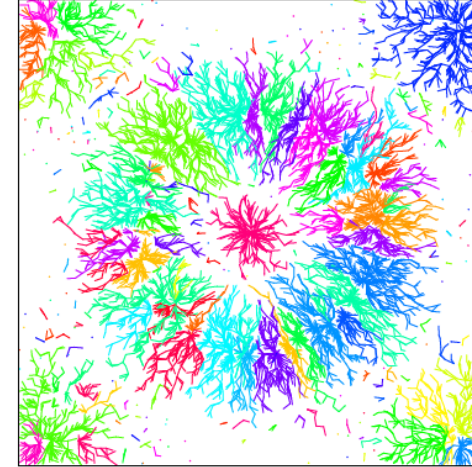
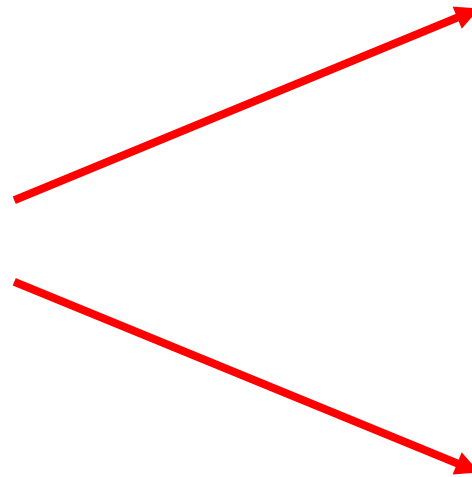
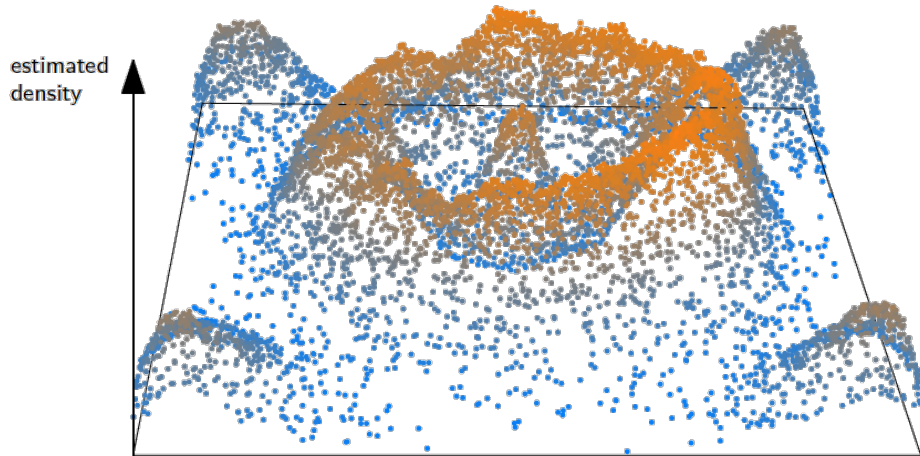
build neighborhood graph



approximate gradient  
by a graph edge  
at each data point



# Instability in Density Estimate can Cause Unstable Output



Sources of instability:

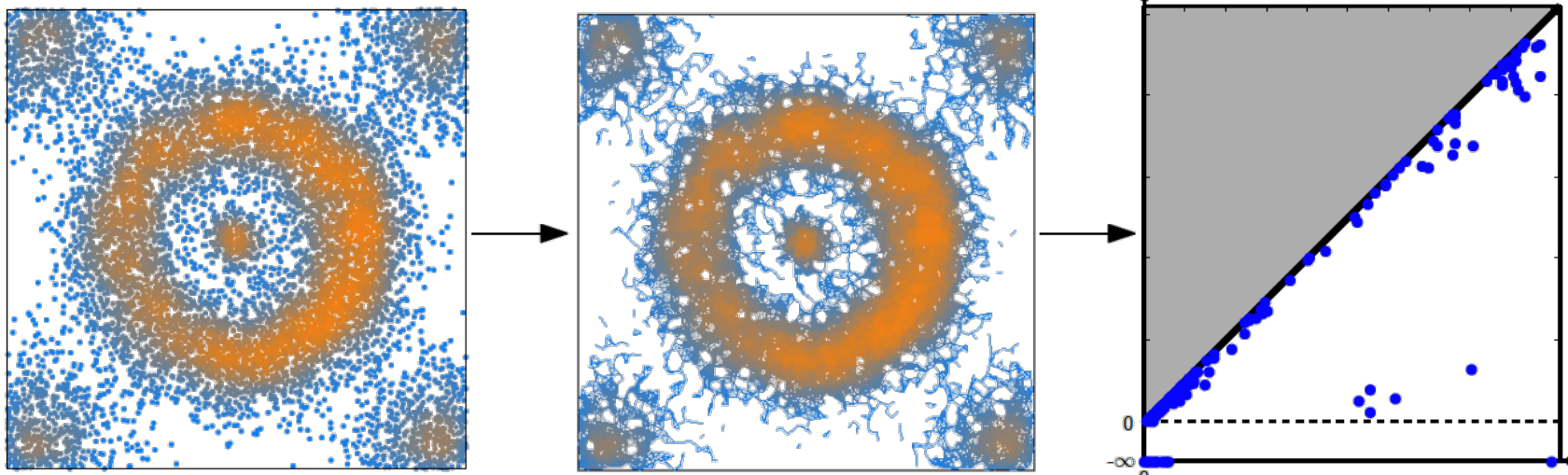
- Variations (noise) in density estimate
- Bad neighborhood graph

# Topological Mode Analysis Tool (ToMATo)

Chazal, Oudot, Skraba, Guibas *Persistence-Based Clustering in Riemannian Manifolds*

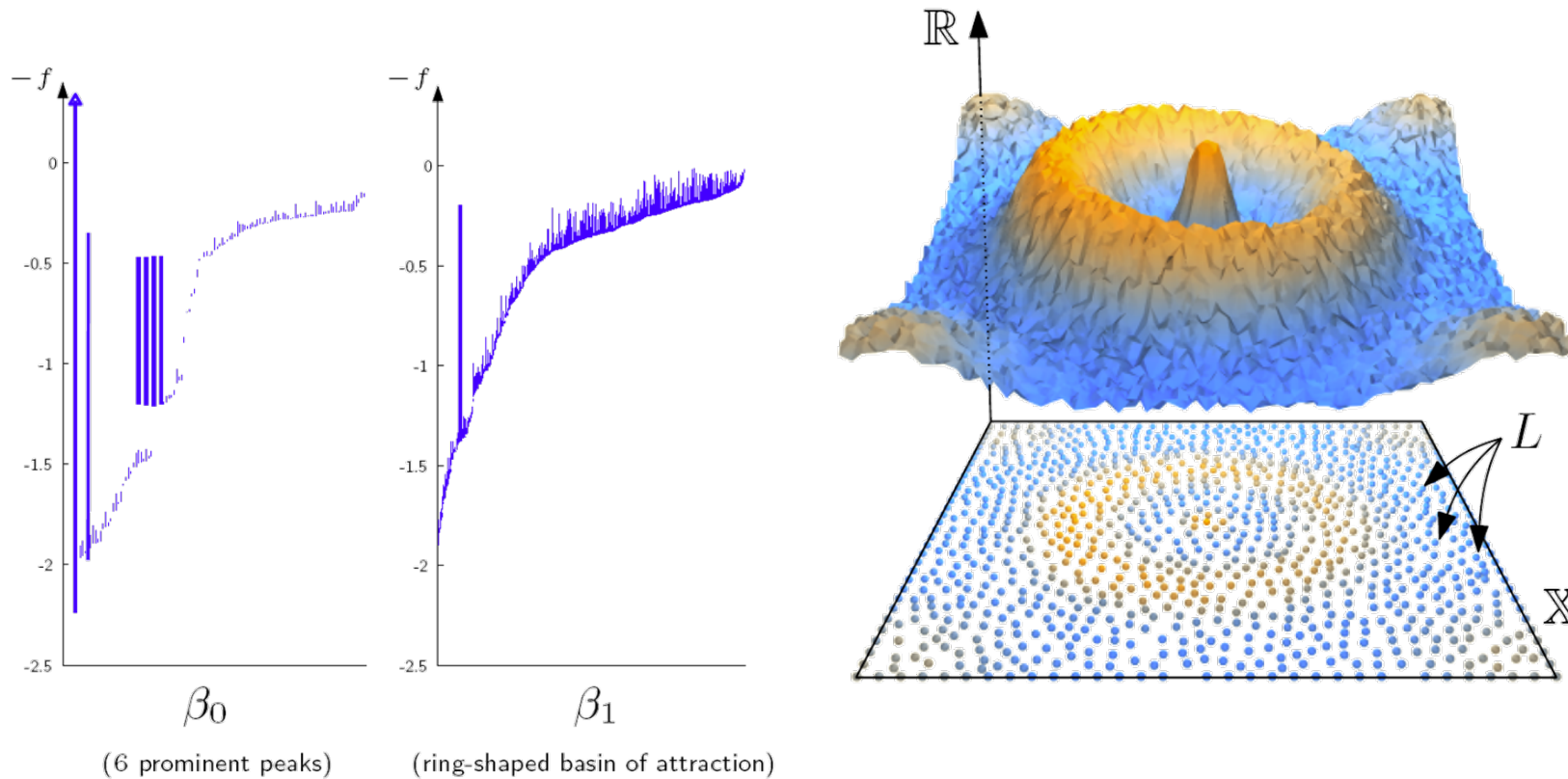
- Density estimator  $\hat{f}$  defines an order on the point cloud  
(sort data points by **decreasing** estimated density values)
- Extend order to the graph edges  $\rightarrow$  *upper-star filtration*  
( $\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\}$ )
- Compute the 0-dimensional persistence diagram of this filtration  
(apply 0-dimensional persistence algorithm  $\rightarrow$  union-find data structure)

0-d persistence can be computed via Kruskal's minimum spanning tree algorithm + union-find data structure for  $O(\alpha(n) \cdot n^2)$  complexity, where  $\alpha(n)$  is an inverse Ackermann function

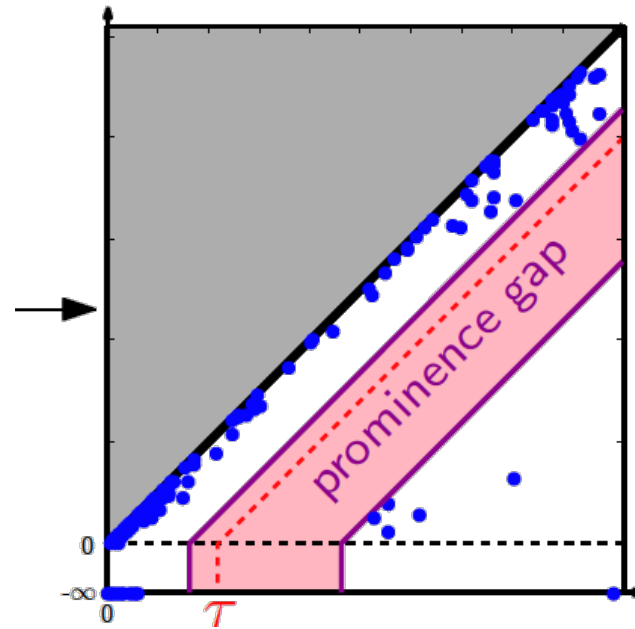
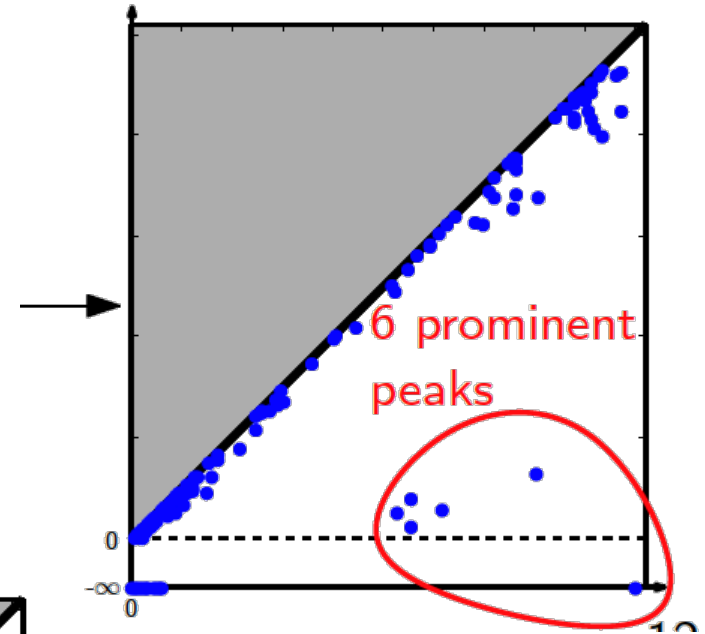
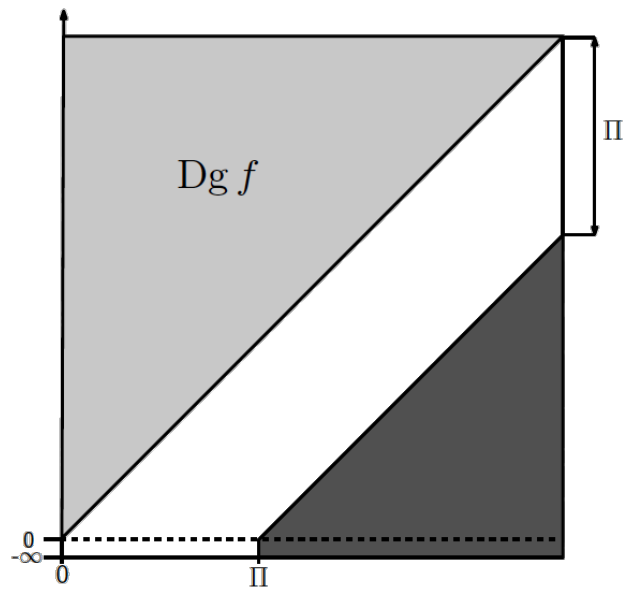


# Clustering Example A

**Assumptions:**  $\mathbb{X}$  Riemannian manifold,  $f : \mathbb{X} \rightarrow \mathbb{R}$   $c$ -Lipschitz,  
 $L$  geodesic  $\varepsilon$ -cover of  $\mathbb{X}$ , for some unknown  $\varepsilon > 0$ .



# Estimating the Merge Parameter

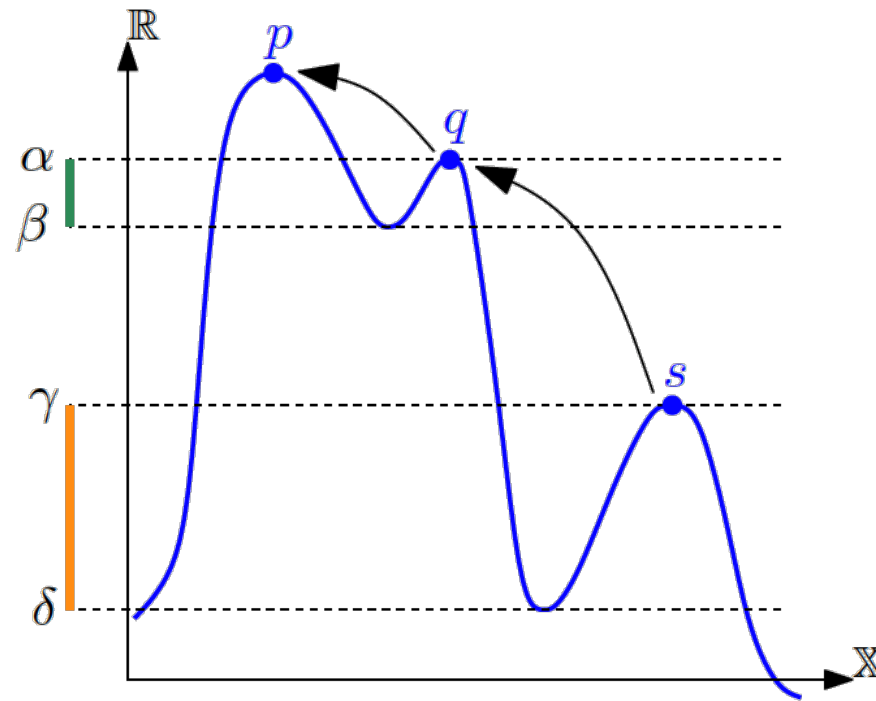


The gap parameter  $\tau$

# Merging Clusters

- 0-dimensional persistence builds a hierarchy of the peaks of  $\hat{f}$  (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)
- given a fixed threshold  $\tau \geq 0$ , only merge those clusters of prominence  $< \tau$

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



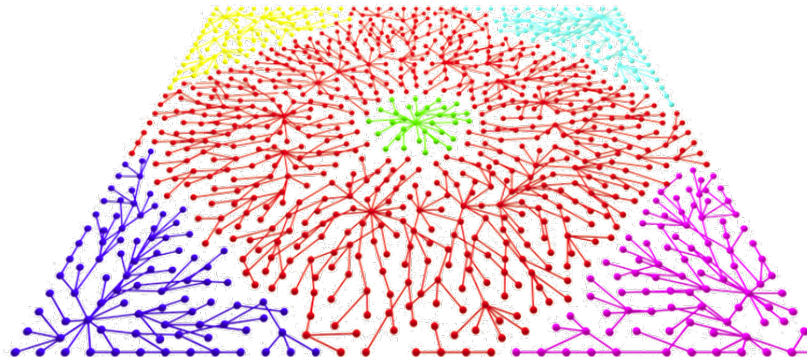
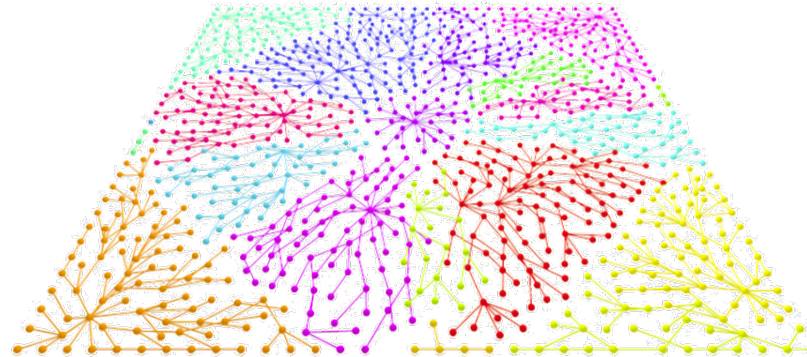
# Basins of Attraction for A

**Goal:** approximate basins of attraction of significant peaks of  $f$   
 $\Rightarrow$  segmentation/clustering of point cloud  $L$

## Approach:

- rough approximation of gradient of  $f$  within Rips graph,
- merge clusters according to 0-dimensional barcode.

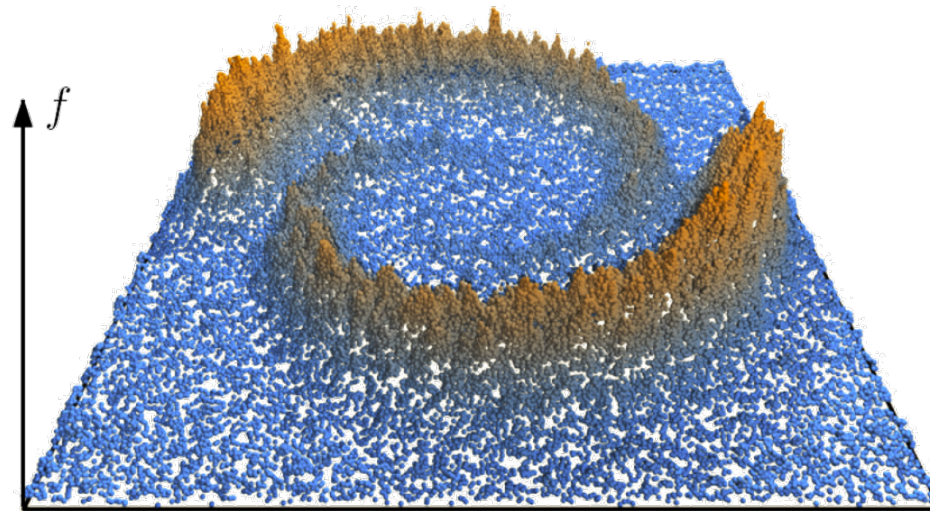
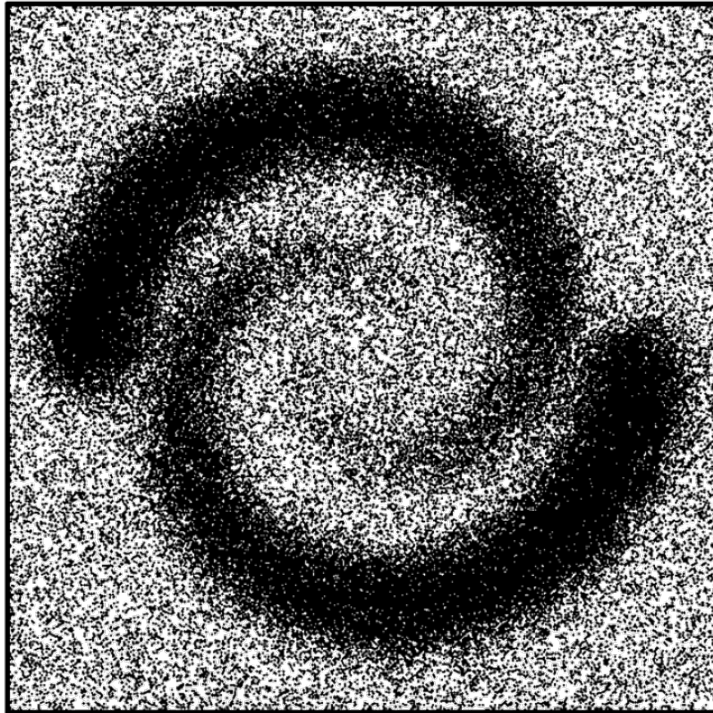
$\rightarrow$  union-find data structure



# Clustering B – The Rips Parameter $\delta$

**Input:**  $\mathbb{X} = [0, 1]^2$ ;  $|L| = 100,000$ ;

$f = \# \{ \text{data pts in fixed-radius ball} \}$

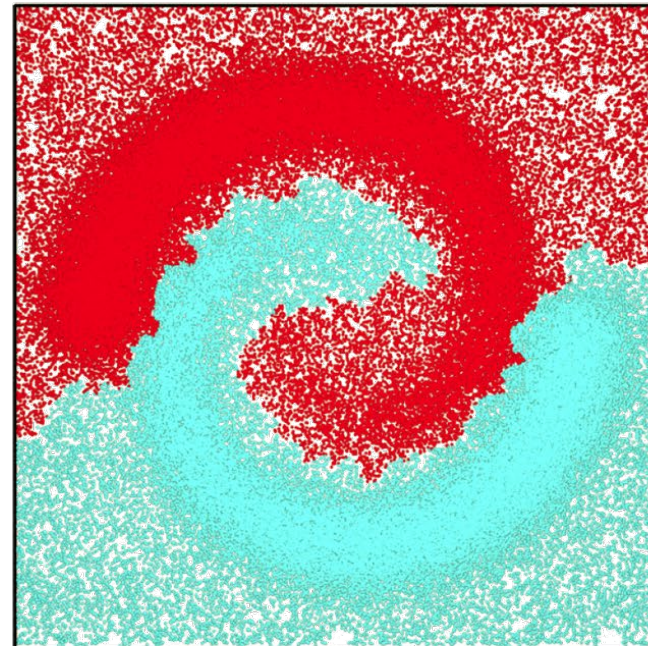
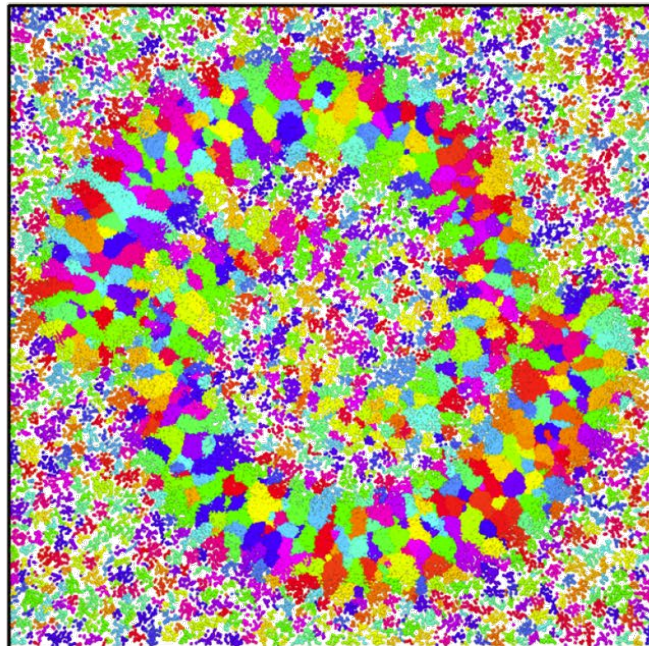
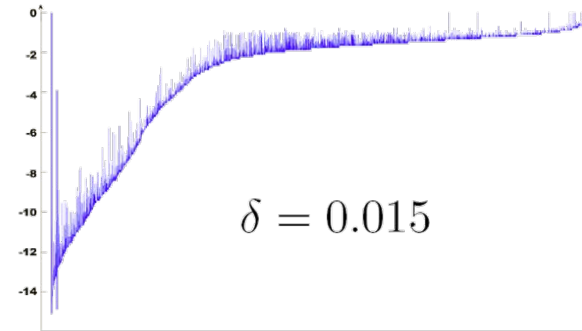


# Clustering B

## Clustering B

**Input:**  $\mathbb{X} = [0, 1]^2$ ;  $|L| = 100,000$ ;

$f = \# \{ \text{data pts in fixed-radius ball} \}$

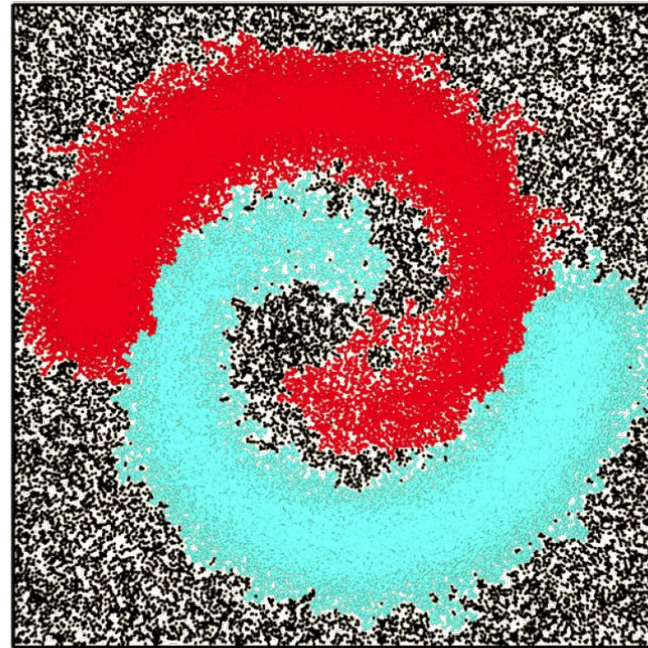
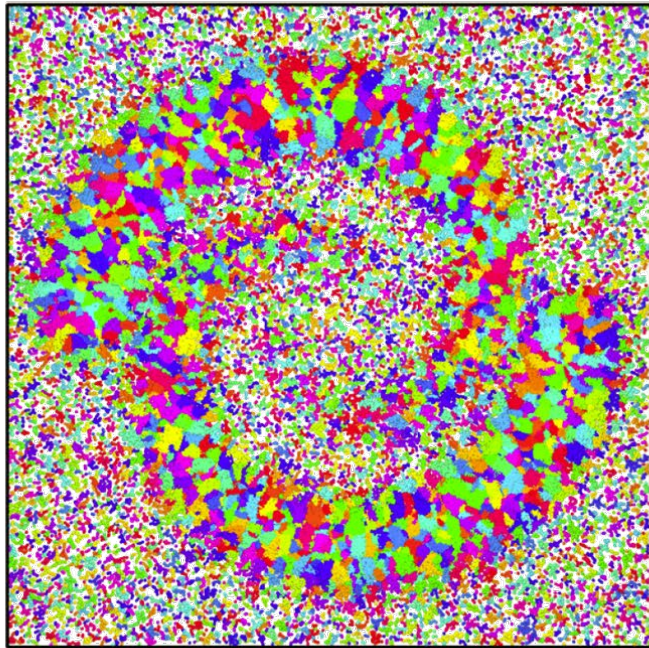
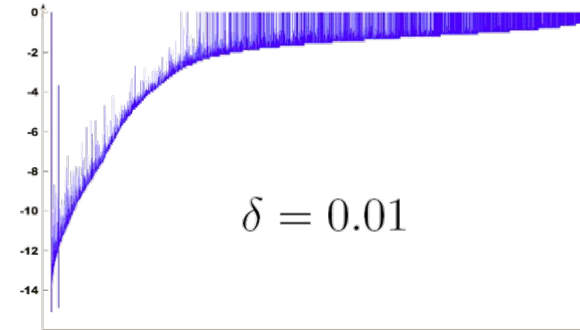


# Clustering B

## Clustering B

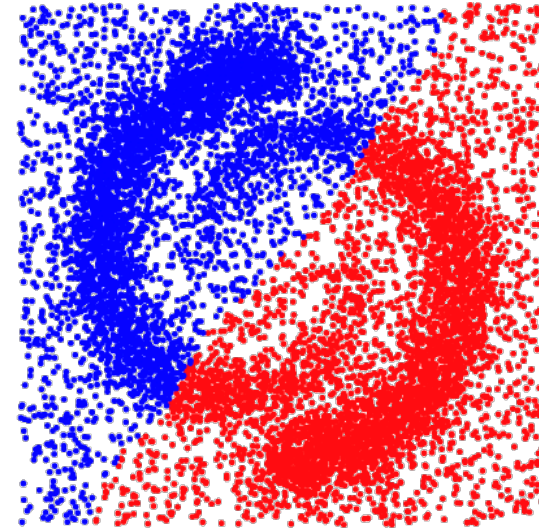
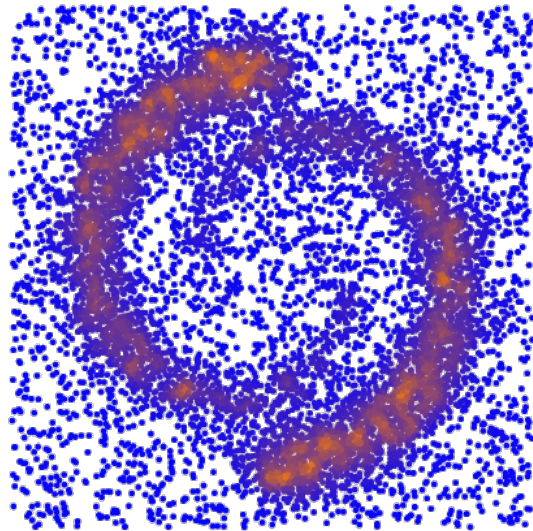
**Input:**  $\mathbb{X} = [0, 1]^2$ ;  $|L| = 100,000$ ;

$f = \# \{ \text{data pts in fixed-radius ball} \}$

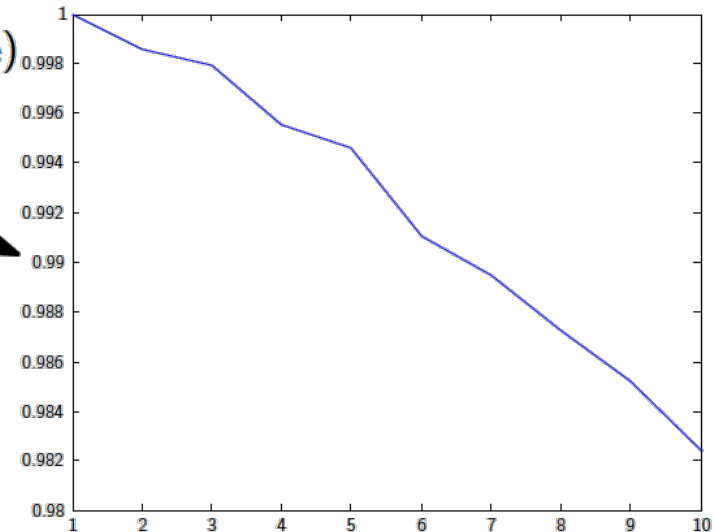


# FYI, Spectral Clustering

Synthetic Data



Spectral clustering  
( $k$ -means in eigenspace)



# Another Hard Example

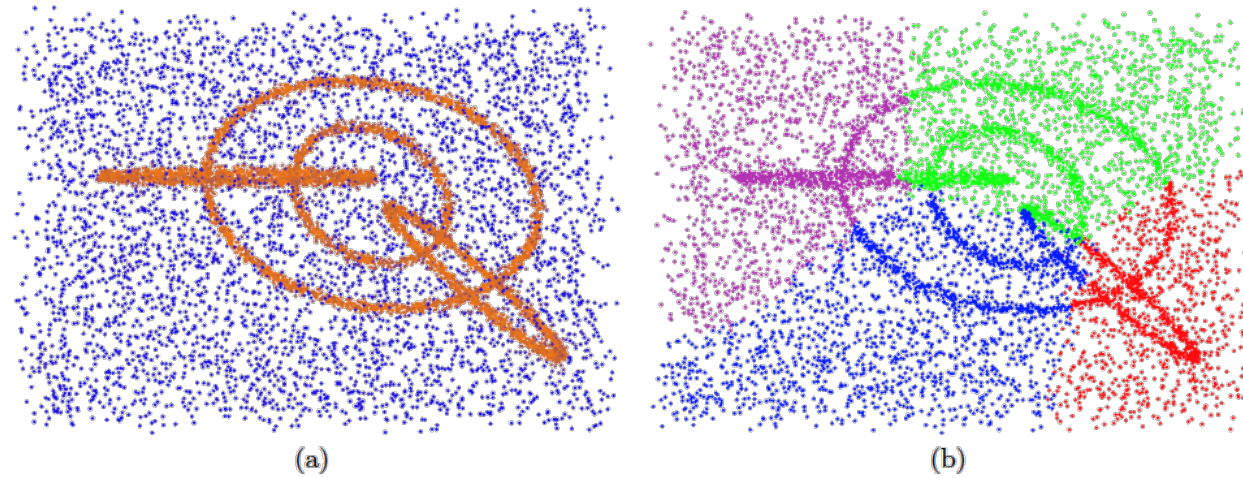


Figure 7: (a) The rings data set with the estimated density function. (b) The result obtained using spectral clustering.

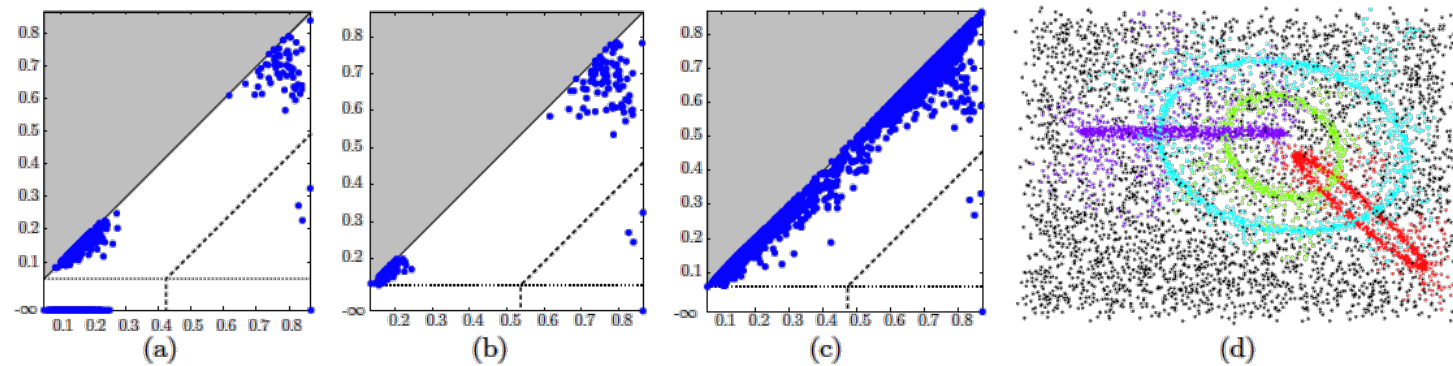
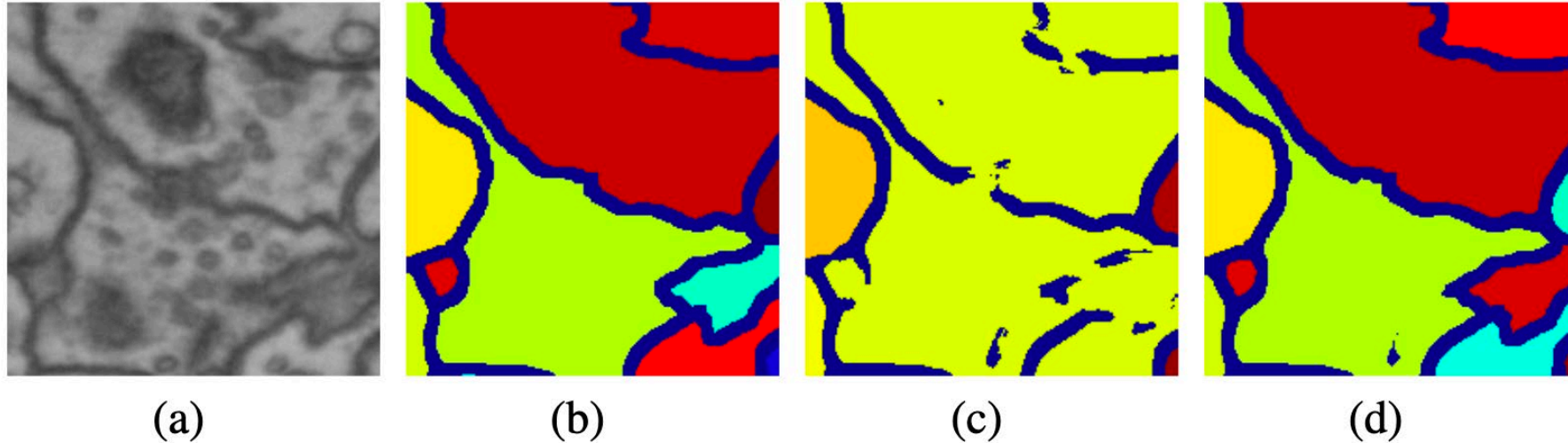


Figure 8: Outputs of ToMATo on the rings data set: the obtained PD with (a)  $\delta$ -Rips graph, (b)  $k$ -nn graph, and (c) Delaunay graph. (d) Clustering obtained with the  $\delta$ -Rips graph.

# Topological Regularization for Deep Learning Applications

# Topology for Image Segmentation

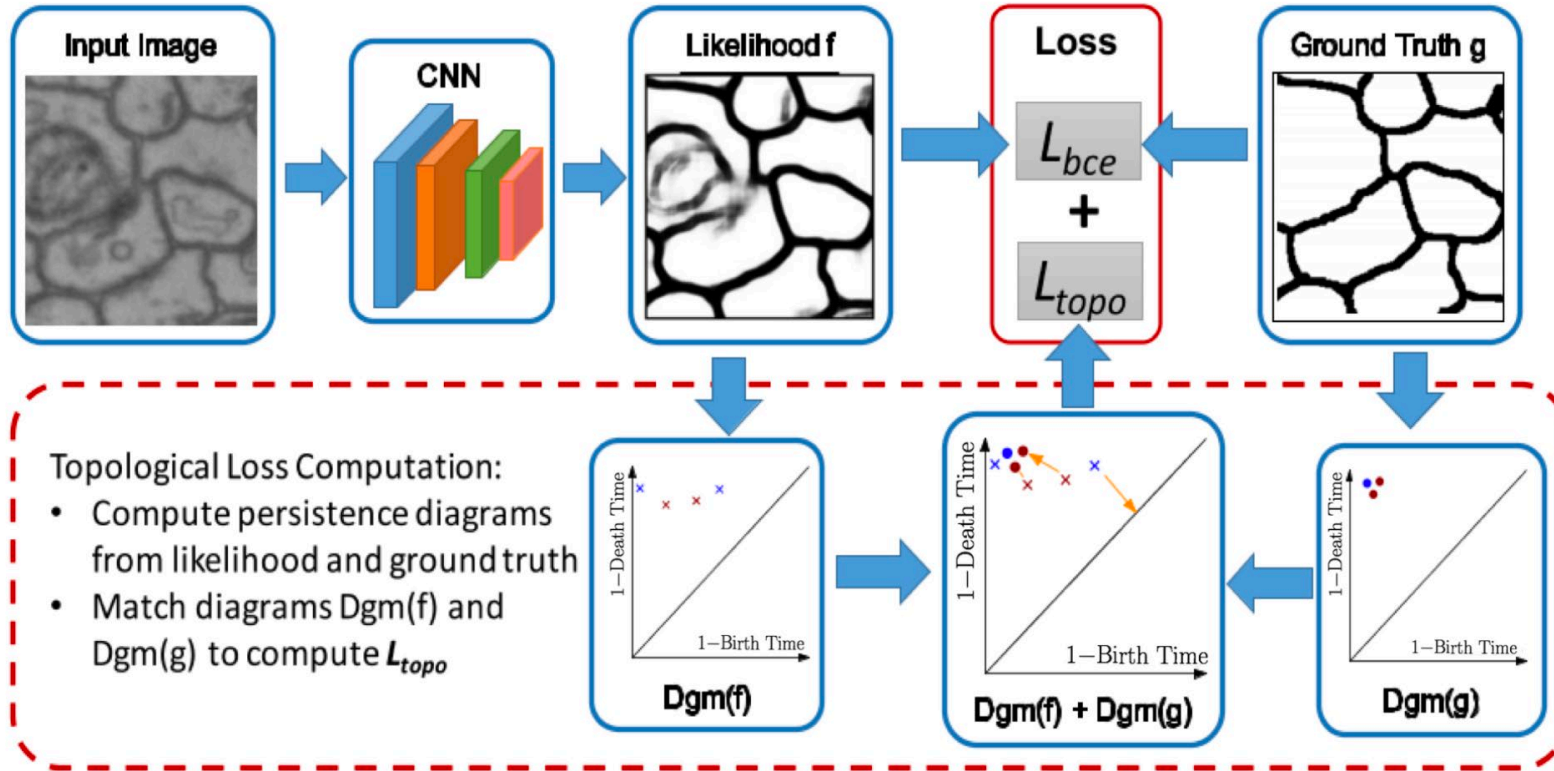
Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeuRIPS 2019*



- ◆ Segmentation errors in fine structures (e.g. membranes and vessels in biomedical imaging) introduce marginal per-pixel error, but major functional error
- ◆ Figure above: (a) input neuron image, (b) ground-truth segmentation, (c) baseline method without topological guarantees, (d) TopoLoss reconstruction
- ◆ Why *persistent* homology was needed: Betti numbers are discrete, but need a continuous, differentiable function for backprop---provided by persistent homology

# Pipeline

Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeuRIPS 2019*

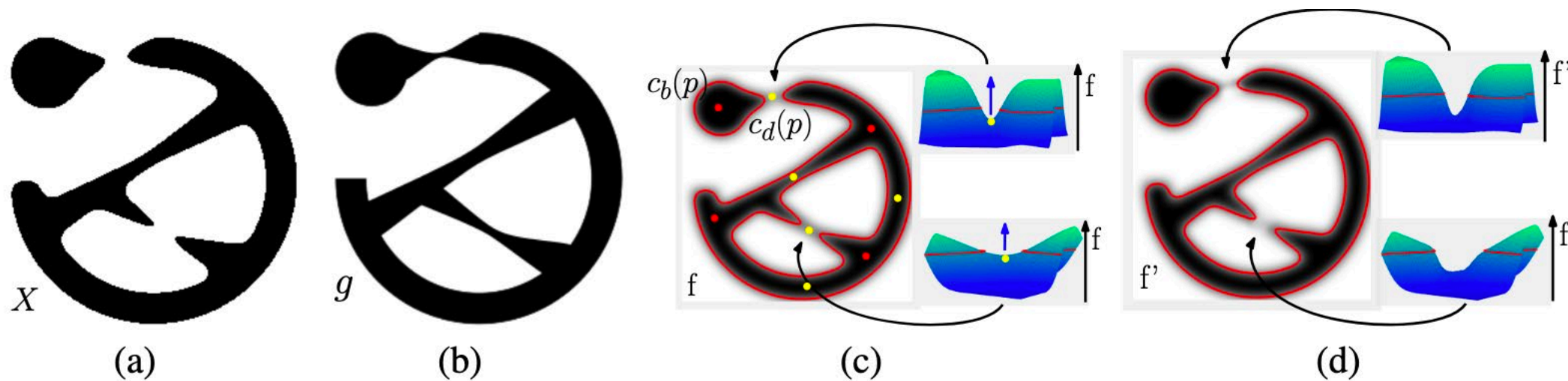


$f$  the likelihood map predicted by the network,  $g$  the ground truth,  $L_{ce}$  the cross-entropy loss,  $L_{topo}$  the topological loss,  $\lambda$  a regularization parameter. Loss function:

$$L(f, g) = L_{ce}(f, g) + \lambda L_{topo}(f, g)$$

# Super-Level Set Filtration on Likelihood

Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeuRIPS 2019*

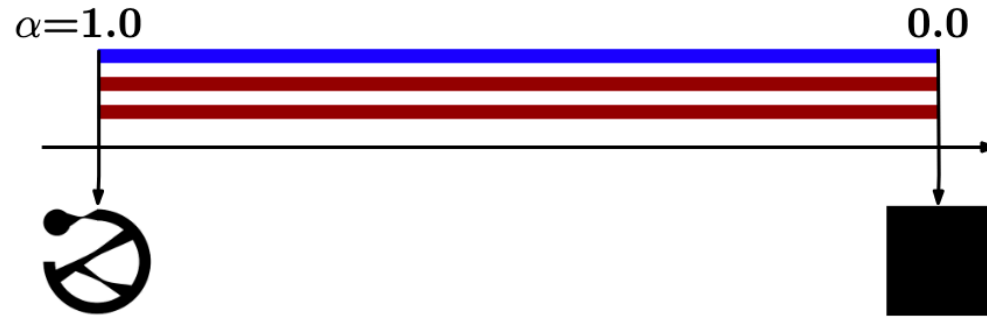


Note: likelihood maps are valued in  $[0,1]$

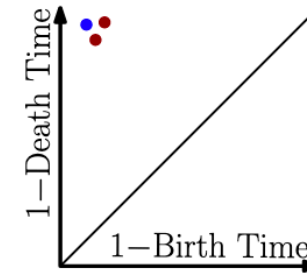
- (a) example segmentation  $X$  with one loop and two connected components
- (b) ground truth segmentation with two loops and one connected component---view  $g$  as binary  $\{0,1\}$ -valued map
- (c,d) likelihood maps  $f, f'$ , both produce segmentation  $X$  if we threshold at  $\alpha = 0.5$ . But  $f$  is closer to being correct, due to shallower gaps
- Conclusion: single threshold  $\alpha = 0.5$  not enough, but **persistent homology**  $\rightarrow$  all possible threshold values

# Topological Loss

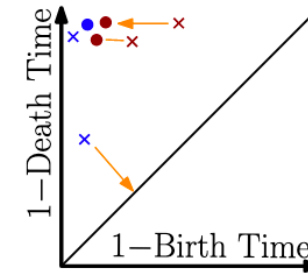
Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeurIPS 2019*



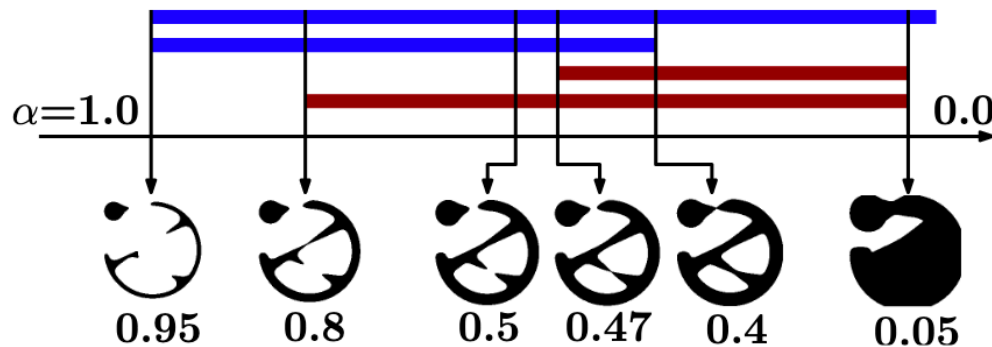
(a) Filtration induced by the ground truth function,  $g$ .



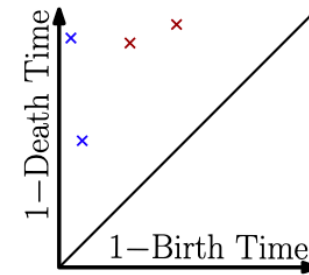
(b)  $Dgm(g)$



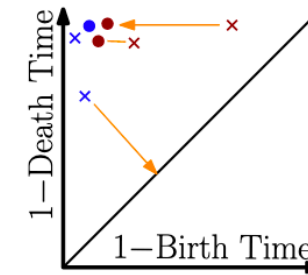
(c)  $Dgm(g)+Dgm(f)$



(d) Filtration induced by the likelihood function,  $f$ .



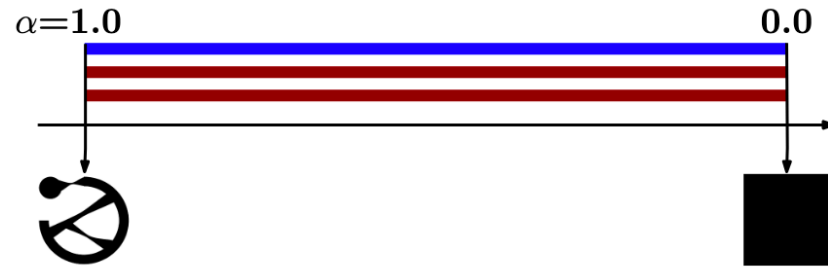
(e)  $Dgm(f)$



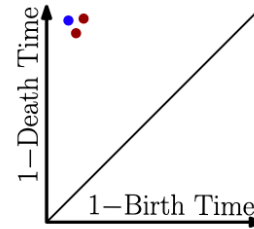
(f)  $Dgm(g)+Dgm(f')$

# Topological Loss

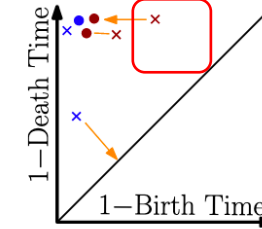
Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeurIPS 2019*



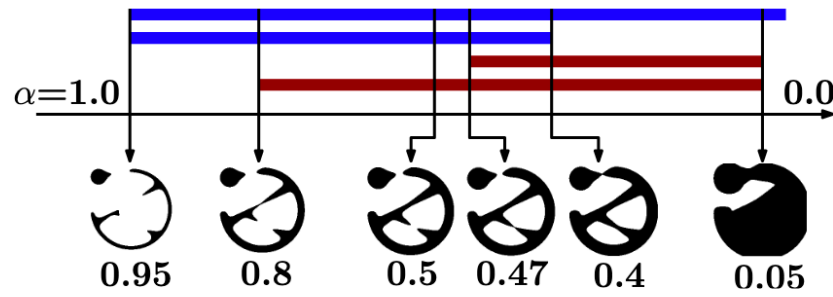
(a) Filtration induced by the ground truth function,  $g$ .



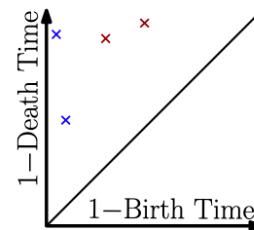
(b)  $Dgm(g)$



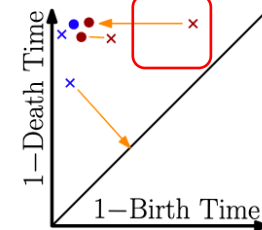
(c)  $Dgm(g)+Dgm(f)$



(d) Filtration induced by the likelihood function,  $f$ .



(e)  $Dgm(f)$



(f)  $Dgm(g)+Dgm(f')$

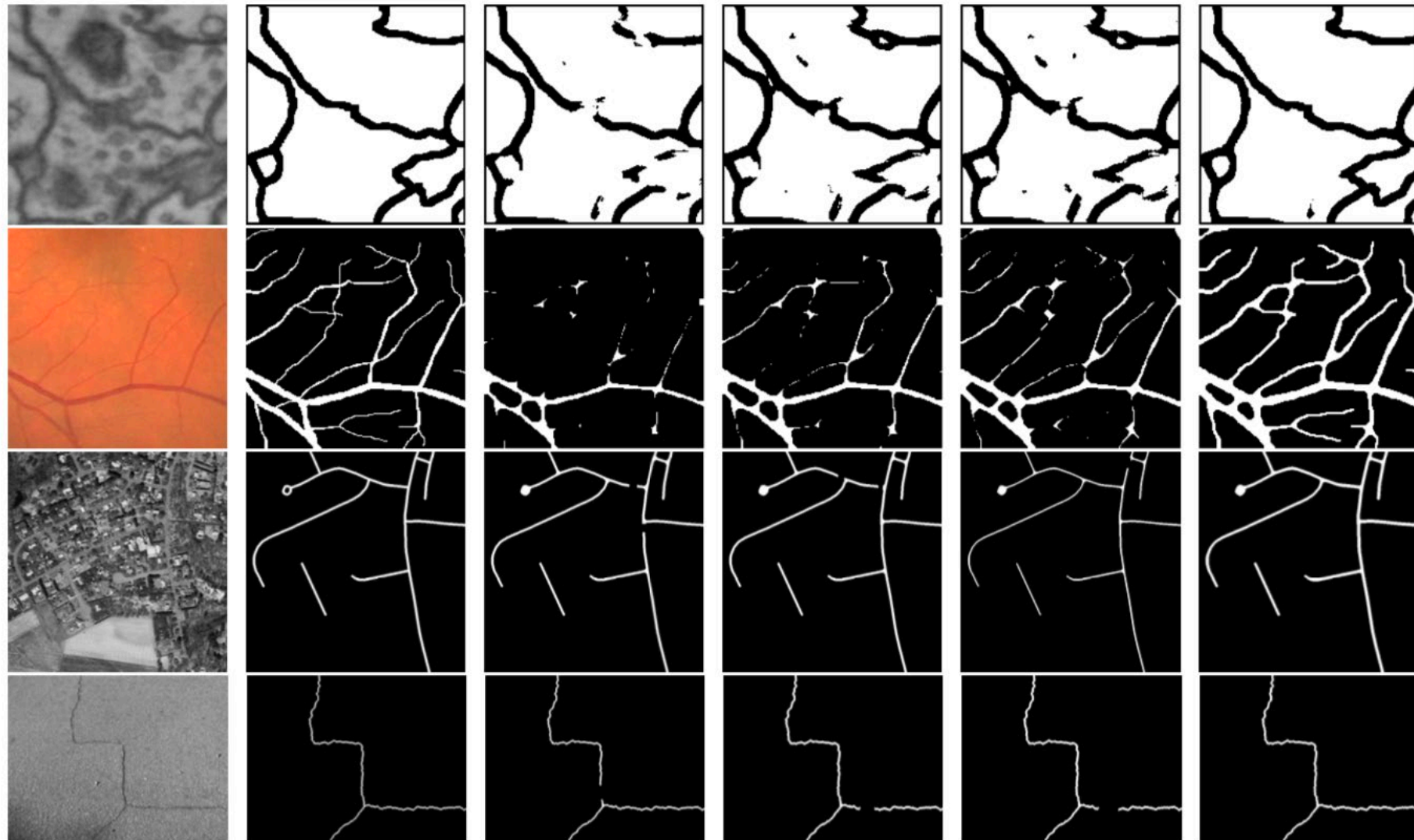
A slightly modified bottleneck distance:  $\gamma^*$  an optimal matching  $Dgm_k(f) \rightarrow Dgm_k(g)$ ,  $k \geq 0$ ,

$$L_{top}(f, g) = \sum_k \sum_{p \in Dgm_k(f)} \|p - \gamma^*(p)\|_2^2 = \sum_k \sum_{p \in Dgm_k(f)} [birth(p) - birth(\gamma^*(p))]^2 + [death(p) - death(\gamma^*(p))]^2$$

Intuition:  $L_{top}(f, g)$  measures the cost of changing  $f$  to look like  $g$ .

# Results

Hu, Fuxin, Samaras, Chen. *Topology-preserving deep image segmentation. NeuRIPS 2019*



Sample images

ground truth

DIVE '16

U-Net '15

Mosin '18

**TopoLoss'19**

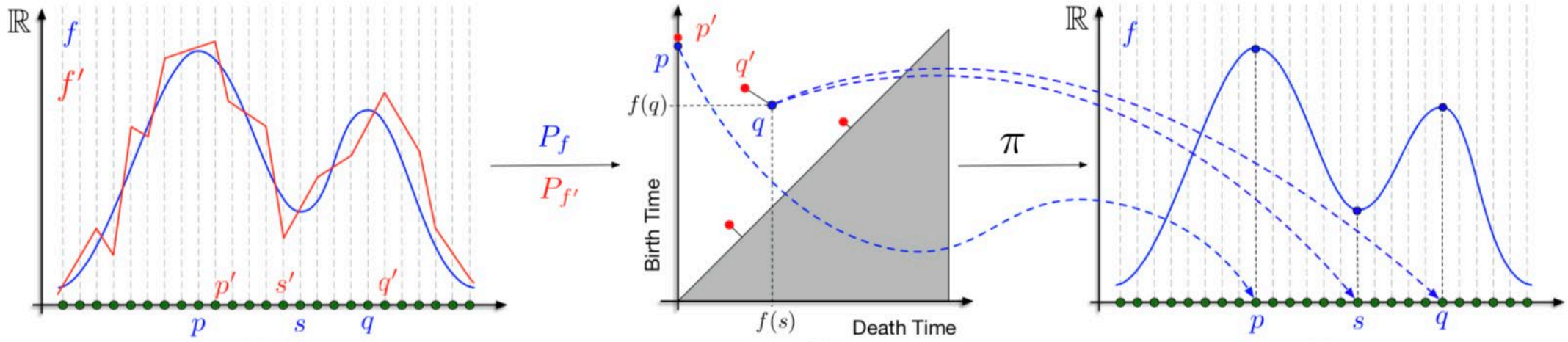
# A Closer Look at the Topological Loss

Poulenard, Skraba, Ovsjanikov. *Topological Function Optimization for Continuous Shape Matching*. SGP '18

- ◆ Neural network parameters  $\alpha$  produced a likelihood function  $f$
- ◆ Superlevel set filtration of  $f$  created a persistence diagram  $\text{Dgm}(f)$
- ◆ Consider functional  $\mathcal{L} := L_{top}(-, g)$ , where  $g$  is the ground truth function and  $L_{top}$  measures the cost of matching  $\text{Dgm}(-)$  to  $\text{Dgm}(g)$
- ◆ Want to optimize  $f$  to minimize  $L_{top}(f, g)$ , i.e. need gradient  $\nabla_{\alpha} \mathcal{L}$
- ◆ First step: for a point  $(b, d)$  in a persistence diagram, compute:  
$$\frac{\partial b}{\partial \alpha} \quad \text{and} \quad \frac{\partial d}{\partial \alpha}$$
- ◆ Strategy:  $(b, d)$  came from a pairing  $(\sigma, \tau)$ ---use this!

# Gradients for Persistence Diagrams

Poulenard, Skraba, Ovsjanikov. *Topological Function Optimization for Continuous Shape Matching*. SGP '18



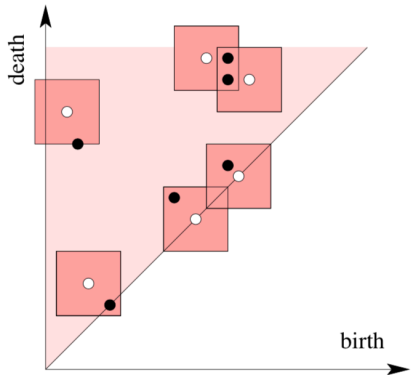
$$\zeta: (b, d) \rightarrow (\sigma, \tau), \quad \eta: \sigma \rightarrow v_\sigma \text{ where } v_\sigma \in \arg \min_{v \in \sigma} f(v)$$

$$\pi := (\eta, \eta) \circ \zeta \quad \text{“reverse” map from persistence diagram to pairs of vertices}$$

By stability,  $\pi$  remains unchanged for small perturbations of the diagram, and so:

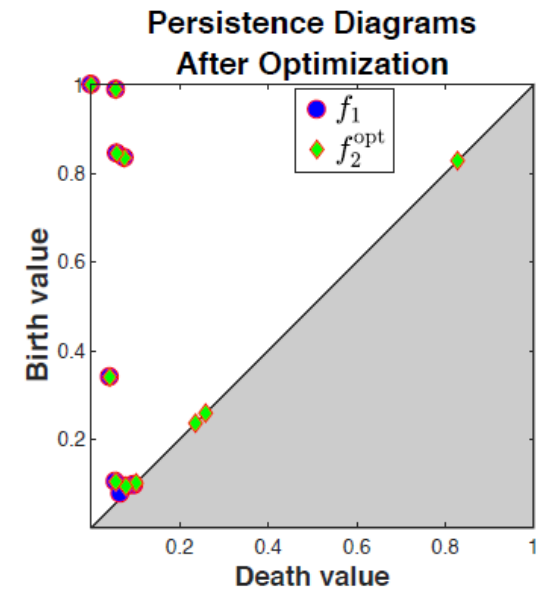
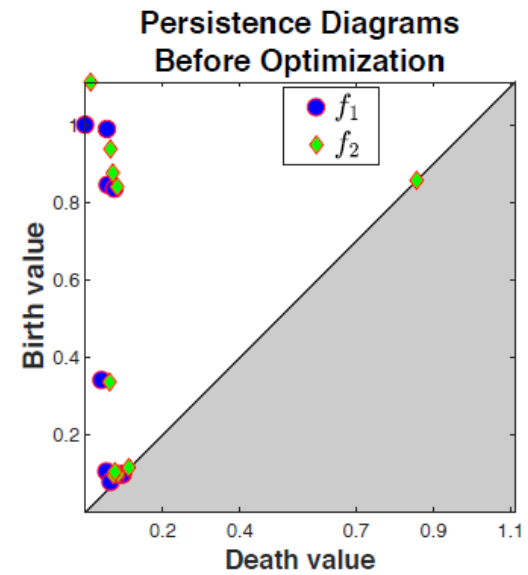
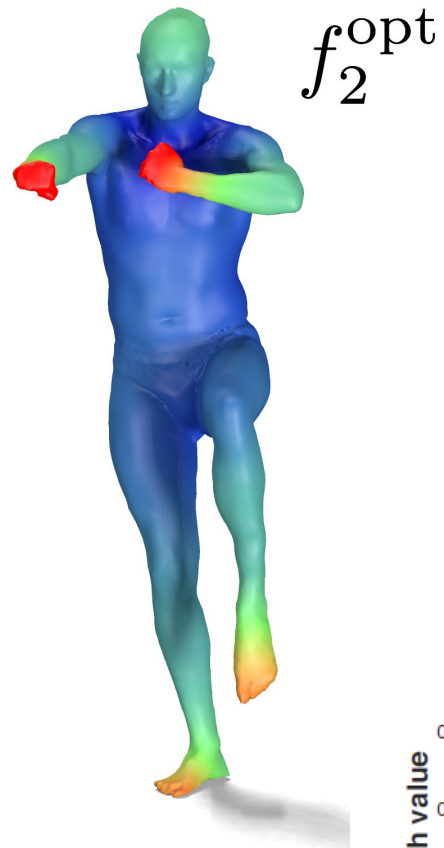
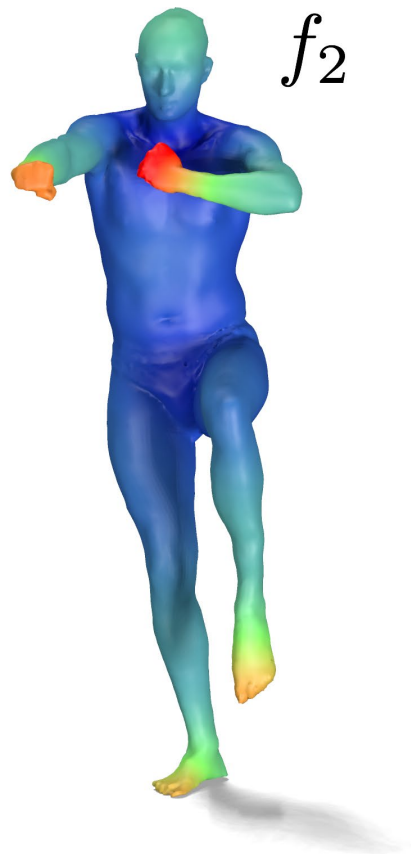
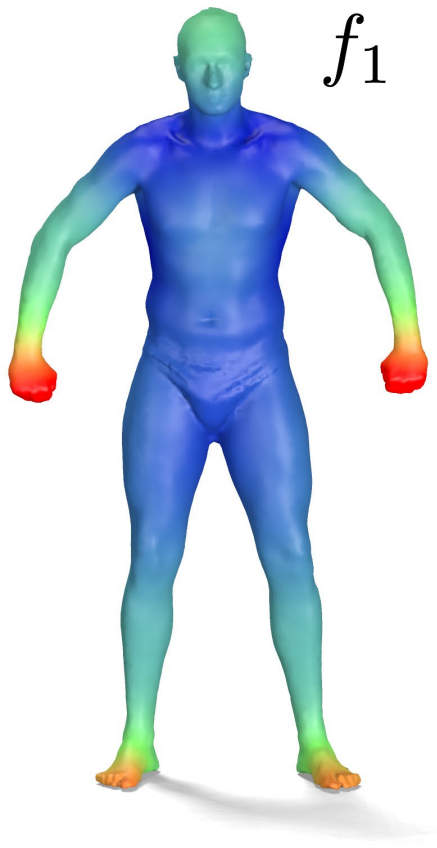
$$\frac{\partial b}{\partial \alpha} = \frac{\partial f(\pi(b))}{\partial \alpha} = \frac{\partial f(v_\sigma)}{\partial \alpha}, \text{ and similarly } \frac{\partial d}{\partial \alpha} = \frac{\partial f}{\partial \alpha}(v_\tau)$$

**Conclusion:** “the derivative (of the diagram) is equivalent to the derivative of the function evaluated at the image of the map  $\pi$ ”



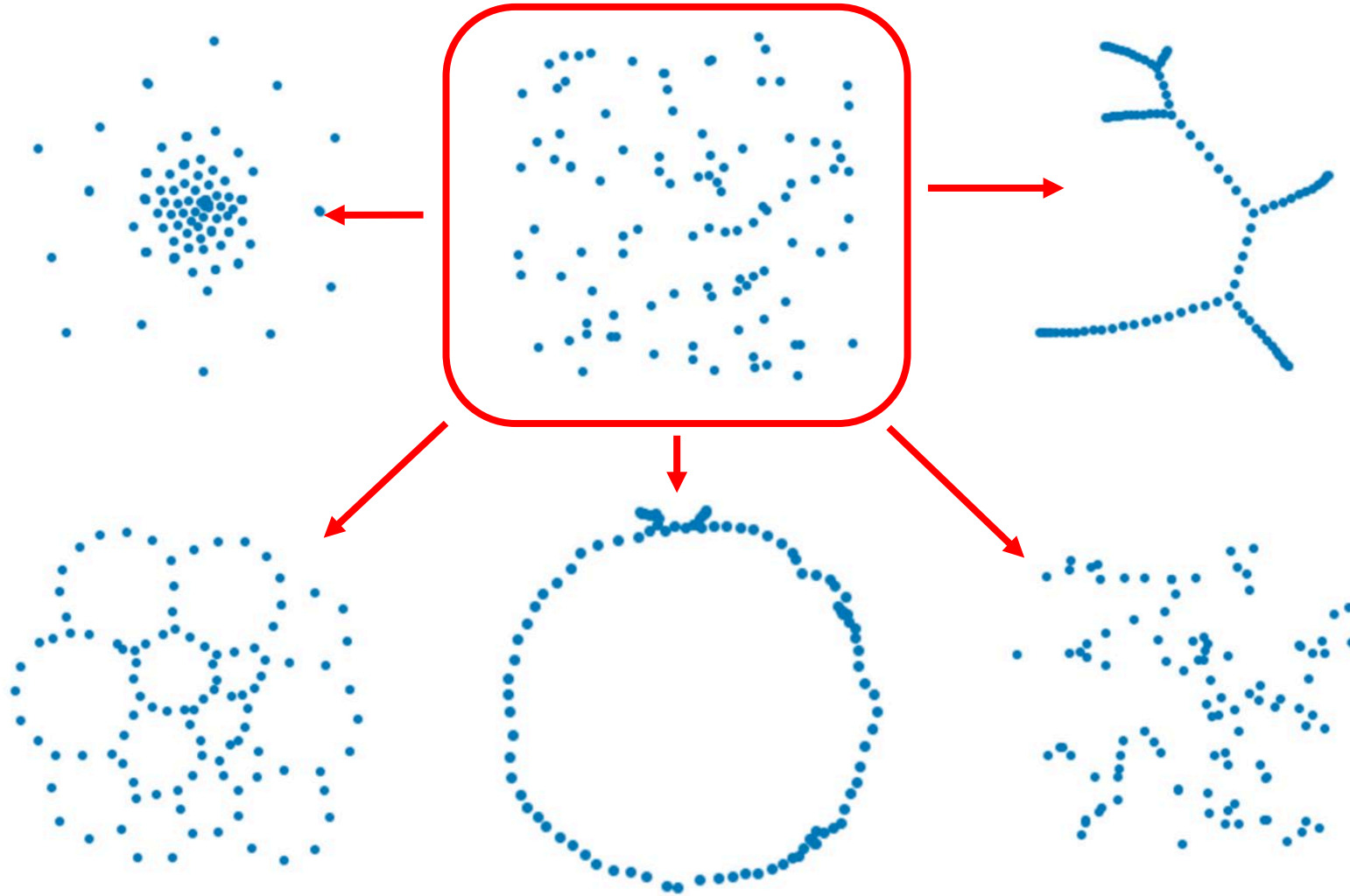
# Gradients for Persistence Diagrams

Poulenard, Skraba, Ovsjanikov. *Topological Function Optimization for Continuous Shape Matching*. SGP '18



# Topological Noise Reduction

Bruel-Gabrielsson, Nelson, Dwaraknath, Skraba, Guibas, Carlsson, *A topology layer for machine learning*. AISTATS 2020



# More TDA Info

- <https://appliedtopology.org/>
- <https://scikit-tda.org/>
- <https://github.com/giotto-ai/giotto-tda>
- <https://tda-in-ml.slack.com/>

# That's All

