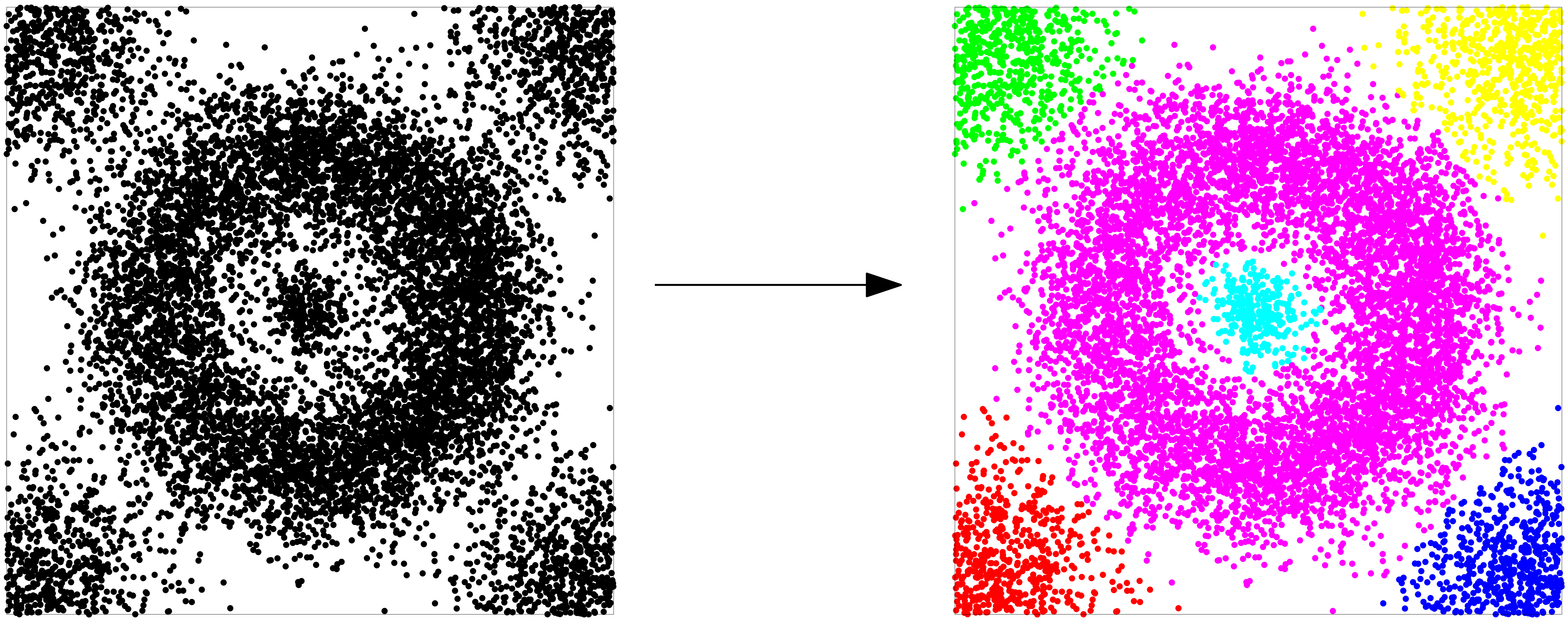
Introduction to persistence theory through an application in clustering

[F. Chazal, L. Guibas, S. Oudot, P. Skraba: Persistence-Based Clustering in Riemannian Manifolds, *Journal of the ACM*, 60(6):1–38, 2013]

Clustering

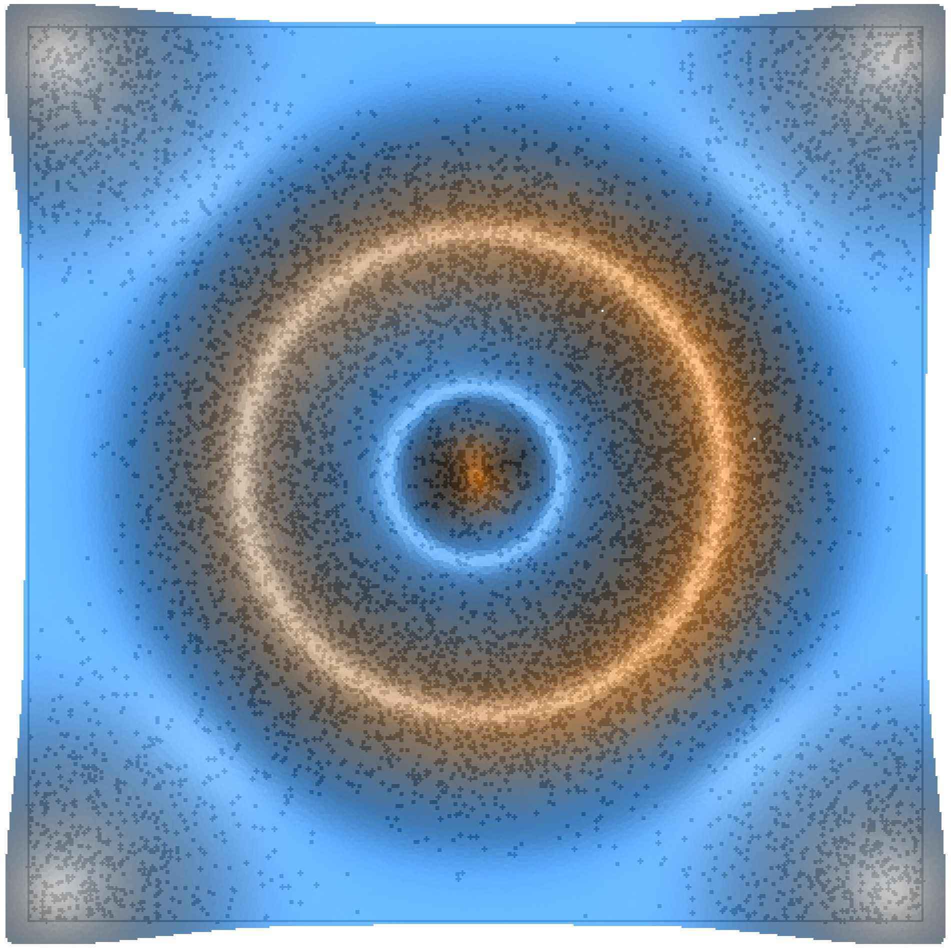
Input: a finite set P of data in some metric space



Task: partition the data set P into *homogeneous* subsets (clusters)

Mode-Seeking Paradigm

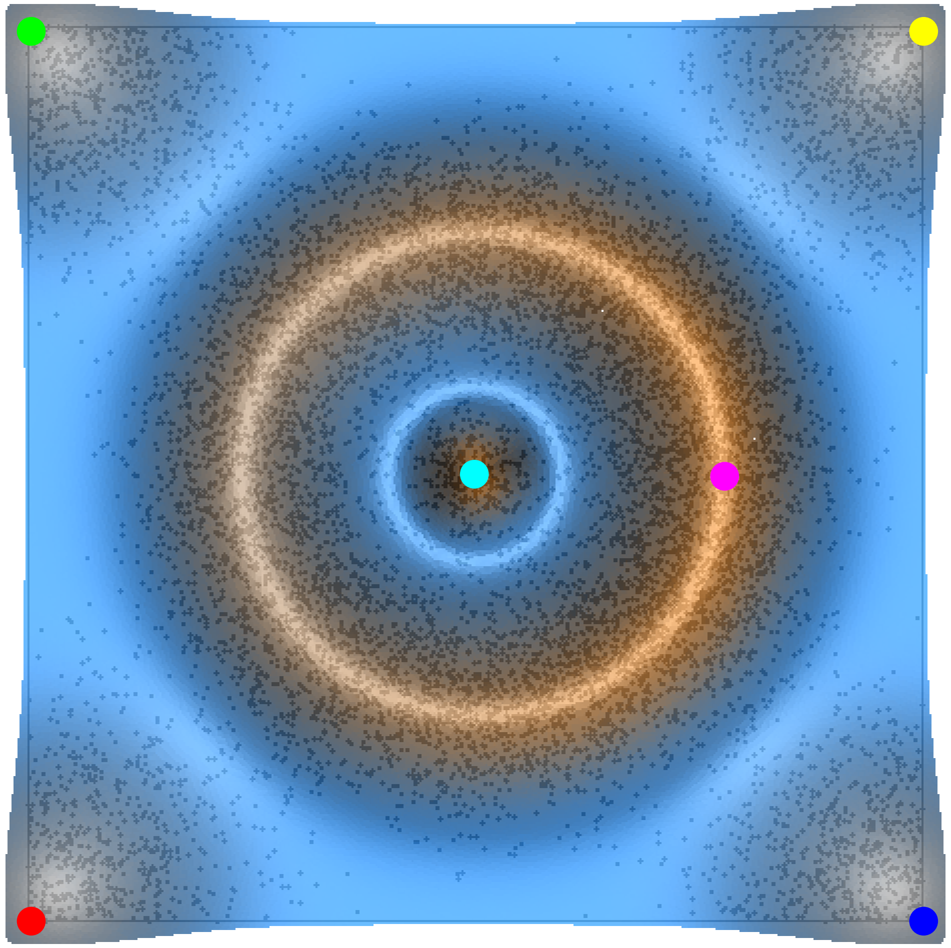
- Hyp: data are sampled iid from a probability measure μ with density f (both **unknown**)



Mode-Seeking Paradigm

- Hyp: data are sampled iid from a probability measure μ with density f (both **unknown**)

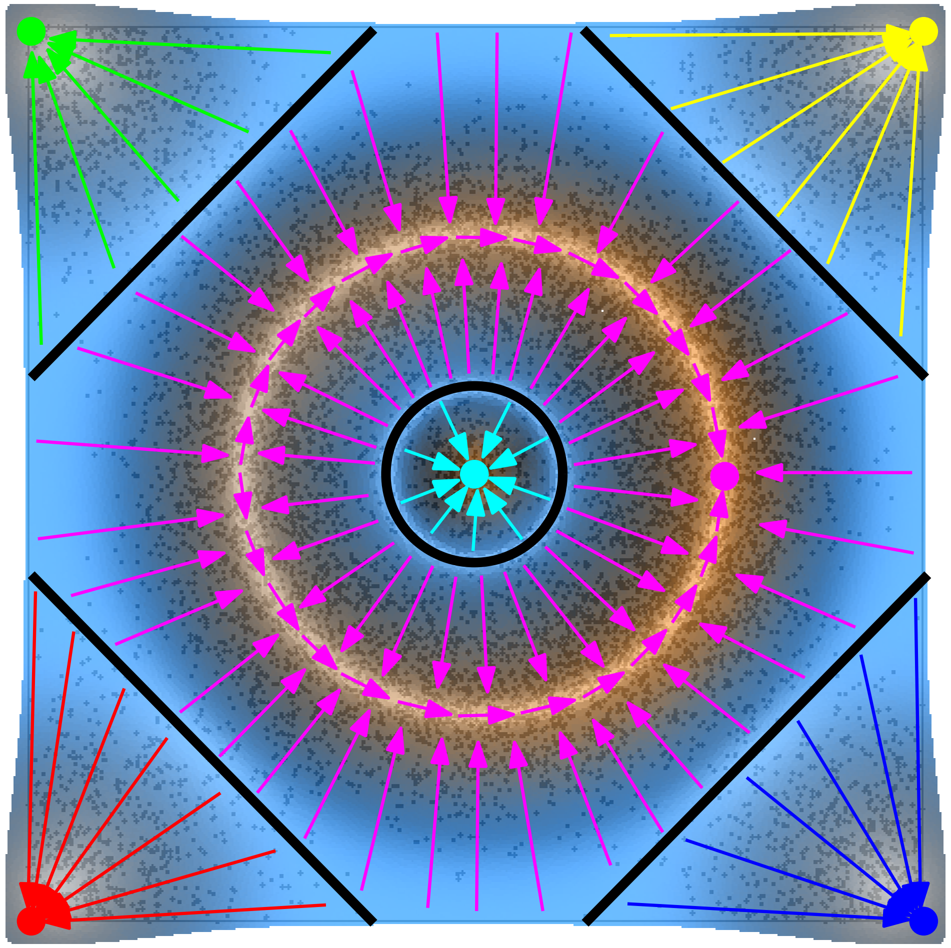
- Partition the data according to the **stable manifolds** of the peaks of f



Mode-Seeking Paradigm

- Hyp: data are sampled iid from a probability measure μ with density f (both **unknown**)

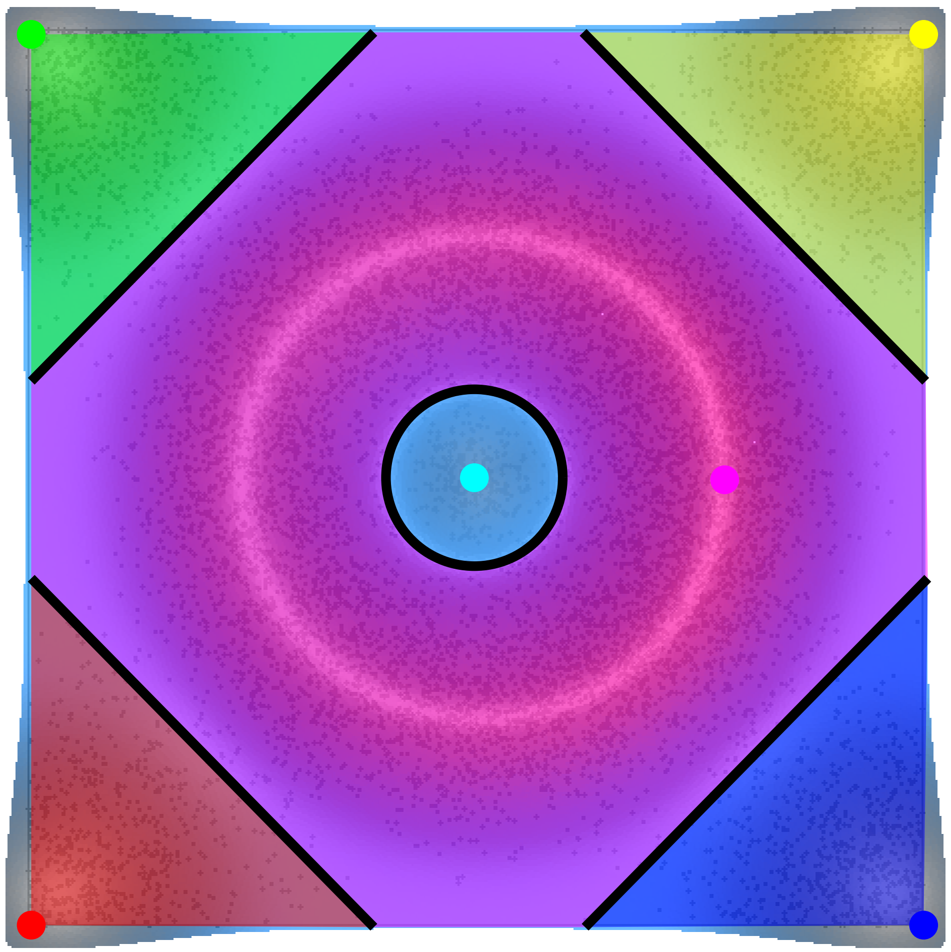
- Partition the data according to the **stable manifolds** of the peaks of f



Mode-Seeking Paradigm

- Hyp: data are sampled iid from a probability measure μ with density f (both **unknown**)

- Partition the data according to the **stable manifolds** of the peaks of f

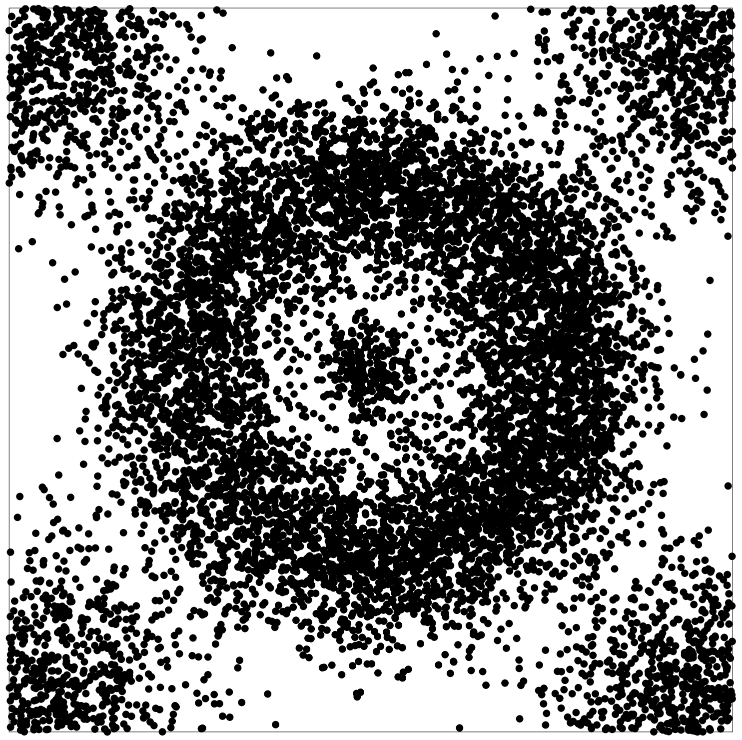


Hill-Climbing Schemes

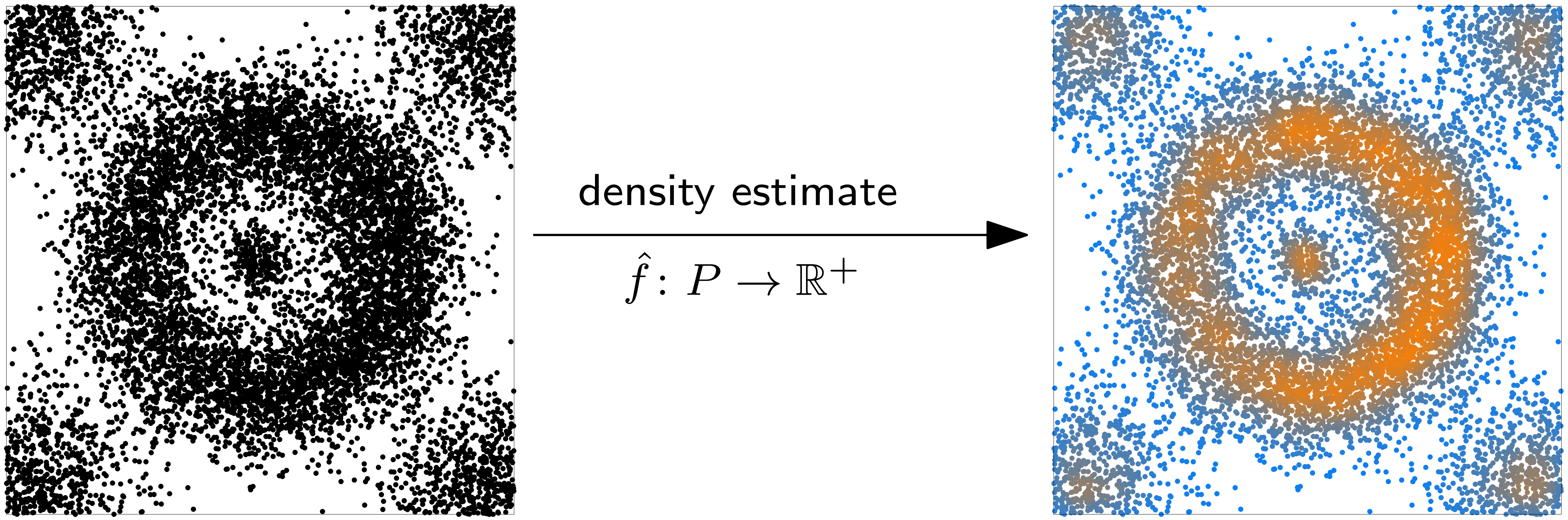
- **Numerical**, e.g. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

- **Combinatorial**, e.g. W. L. Koontz, P. M. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Trans. on Computers*, 24:936–944, September 1976.

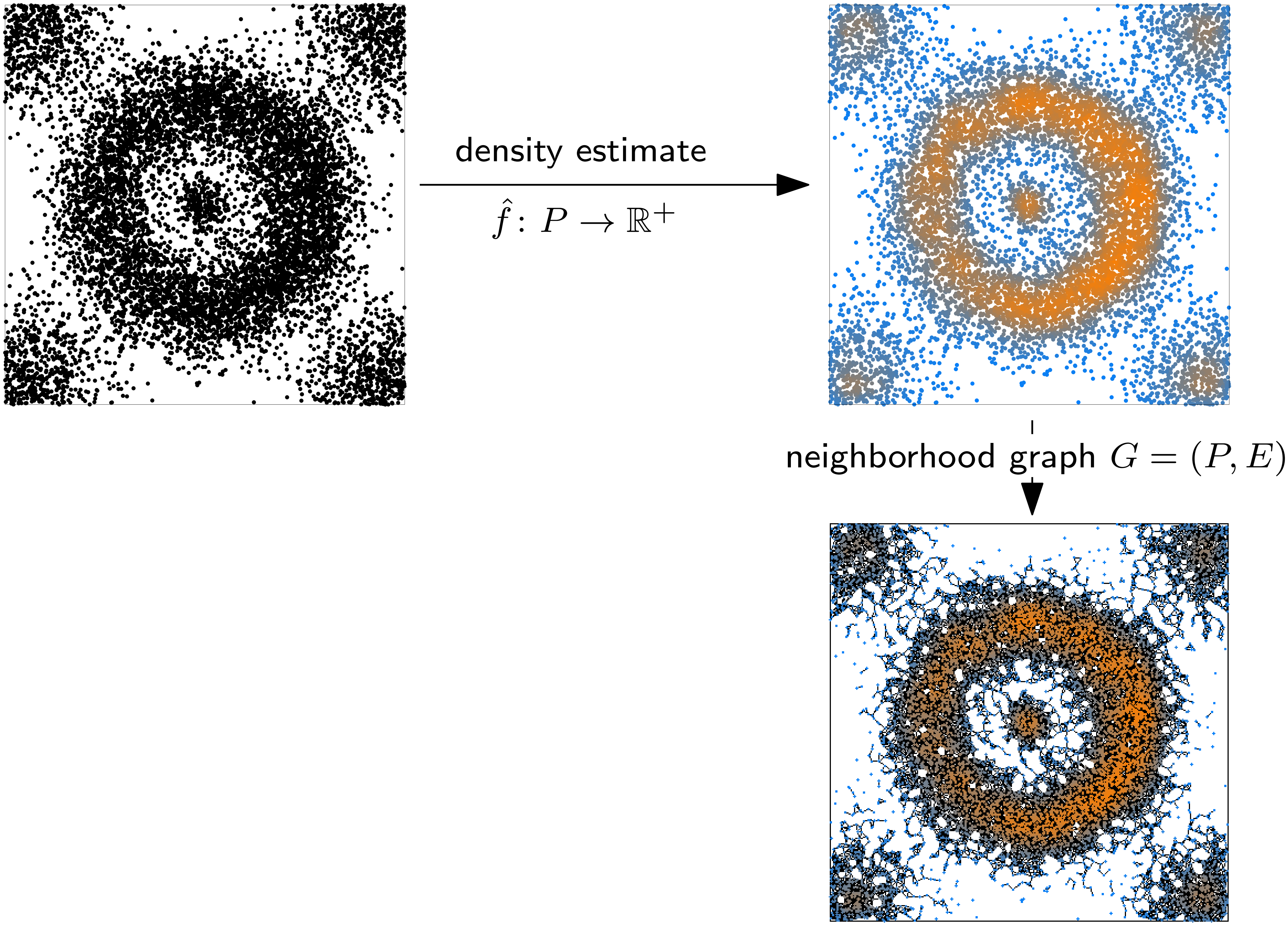
[Koontz, Narendra, Fukunaga'76] in a Nutshell



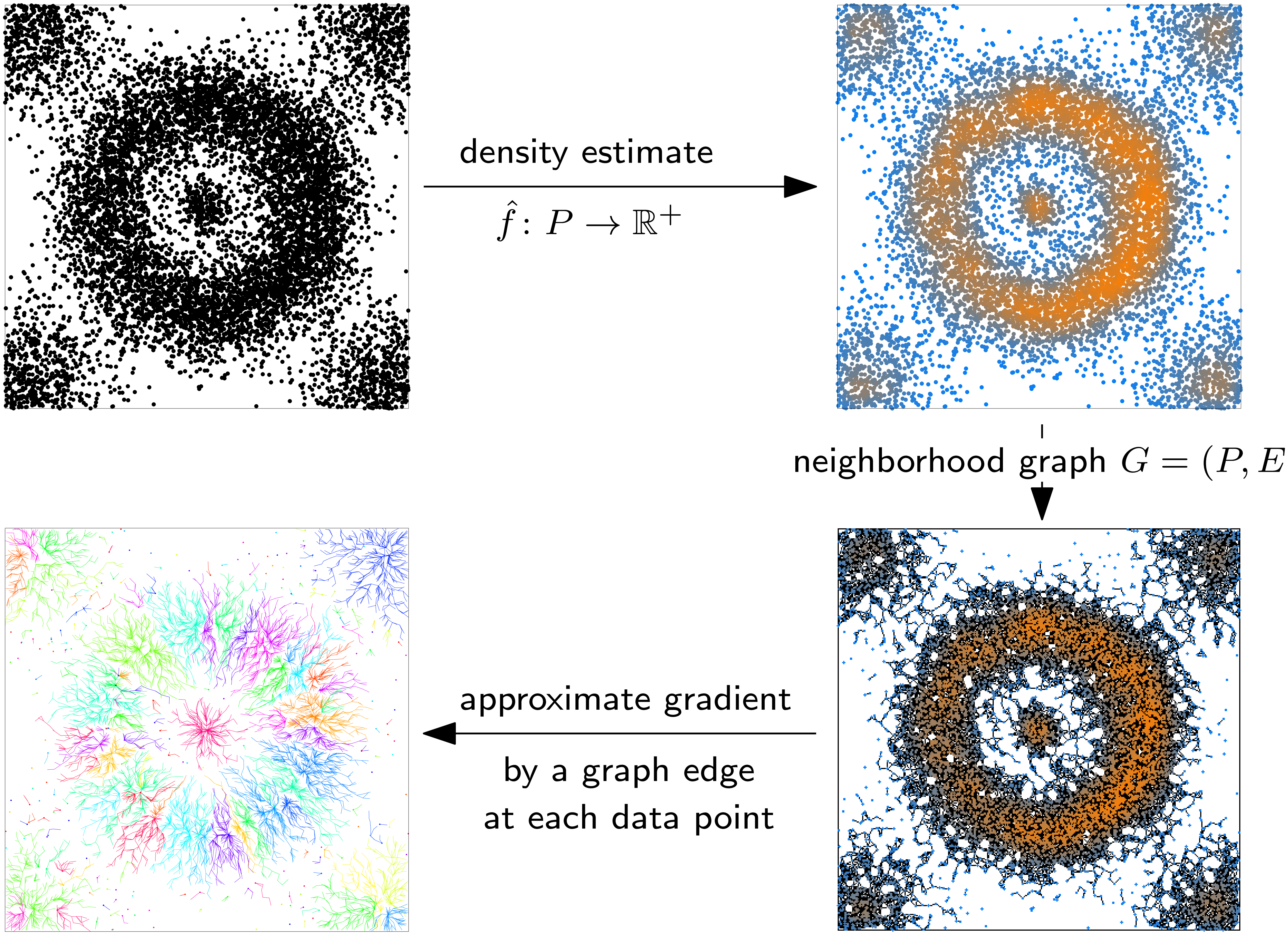
[Koontz, Narendra, Fukunaga'76] in a Nutshell



[Koontz, Narendra, Fukunaga'76] in a Nutshell



[Koontz, Narendra, Fukunaga'76] in a Nutshell



Pseudo-code:

Input: neighborhood graph G with n vertices, n-dimensional vector $\hat{f}$ (density estimator)

Sort the vertex indices $\{1, 2, \cdots, n\}$ so that $\hat{f}(1) \geq \hat{f}(2) \geq \cdots \geq \hat{f}(n)$;
Initialize a union-find data structure (disjoint-set forest) $\mathcal{U}$ and two vectors g, r of size n;

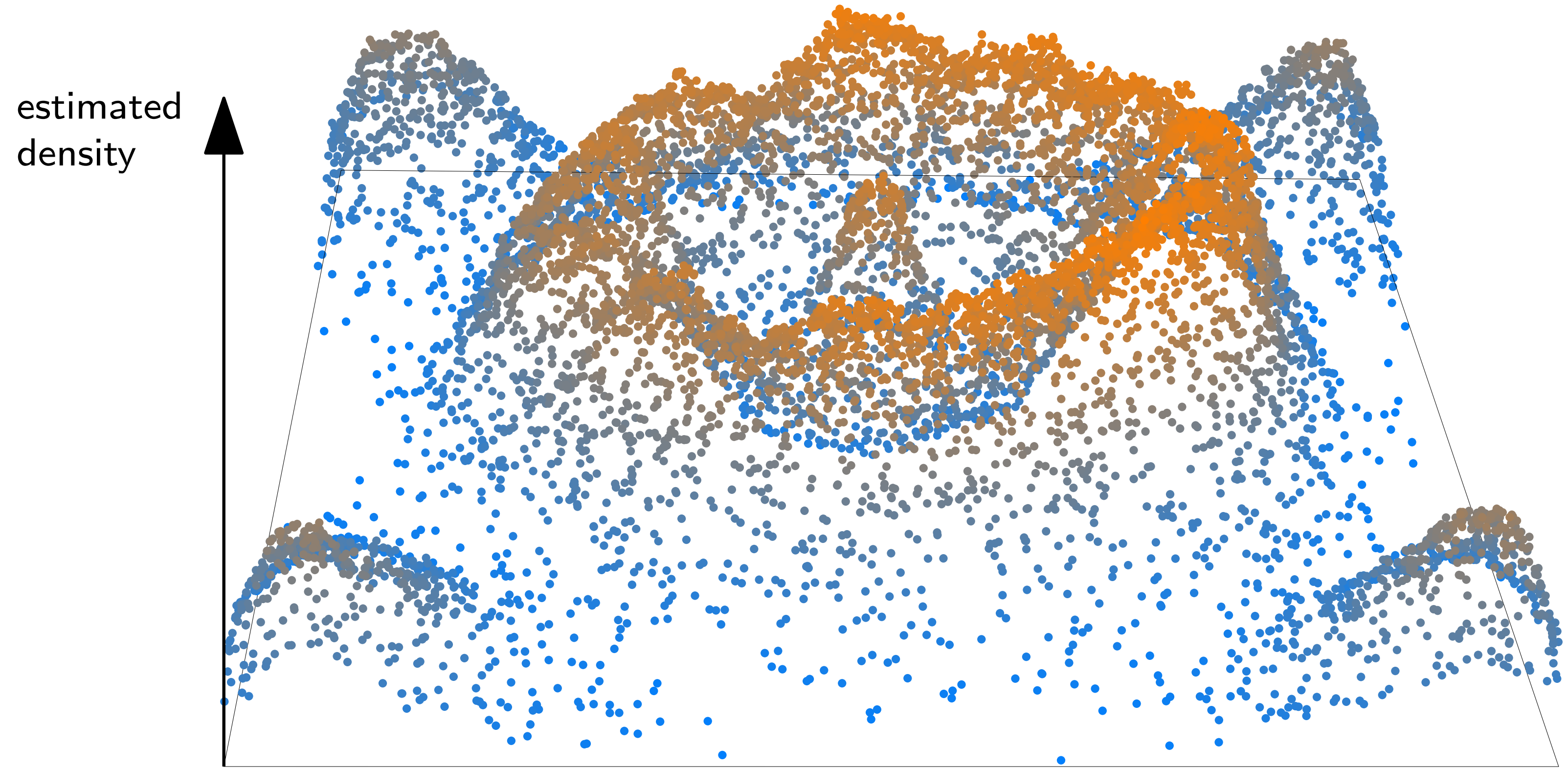
for $i = 1$ to n **do**
  Let $\mathcal{N}$ be the set of neighbors of i in G that have indices lower than i;
  if $\mathcal{N} = \emptyset$ *// vertex i is a peak of $\hat{f}$ within G*
    Create a new entry e in $\mathcal{U}$ and attach vertex i to it;
    $r(e) \leftarrow i$ *// $r(e)$ stores the root vertex associated with the entry e*
  else *// vertex i is not a peak of $\hat{f}$ within G*
    $g(i) \leftarrow \mathrm{argmax}_{j \in \mathcal{N}} \hat{f}(j)$ *// $g(i)$ stores the approximate gradient at vertex i*
    $e_i \leftarrow \mathcal{U}.\mathtt{find}(g(i))$;
    Attach vertex i to the entry e_i;

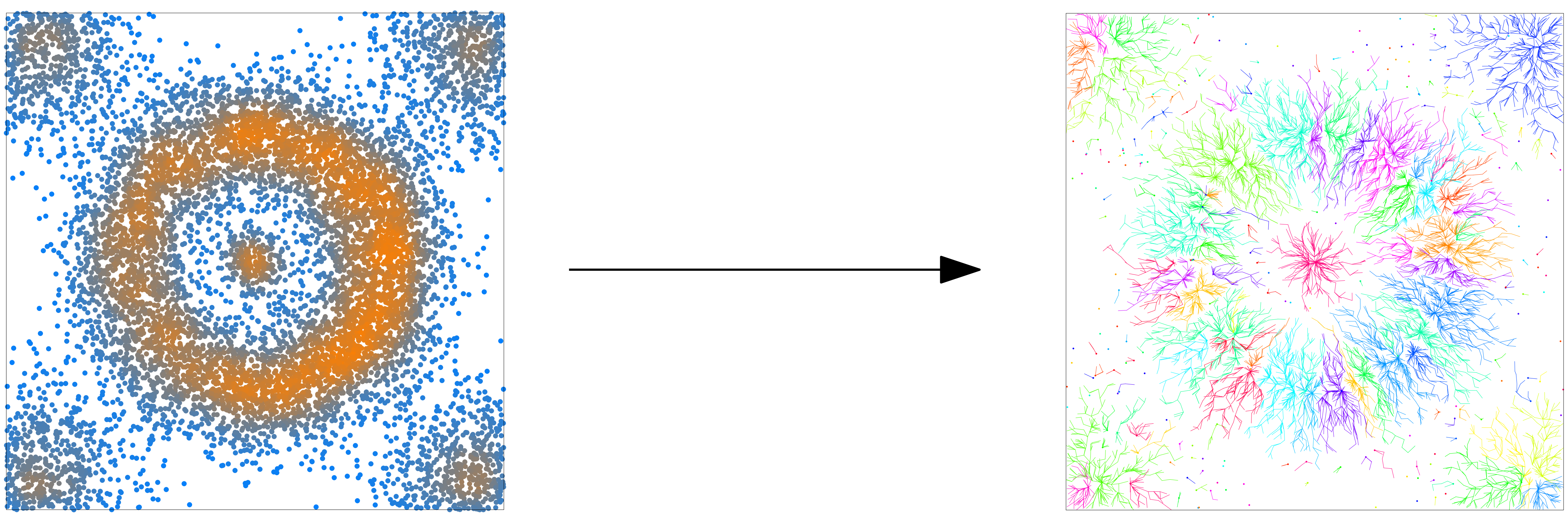
graph-based
hill-climbing
(1976)

Output: the collection of entries e in $\mathcal{U}$

Why things go ill

Noisy estimator



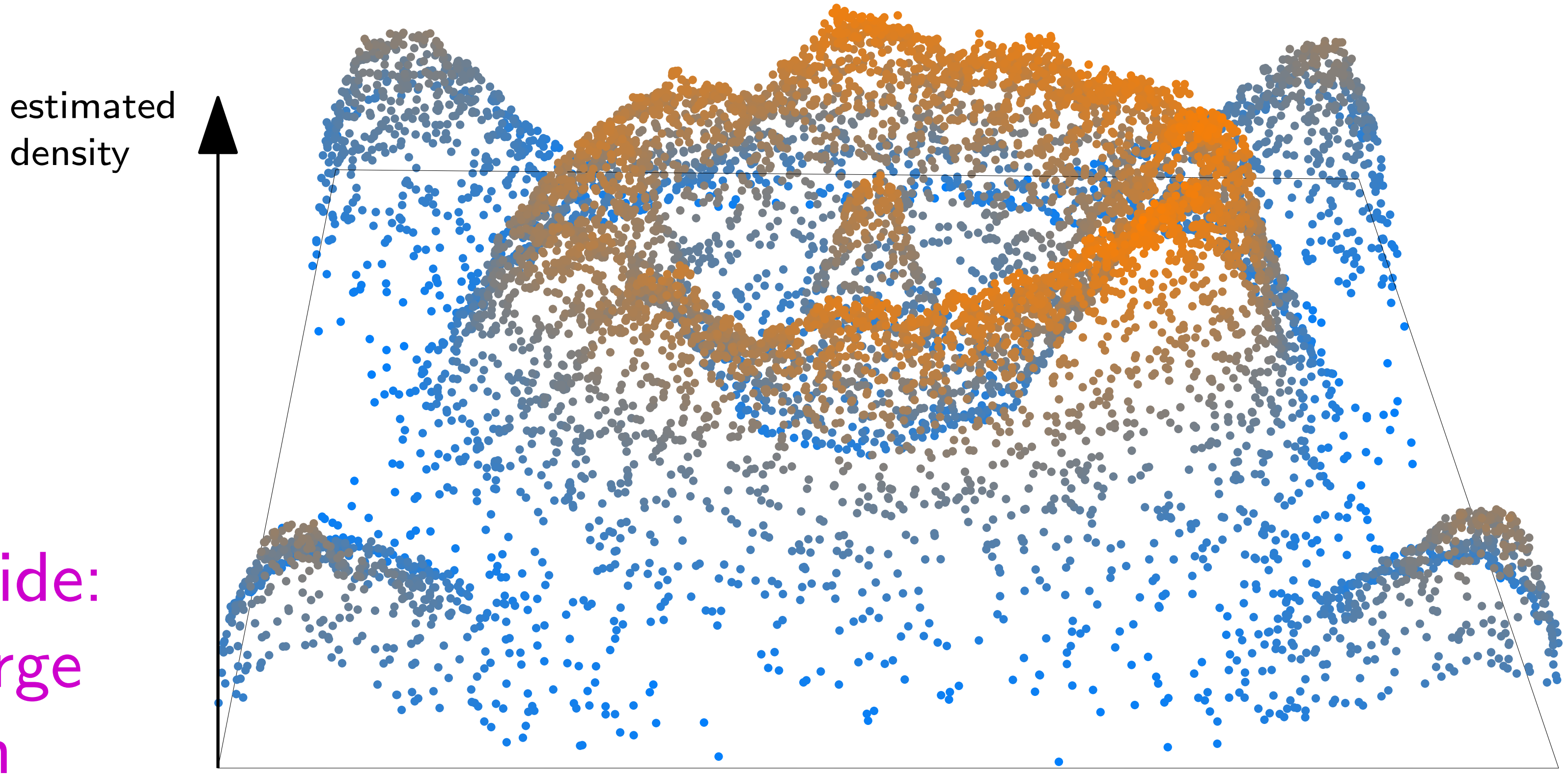


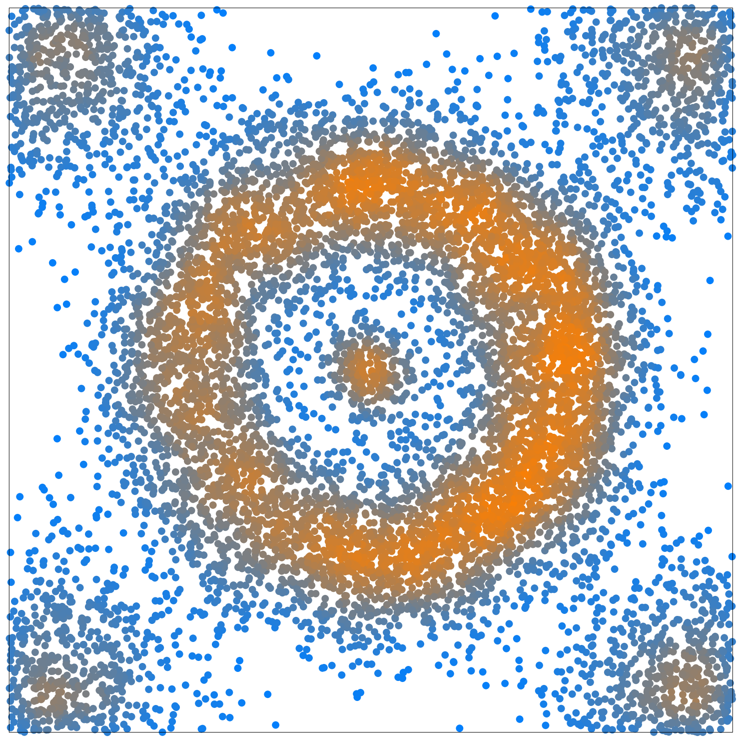
Why things go ill

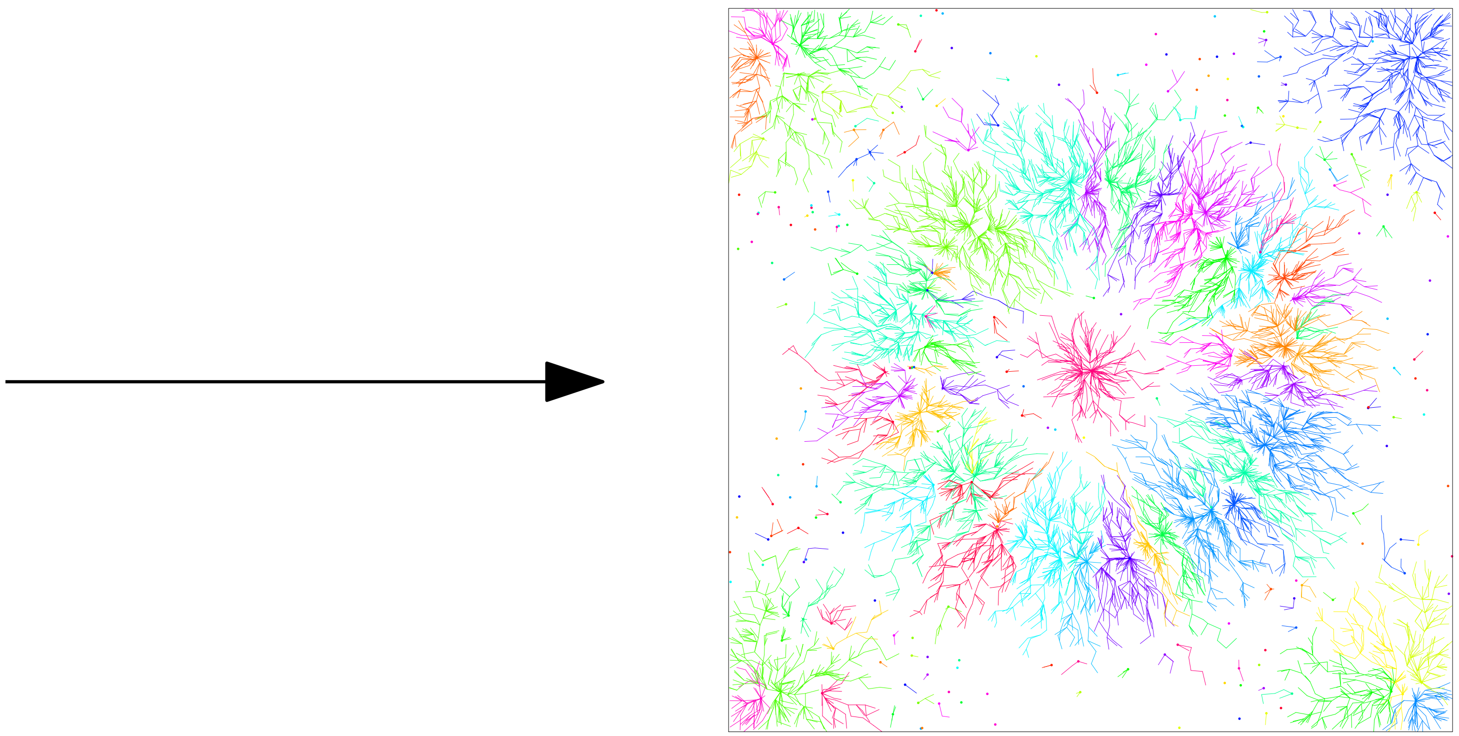
Noisy estimator

Solution:

- merge clusters in post-processing step
- use persistence as a guide:
  - which clusters to merge
  - where to merge them

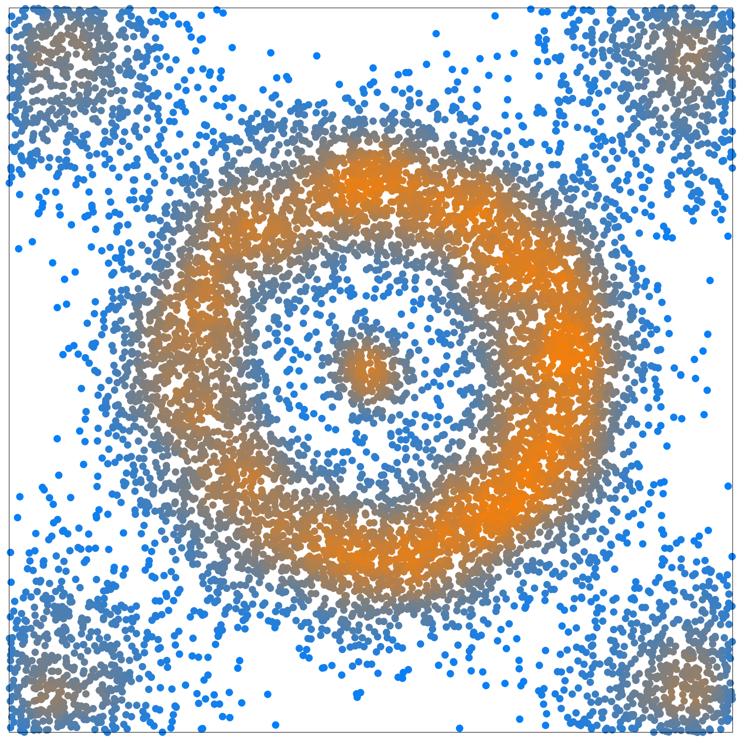




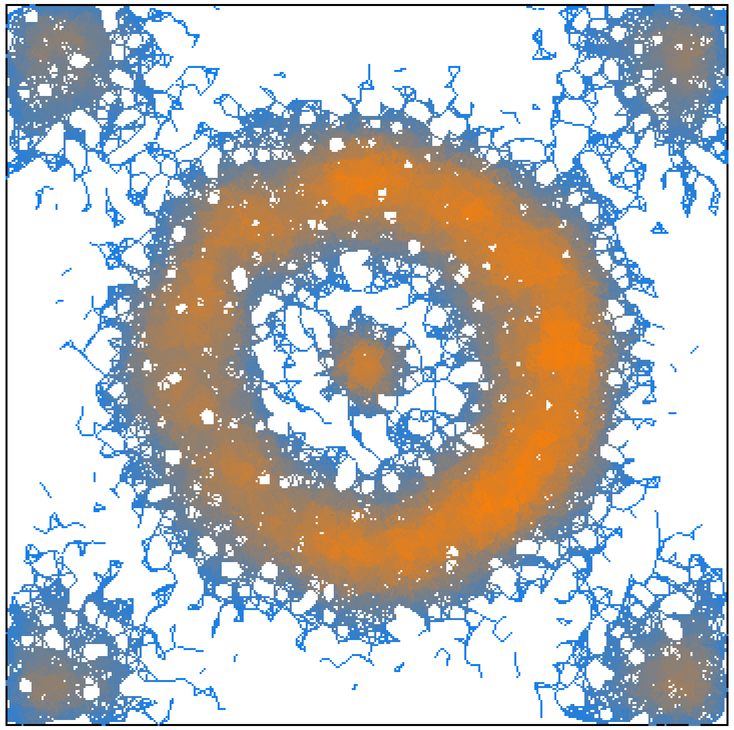


Persistence of the estimated density in the graph

- Extend $\hat{f}\colon P \to \mathbb{R}$ to a map $G \to \mathbb{R}$ by $\hat{f}((u,v)) := \min\left\{\hat{f}(u),\ \hat{f}(v)\right\}$



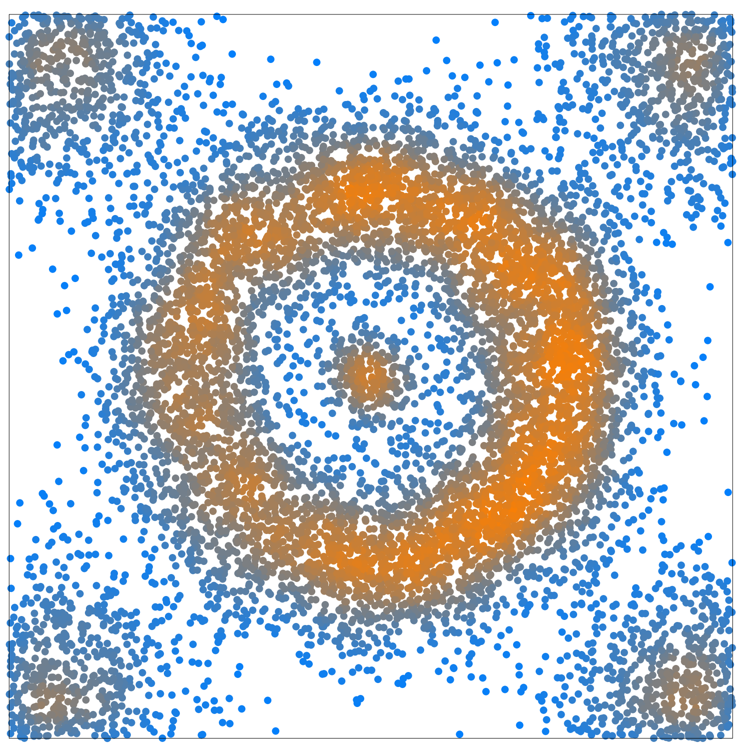
$\hat{f}$



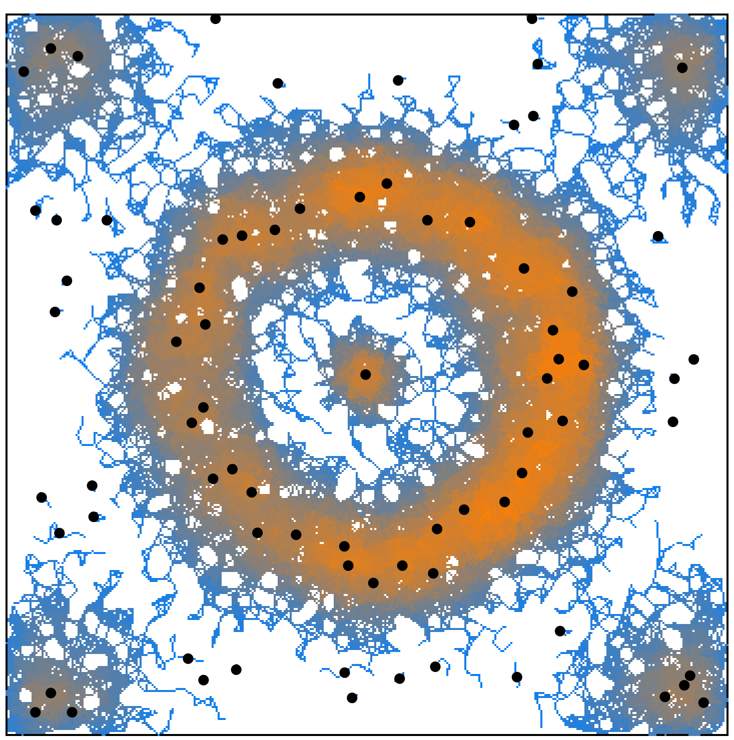
$\hat{f}$ extended to G

Persistence of the estimated density in the graph

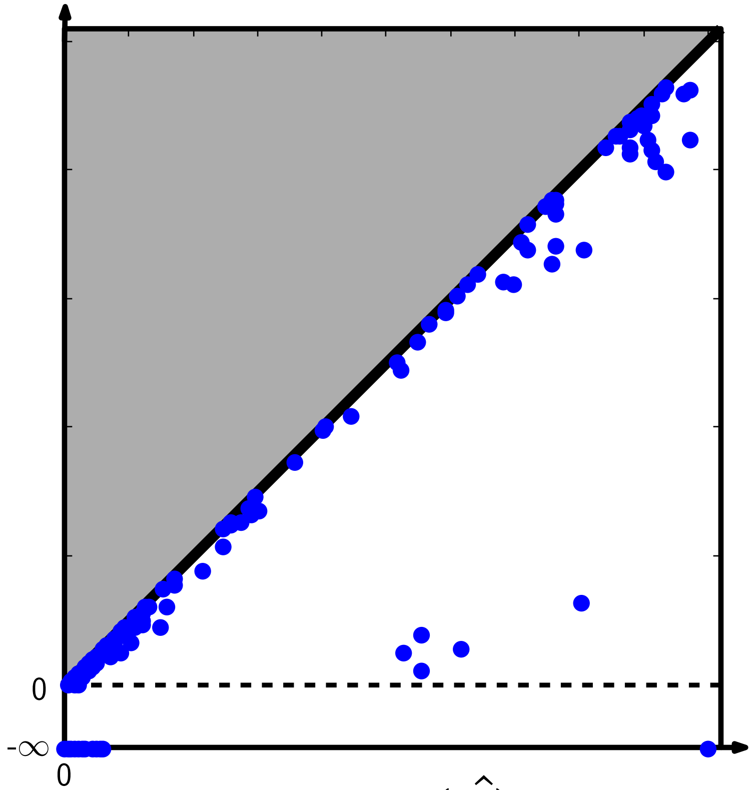
- Extend $\hat{f}\colon P \to \mathbb{R}$ to a map $G \to \mathbb{R}$ by $\hat{f}((u,v)) := \min\left\{\hat{f}(u),\ \hat{f}(v)\right\}$
- $D(\hat{f})$ encodes the lifespans of the peaks of $\hat{f}$ as independent components in G



$\hat{f}$



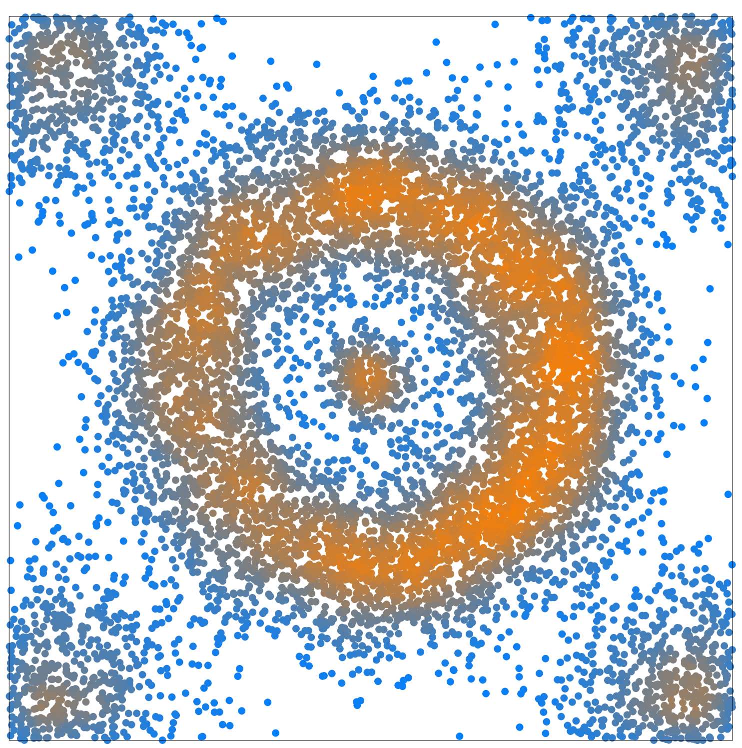
$\hat{f}$ extended to G



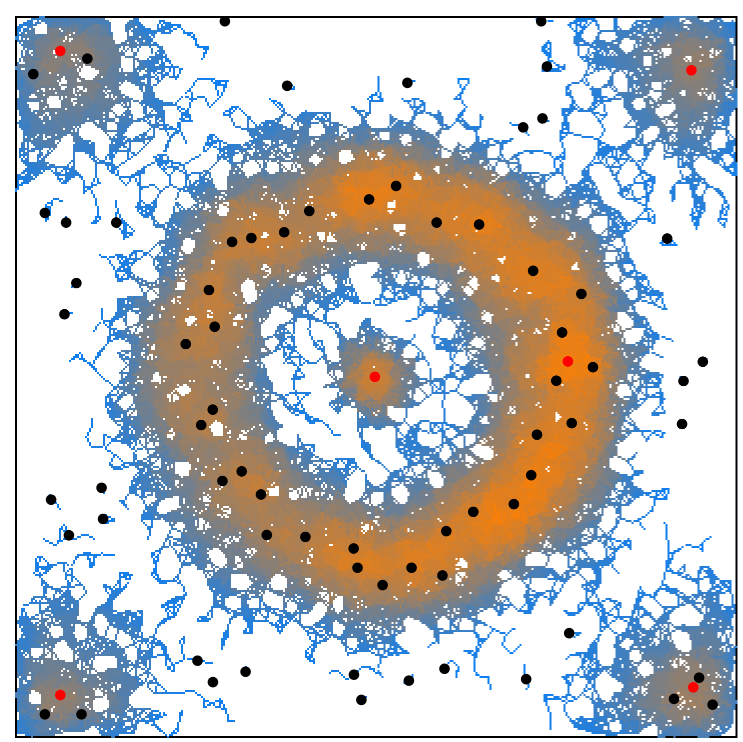
$D(\hat{f})$

Persistence of the estimated density in the graph

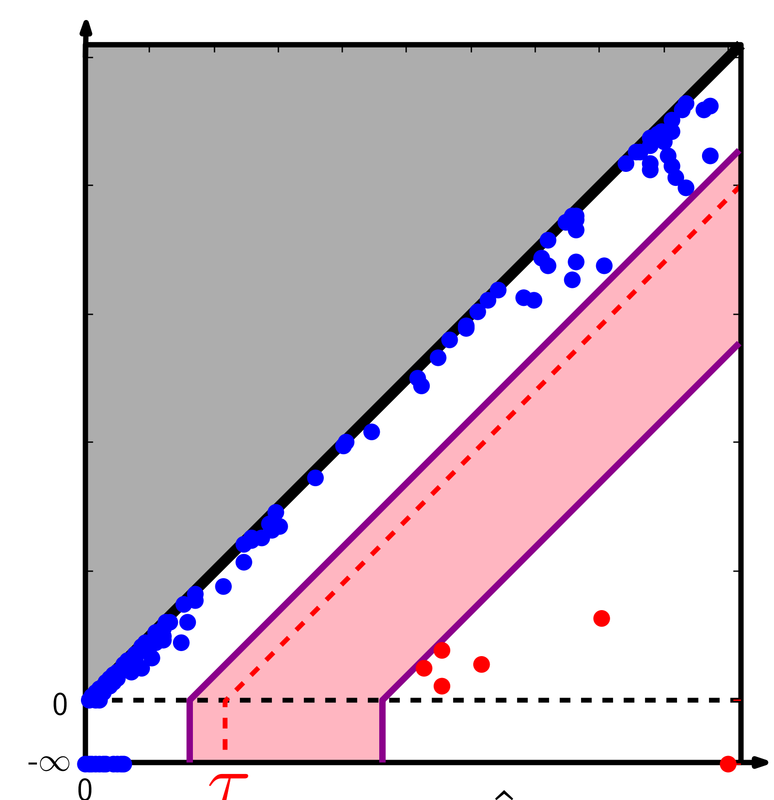
- Extend $\hat{f}\colon P \to \mathbb{R}$ to a map $G \to \mathbb{R}$ by $\hat{f}((u,v)) := \min\left\{\hat{f}(u),\ \hat{f}(v)\right\}$
- $D(\hat{f})$ encodes the lifespans of the peaks of $\hat{f}$ as independent components in G
- from $D(\hat{f})$, the user can infer a persistence threshold τ



$\hat{f}$



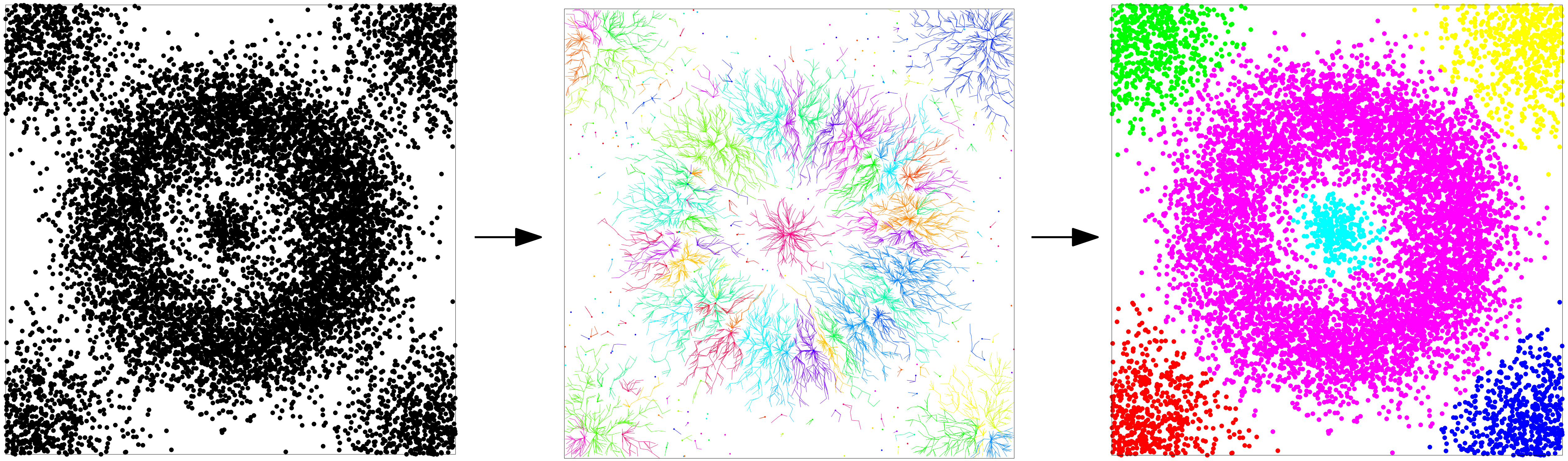
$\hat{f}$ extended to G



$D(\hat{f})$

Persistence of the estimated density in the graph

- Extend $\hat{f}\colon P \to \mathbb{R}$ to a map $G \to \mathbb{R}$ by $\hat{f}((u,v)) := \min\left\{\hat{f}(u),\ \hat{f}(v)\right\}$
- $D(\hat{f})$ encodes the lifespans of the peaks of $\hat{f}$ as independent components in G
- from $D(\hat{f})$, the user can infer a persistence threshold τ
- inductively merge clusters of peaks of prominence $\leq \tau$ into their parent's cluster



Pseudo-code:

Input: simple graph G with n vertices, n-dimensional vector $\hat{f}$, real parameter $\tau \geq 0$.

Sort the vertex indices $\{1, 2, \cdots, n\}$ so that $\hat{f}(1) \geq \hat{f}(2) \geq \cdots \geq \hat{f}(n)$;
Initialize a union-find data structure $\mathcal{U}$ and two vectors g, r of size n;

for $i = 1$ to n **do**
    Let $\mathcal{N}$ be the set of neighbors of i in G that have indices lower than i;
    if $\mathcal{N} = \emptyset$ *// vertex i is a peak of $\hat{f}$ within G*
        Create a new entry e in $\mathcal{U}$ and attach vertex i to it;
        $r(e) \leftarrow i$ *// $r(e)$ stores the root vertex associated with the entry e*
    else *// vertex i is not a peak of $\hat{f}$ within G*
        $g(i) \leftarrow \mathrm{argmax}_{j \in \mathcal{N}} \hat{f}(j)$ *// $g(i)$ stores the approximate gradient at vertex i*
        $e_i \leftarrow \mathcal{U}.\mathtt{find}(g(i))$;
        Attach vertex i to the entry e_i;

graph-based hill-climbing (1976)

        for $j \in \mathcal{N}$ **do**
            $e \leftarrow \mathcal{U}.\mathtt{find}(j)$;
            if $e \neq e_i$ and $\min\{\hat{f}(r(e)),\ \hat{f}(r(e_i))\} < \hat{f}(i) + \tau$
                $\mathcal{U}.\mathtt{union}(e,\ e_i)$;
                $r(e \cup e_i) \leftarrow \mathrm{argmax}_{\{r(e),\ r(e_i)\}} \hat{f}$;
                $e_i \leftarrow e \cup e_i$;

cluster merges with persistence (2013)

Output: the collection of entries e of $\mathcal{U}$ such that $\hat{f}(r(e)) \geq \tau$.

Complexity of the Algorithm

Given a neighborhood graph with n vertices (with density estimates) and m edges:

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

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

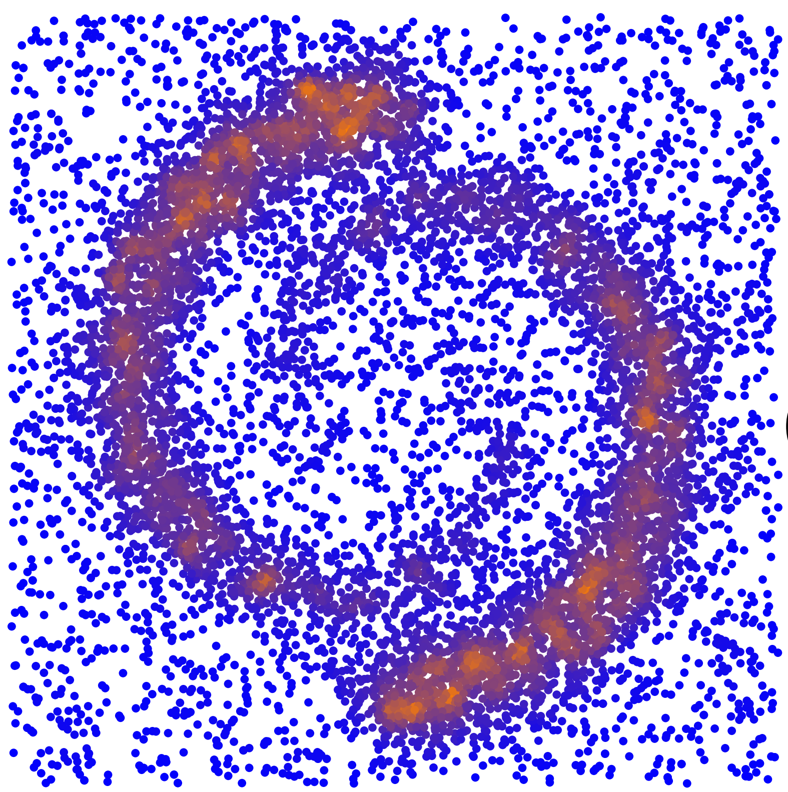
$\rightarrow$ Running time: $O(n \log n + (n+m)\alpha(n))$

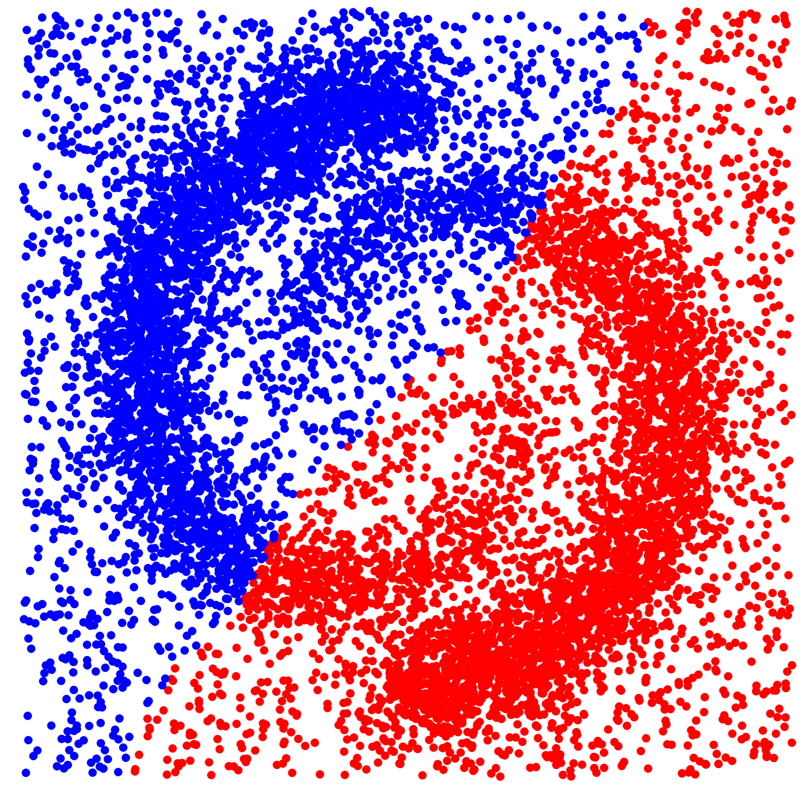
$\rightarrow$ Space complexity: $O(n+m)$

$\rightarrow$ Main memory usage: $O(n)$

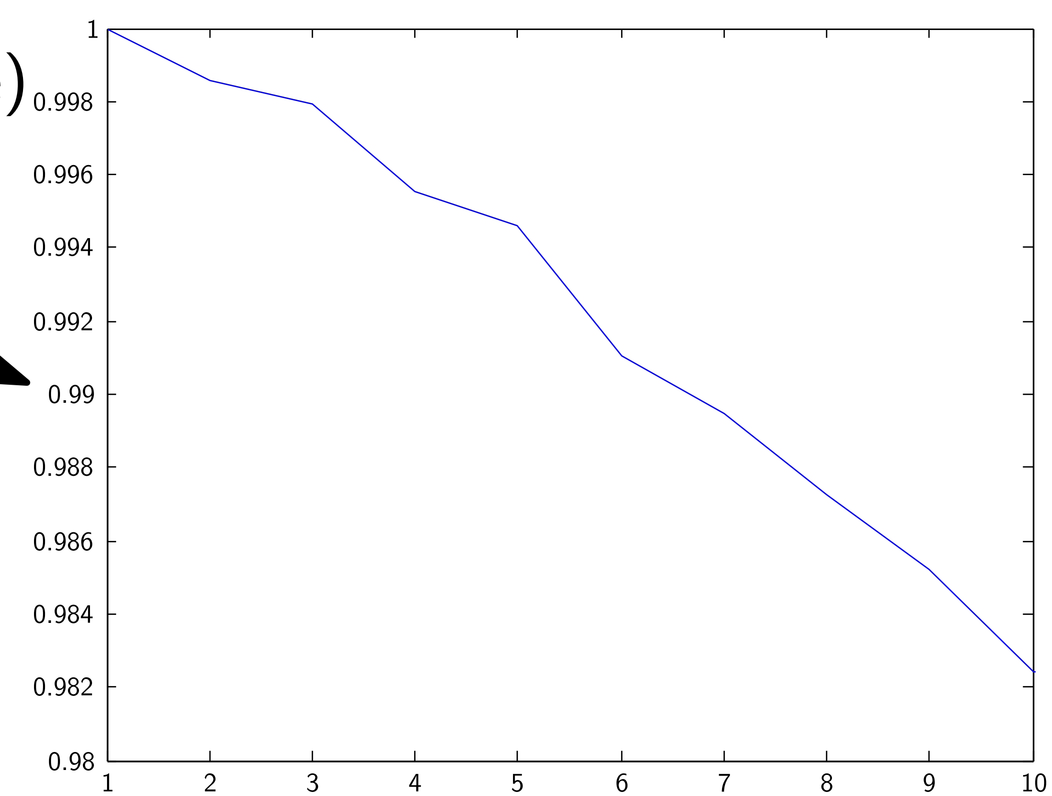
Experimental Results

Synthetic Data



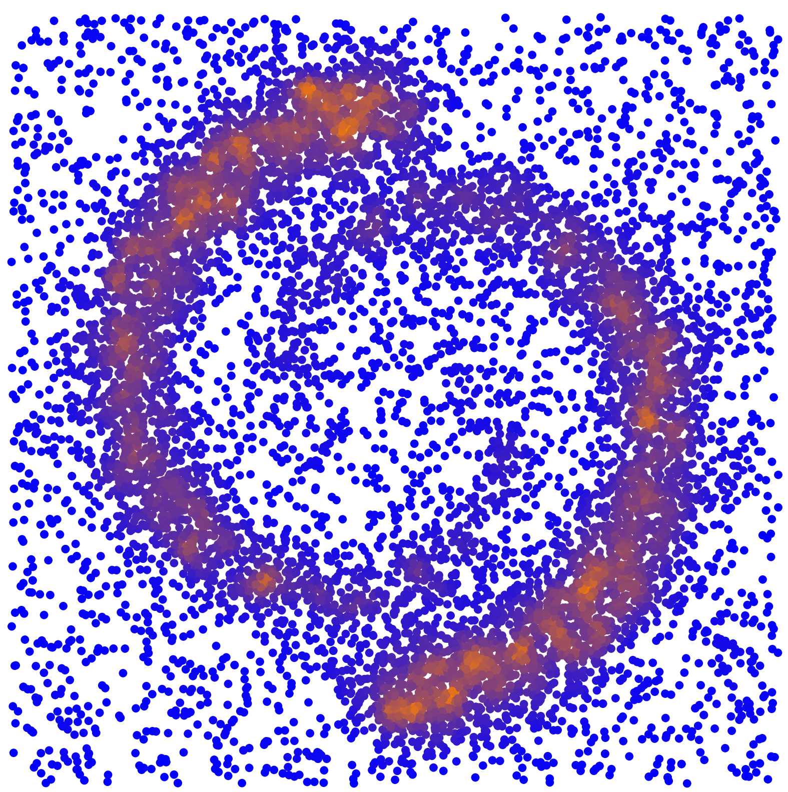


Spectral clustering
(k-means in eigenspace)



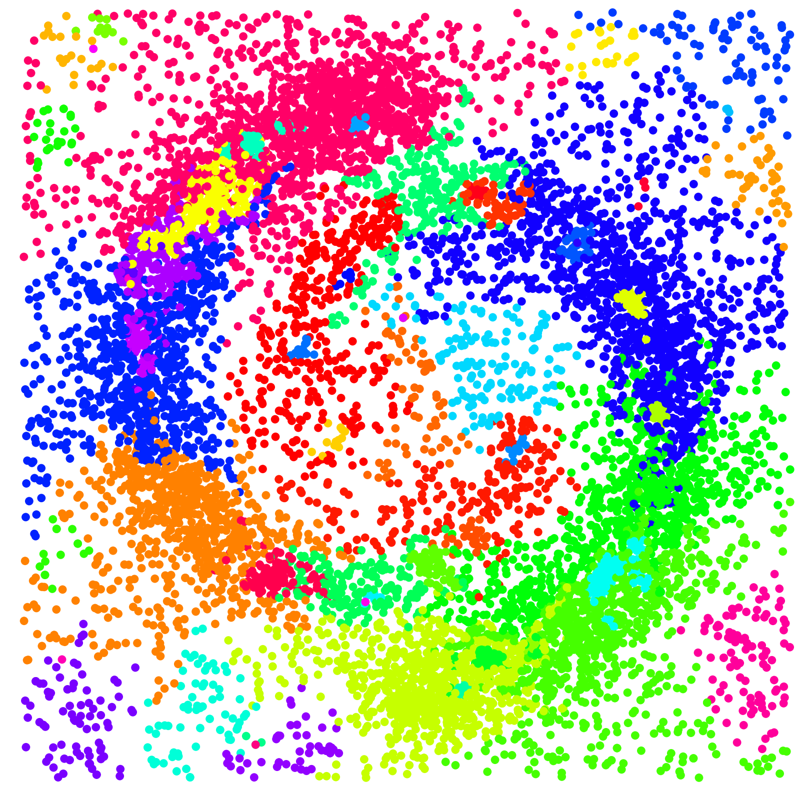
Experimental Results

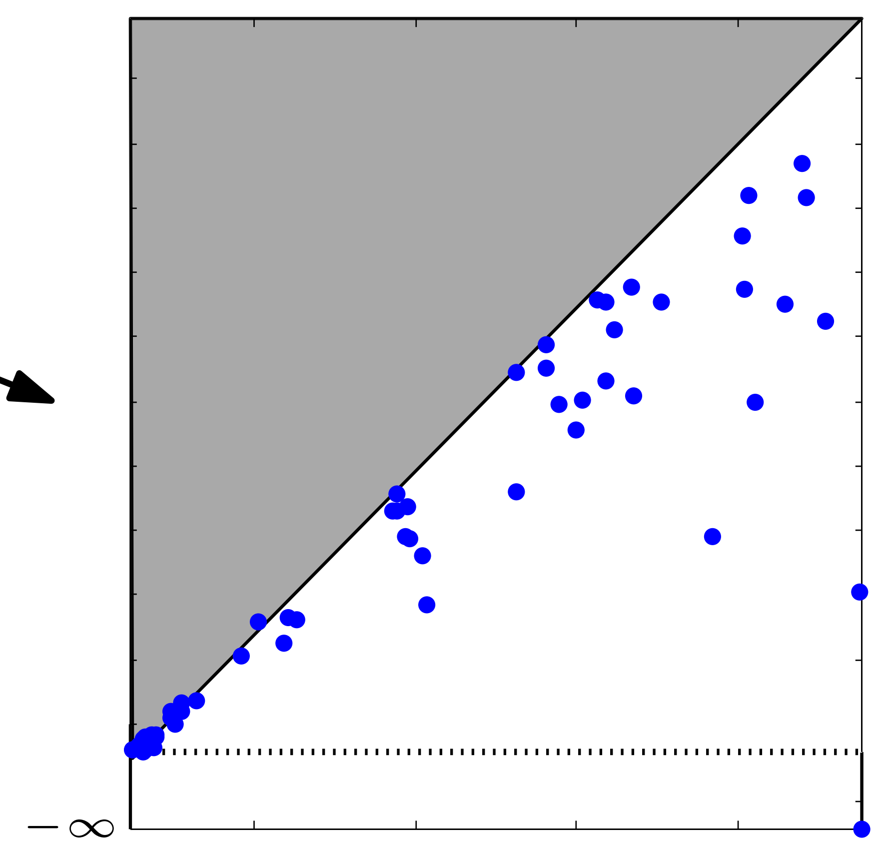
Synthetic Data



$\tau = 0$

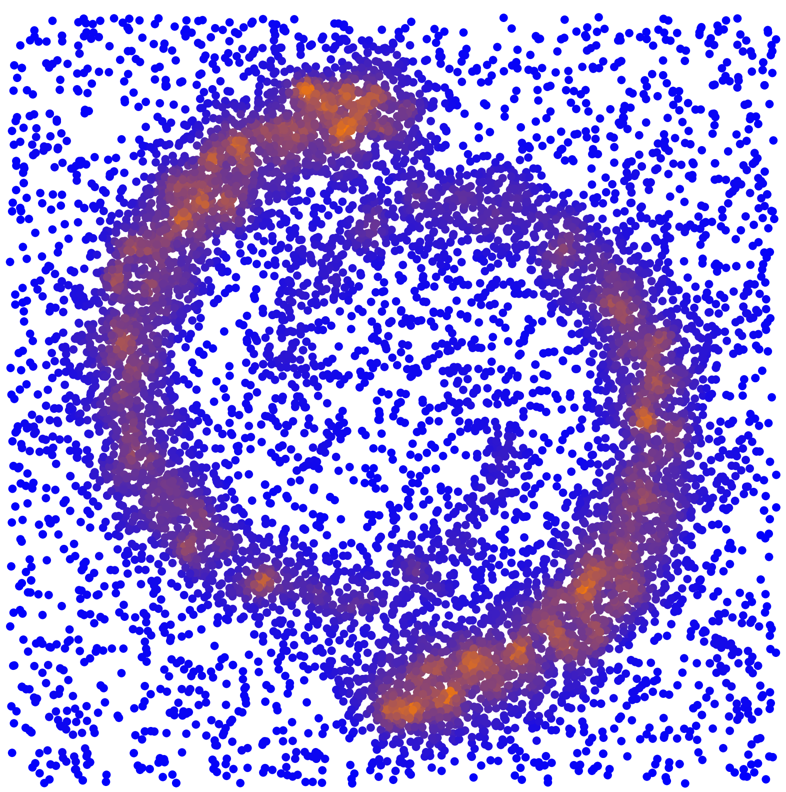
ToMATo

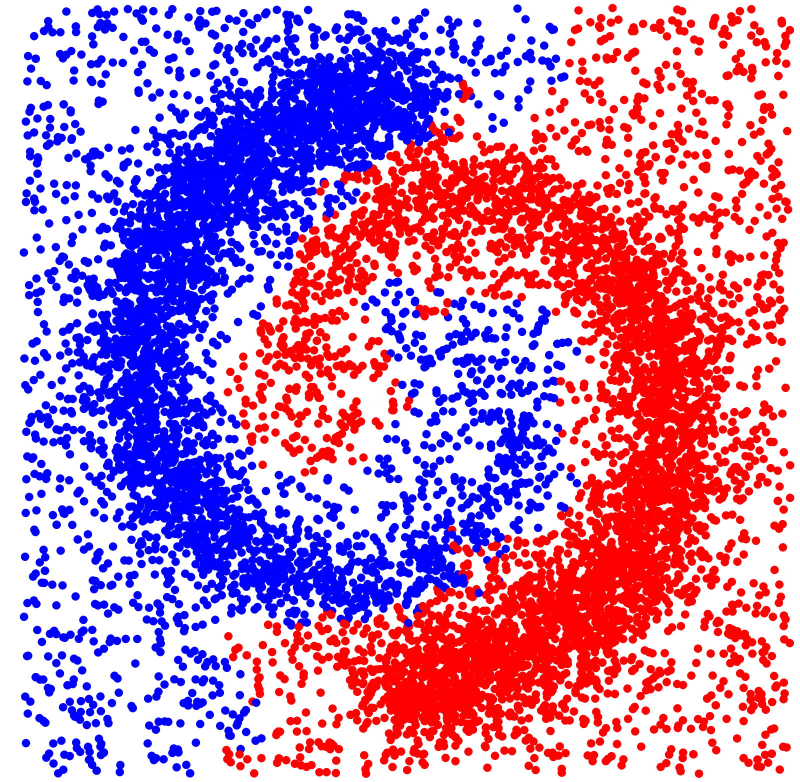




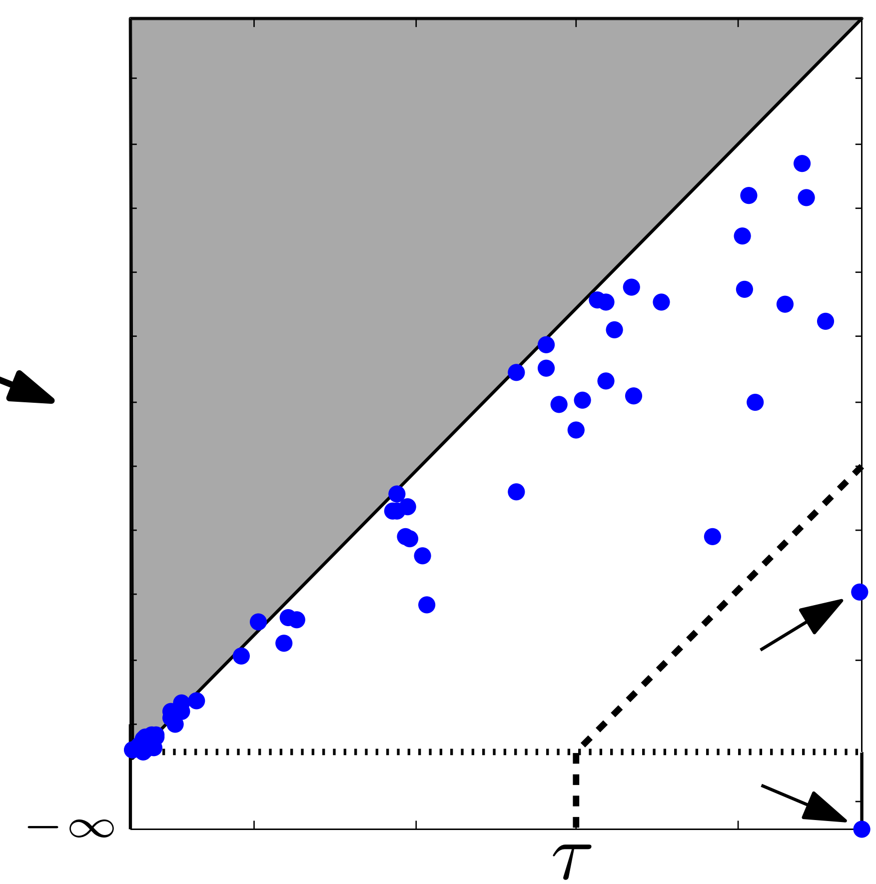
Experimental Results

Synthetic Data



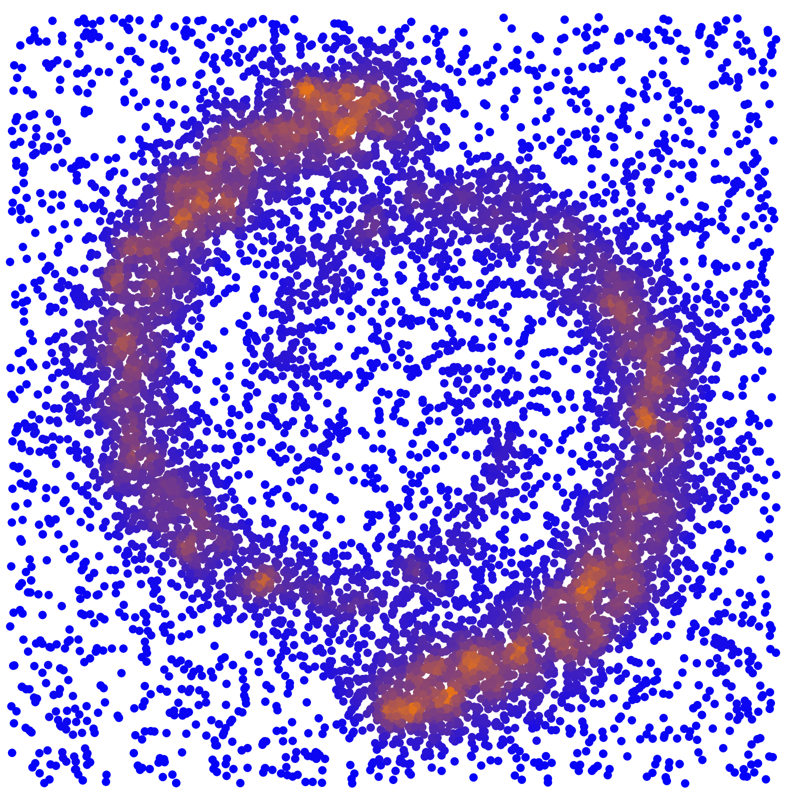


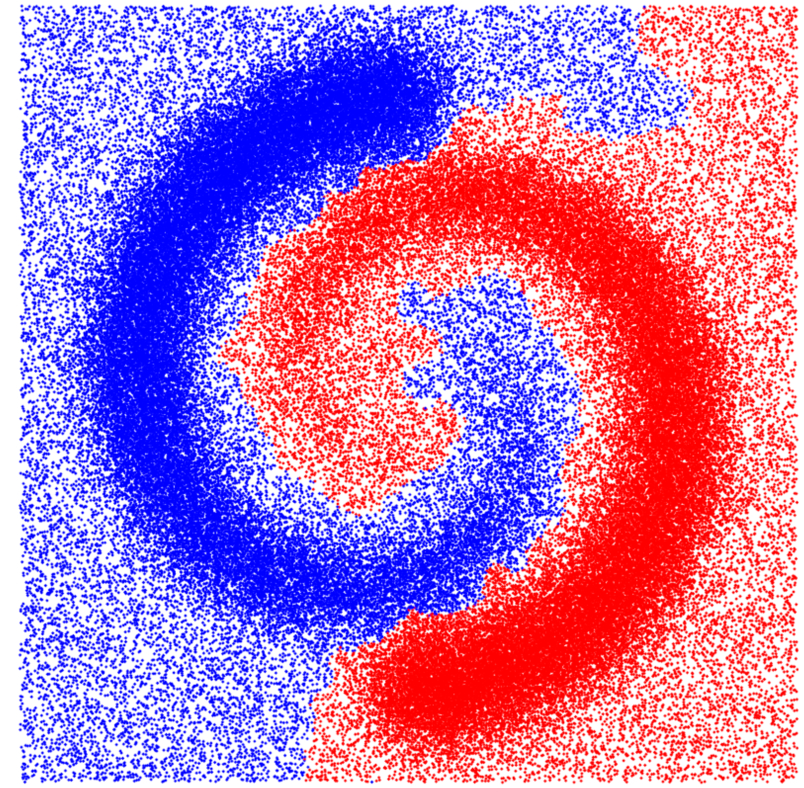
ToMATo

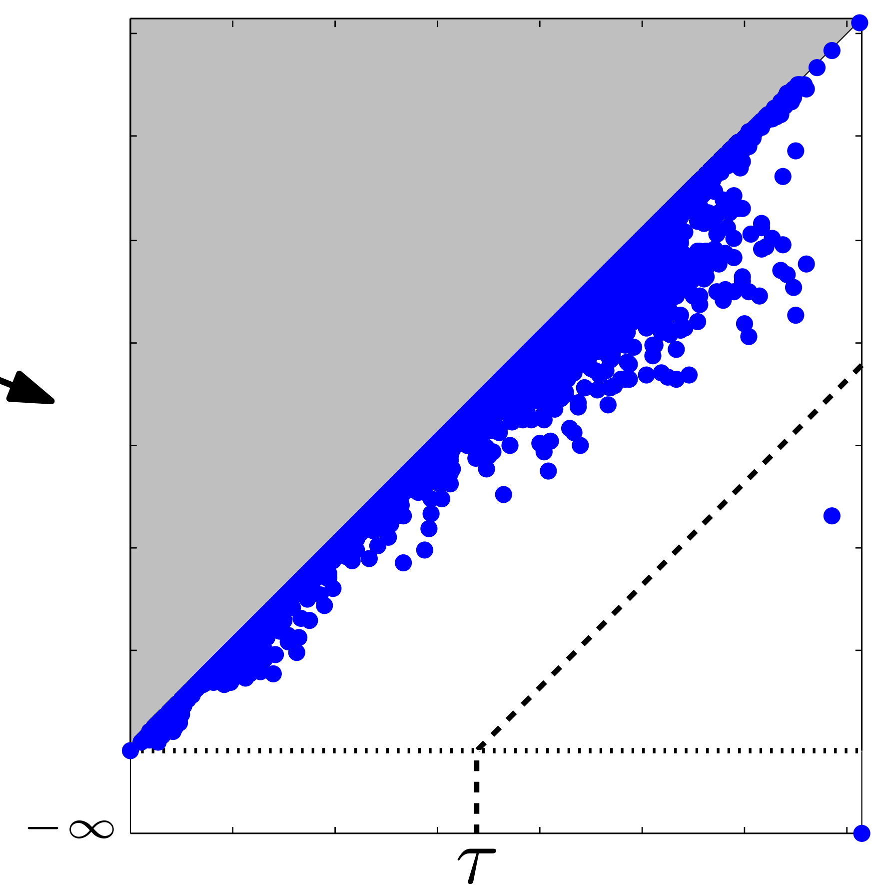


Experimental Results

Synthetic Data





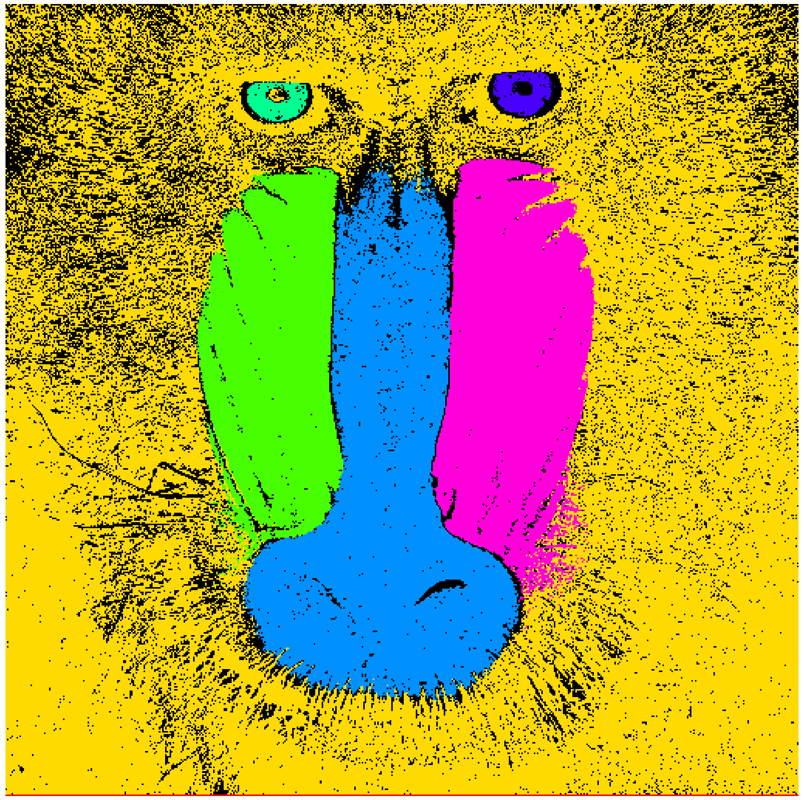


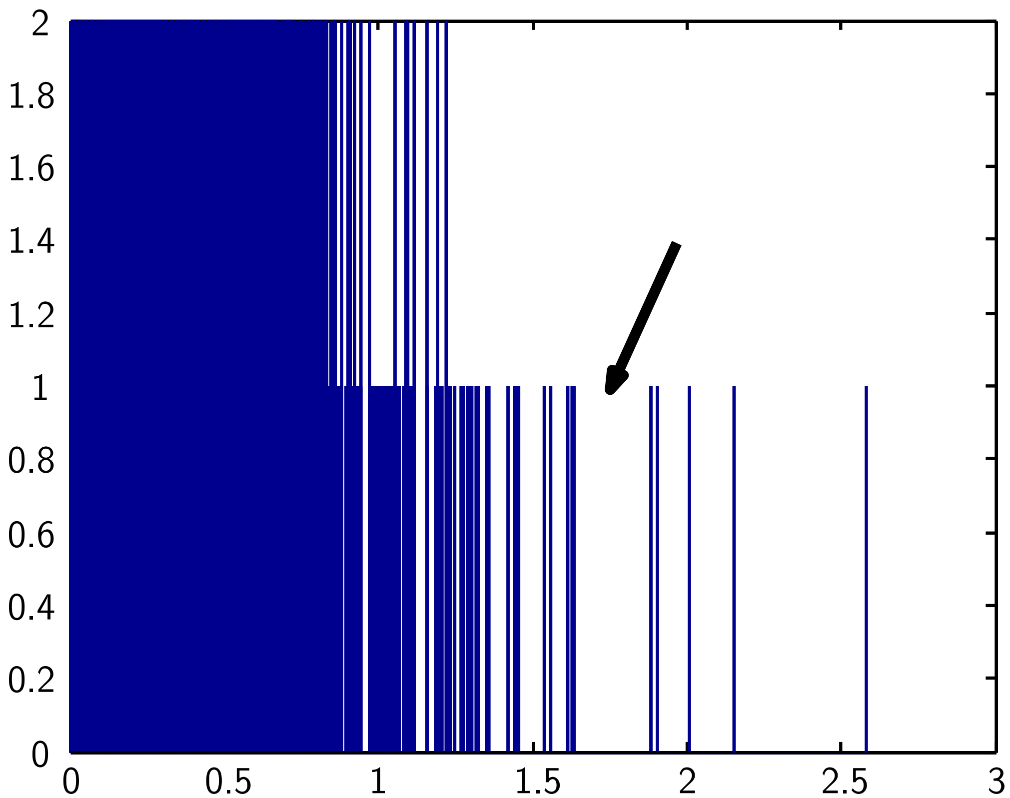
Experimental Results

Image Segmentation

Density is estimated in 3D color space (Luv)

Neighborhood graph is built in image domain





Distribution of prominences does not usually show a clear unique gap

Still, relationship between choice of τ and number of obtained clusters remains explicit

Estimating the Correct Number of Clusters

Hypotheses:

- $f : \mathbb{X} \to \mathbb{R}$ a c-Lipschitz probability density function,
- P a finite set of n points of $\mathbb{X}$ sampled i.i.d. according to f,
- $\hat{f} : P \to \mathbb{R}$ a density estimator such that $\eta := \max_{p \in P} |\hat{f}(p) - f(p)| < \Pi/5$,
- $G = (P, E)$ 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:

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

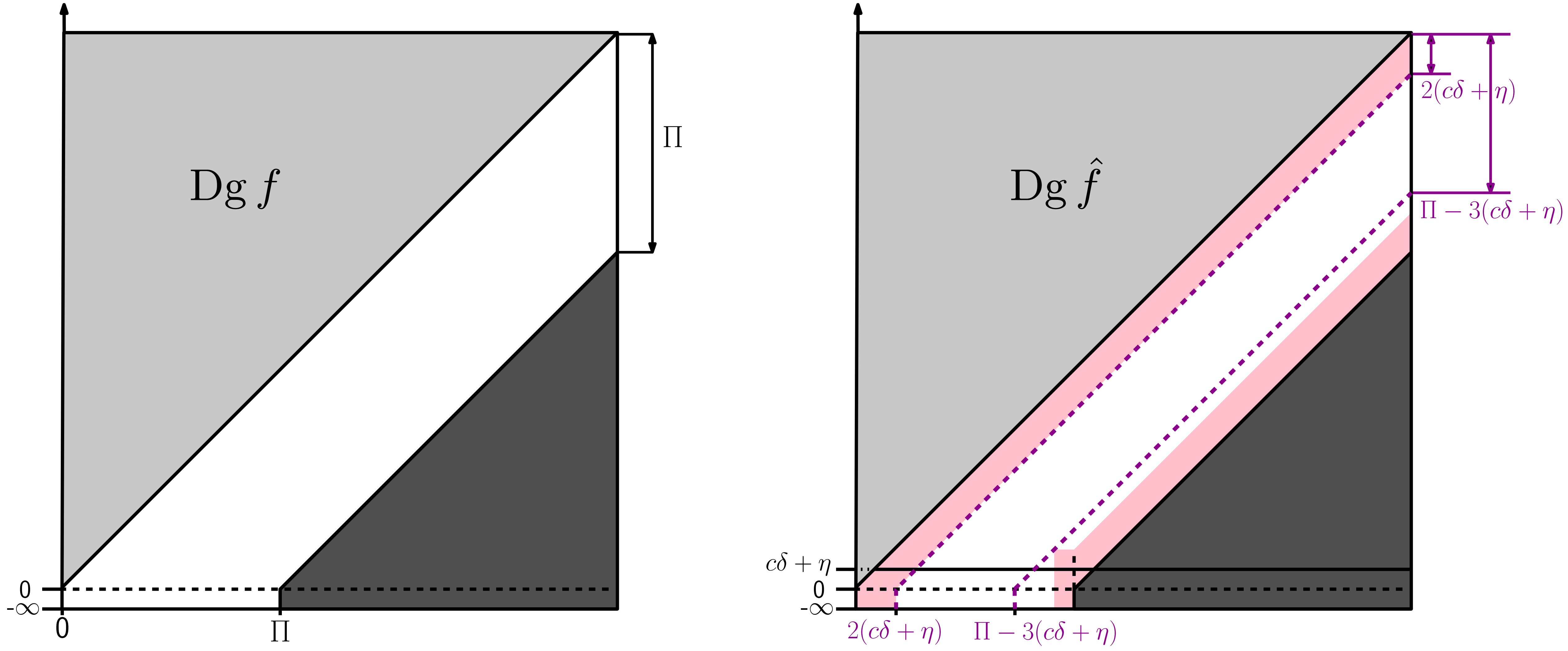
Note: Π is the prominence of the least prominent peak of f

Conclusion:

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

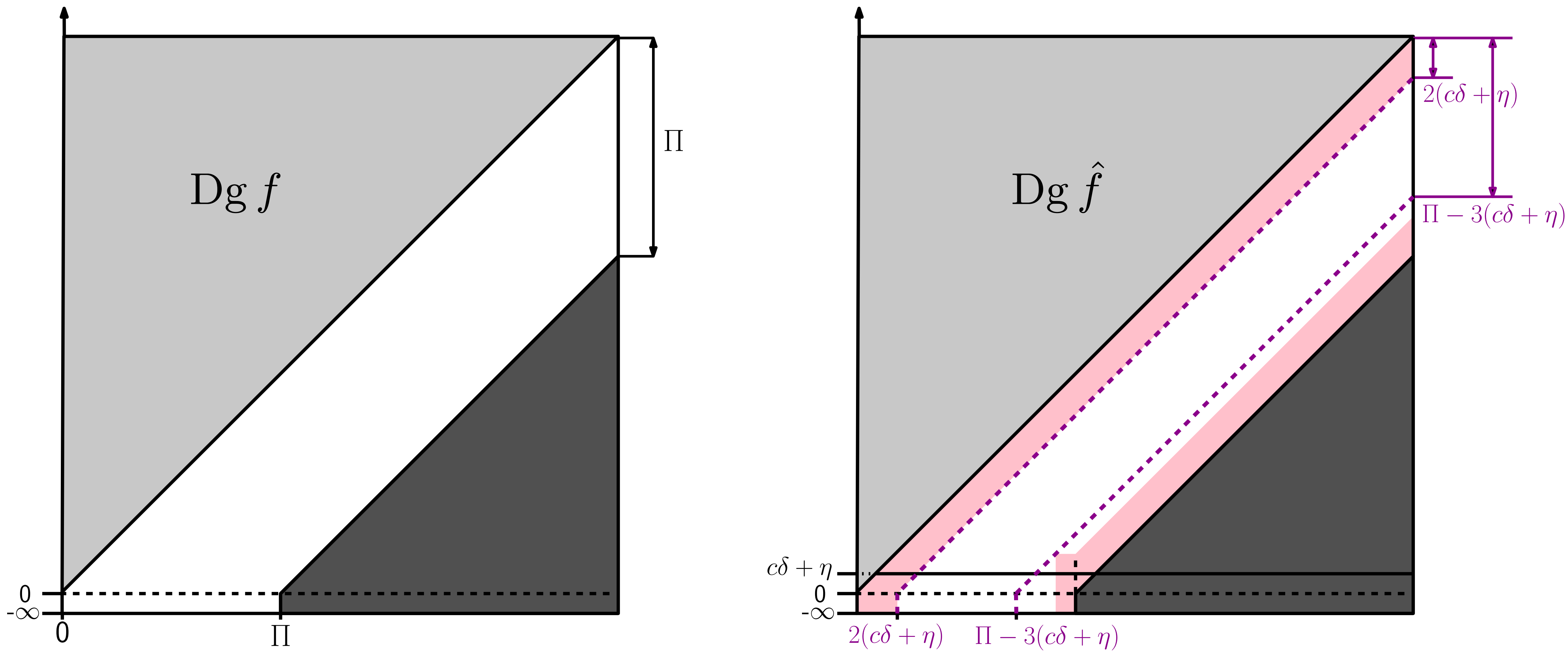


Conclusion:

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



Proof's main ingredient: stability theorem for persistence diagrams

Note: $f, \hat{f}$ are not defined over the same domain