Topological Inference via Meshing

Benoit Hudson, Gary Miller, Steve Oudot and Don Sheehy

SoCG 2010

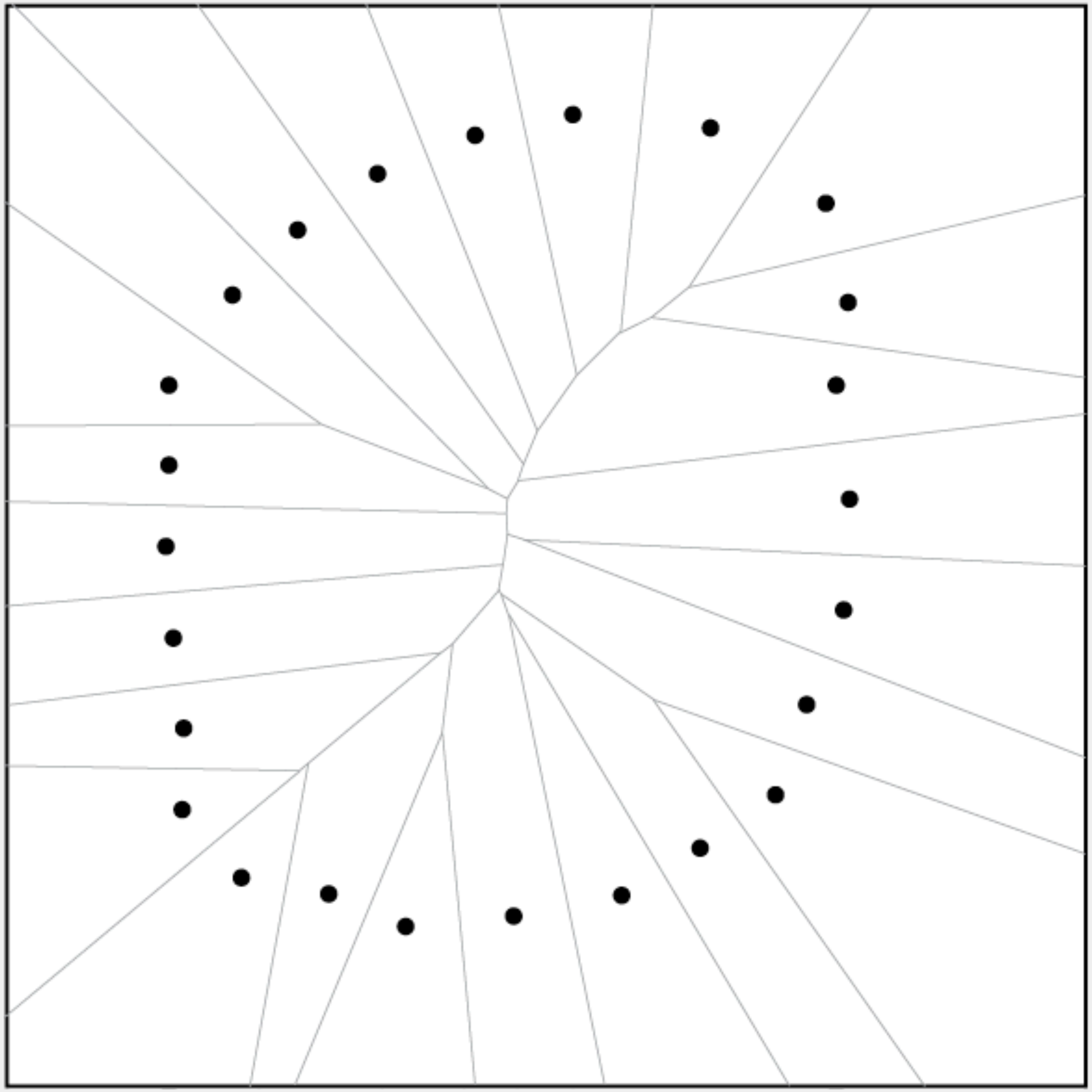
Mesh Generation and Persistent Homology

The Problem

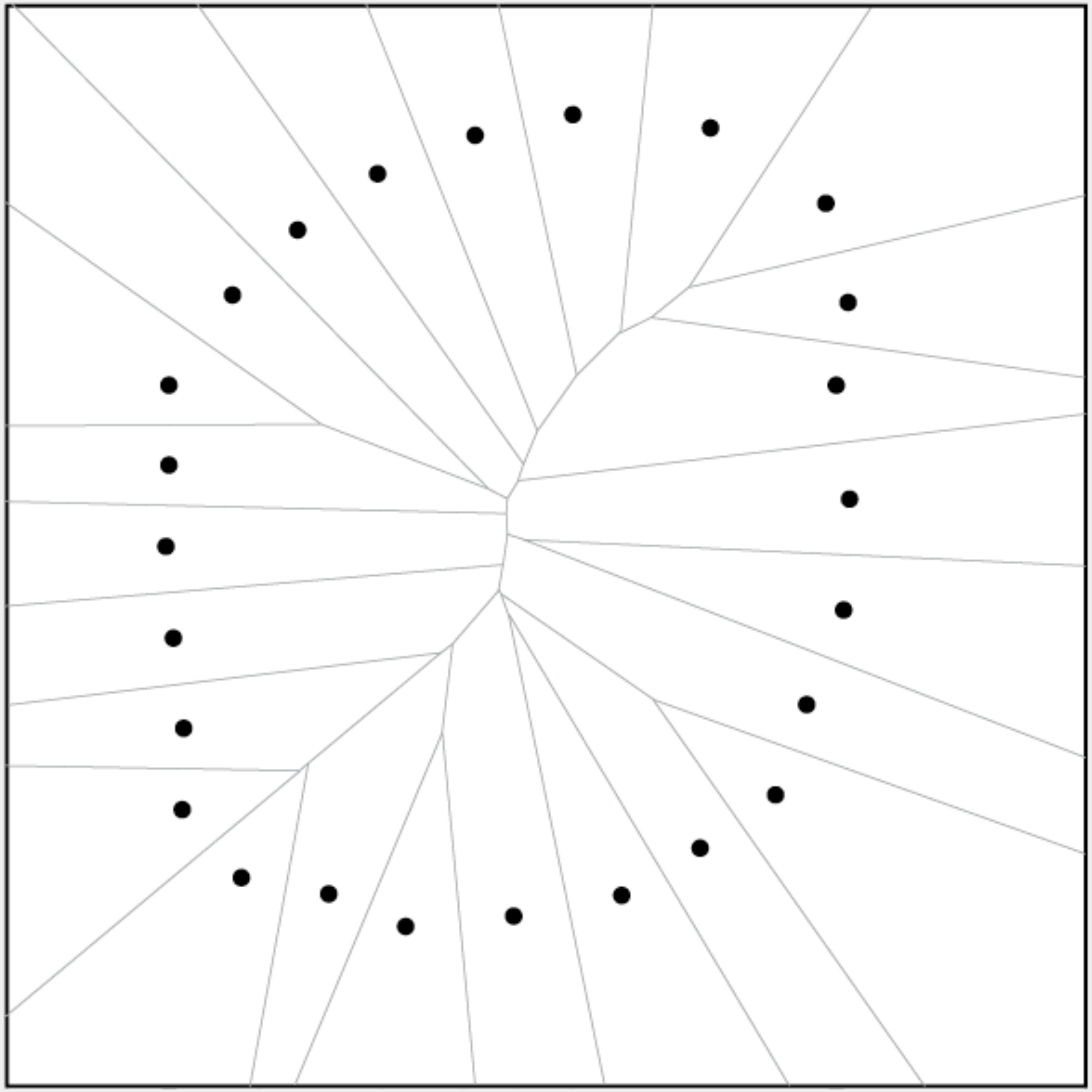
Input: Points in Euclidean space sampled from some unknown object.

Output: Information about the topology of the unknown object.

Points, offsets, homology, and persistence.

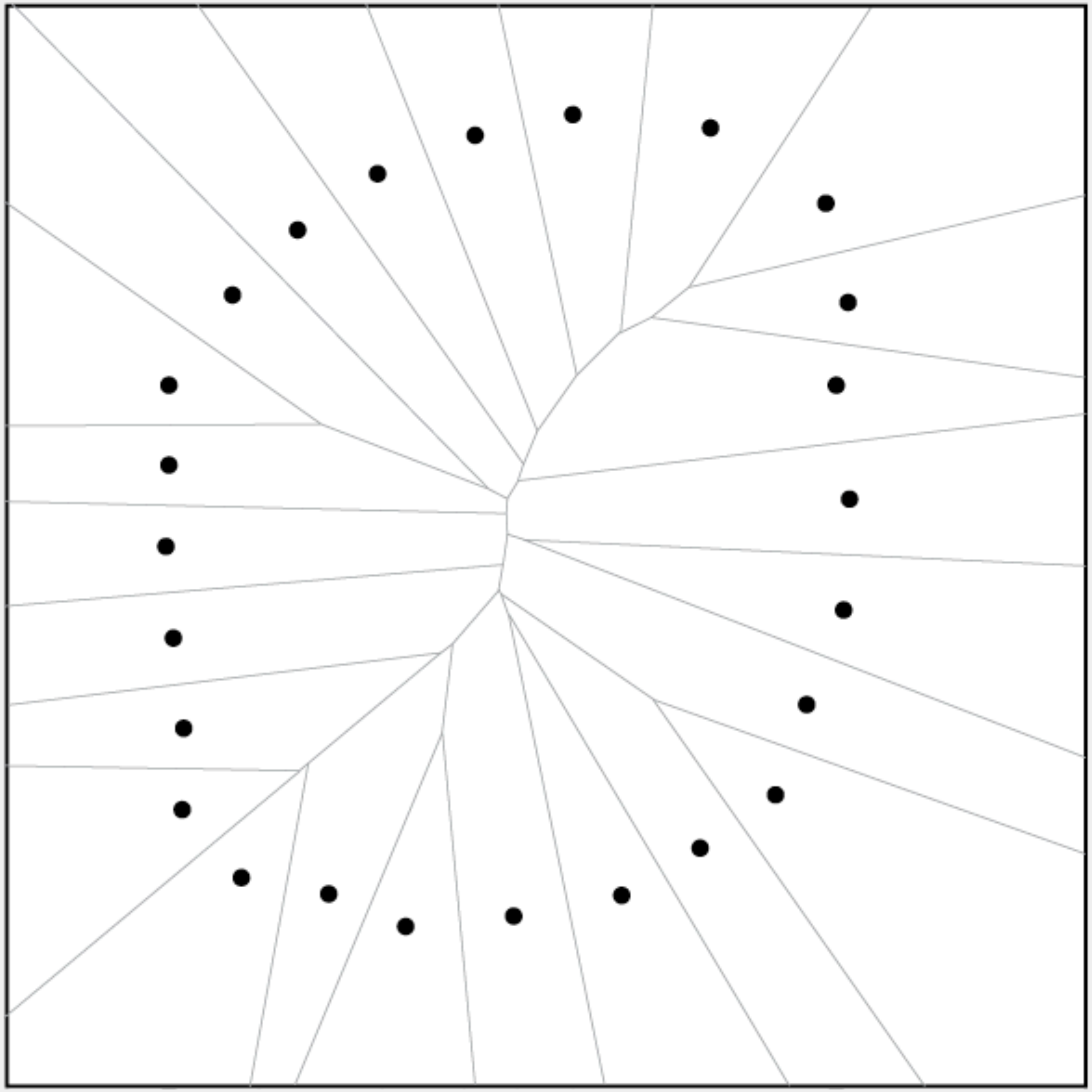


Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

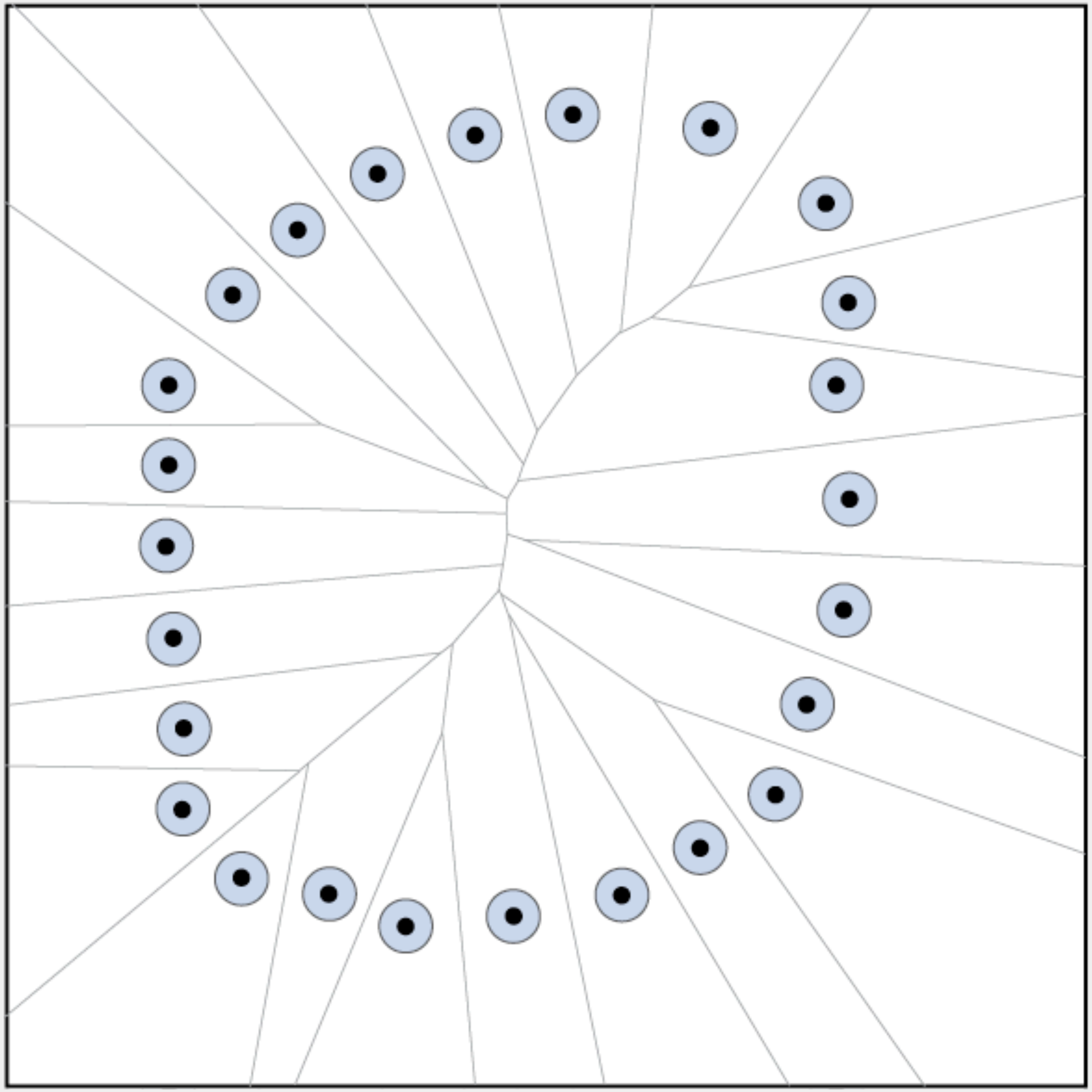
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \operatorname{ball}(p, \alpha)$$

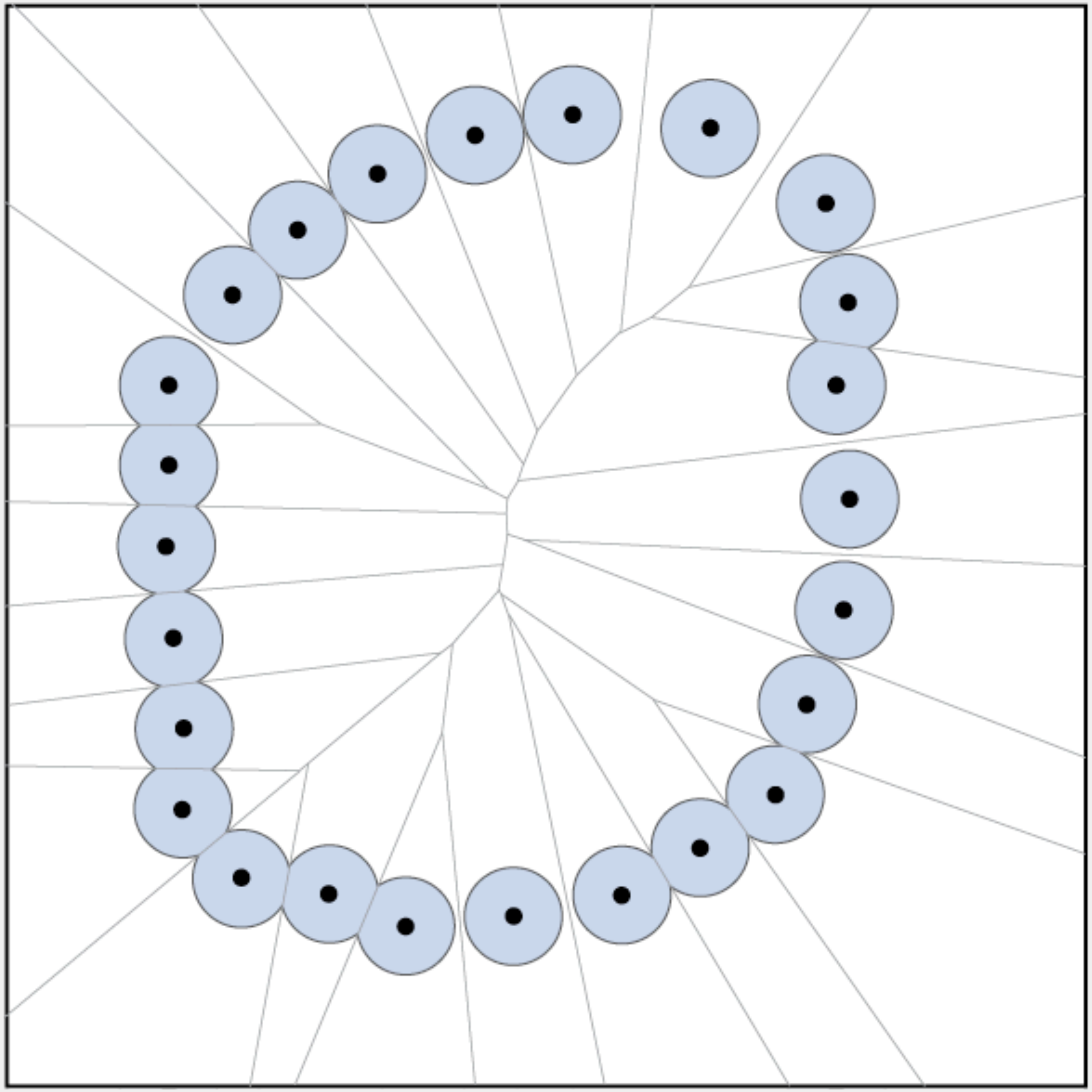
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \text{ball}(p, \alpha)$$

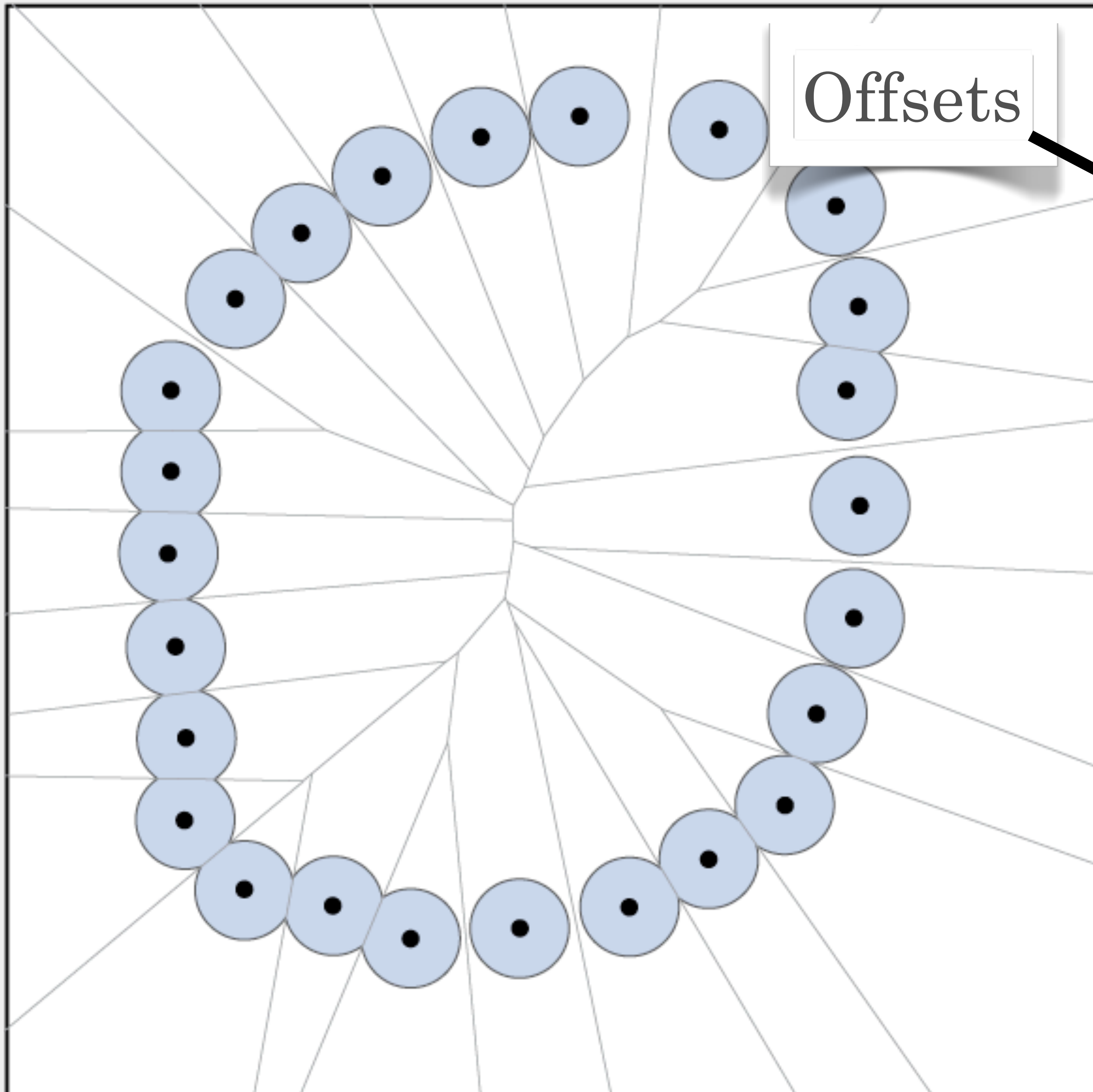
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

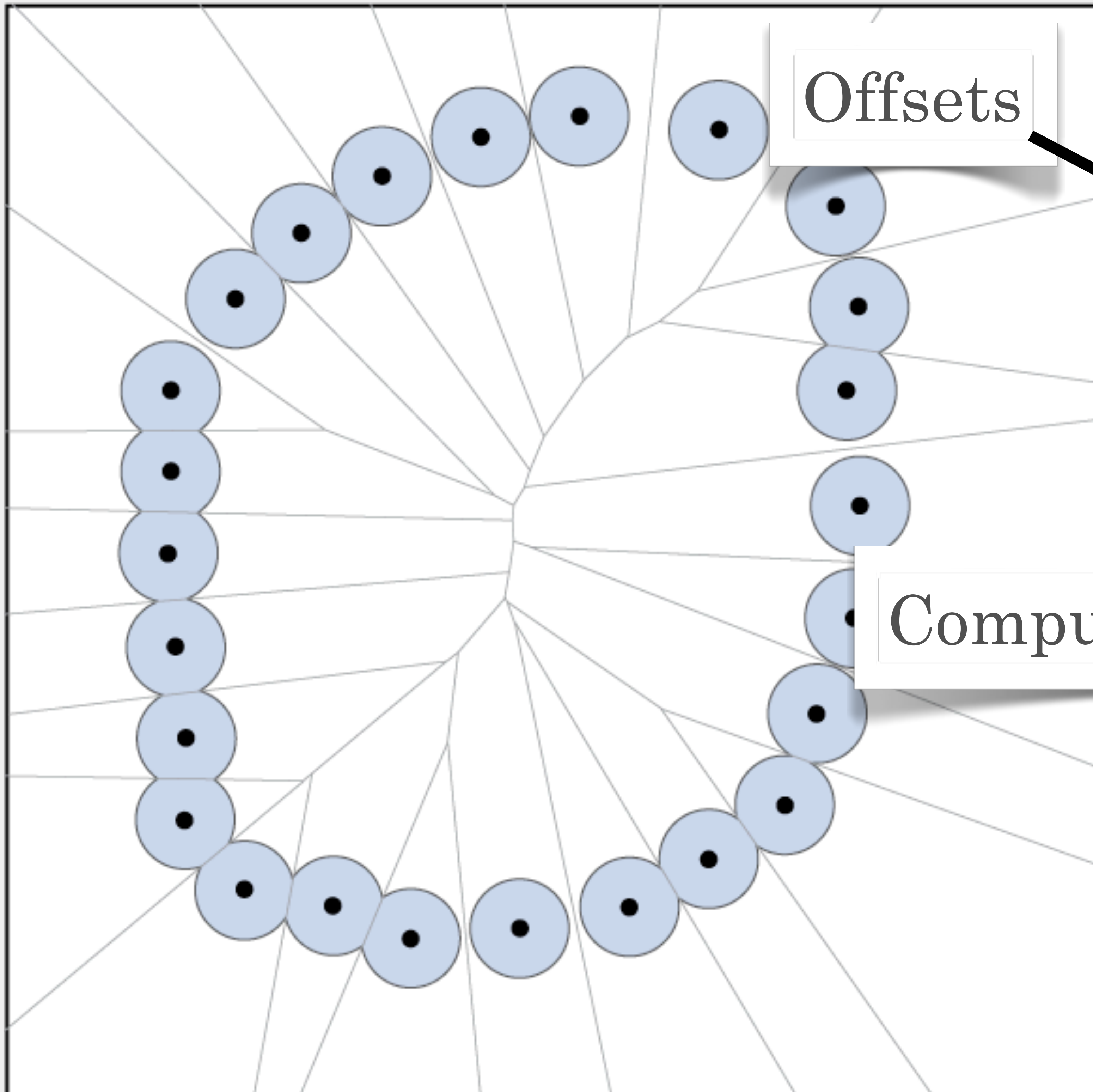
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

Points, offsets, homology, and persistence.



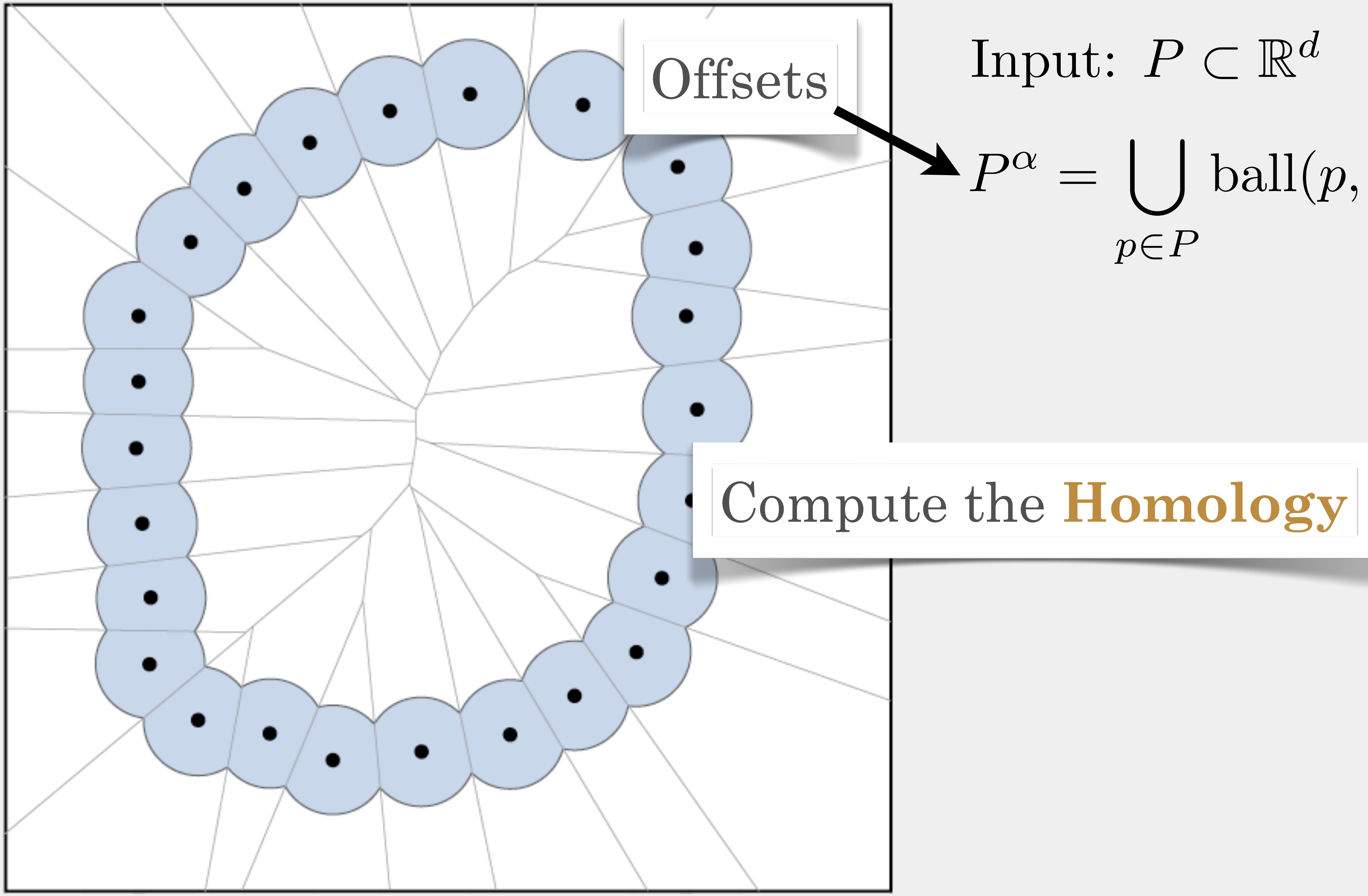
Offsets

Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \text{ball}(p, \alpha)$$

Compute the **Homology**

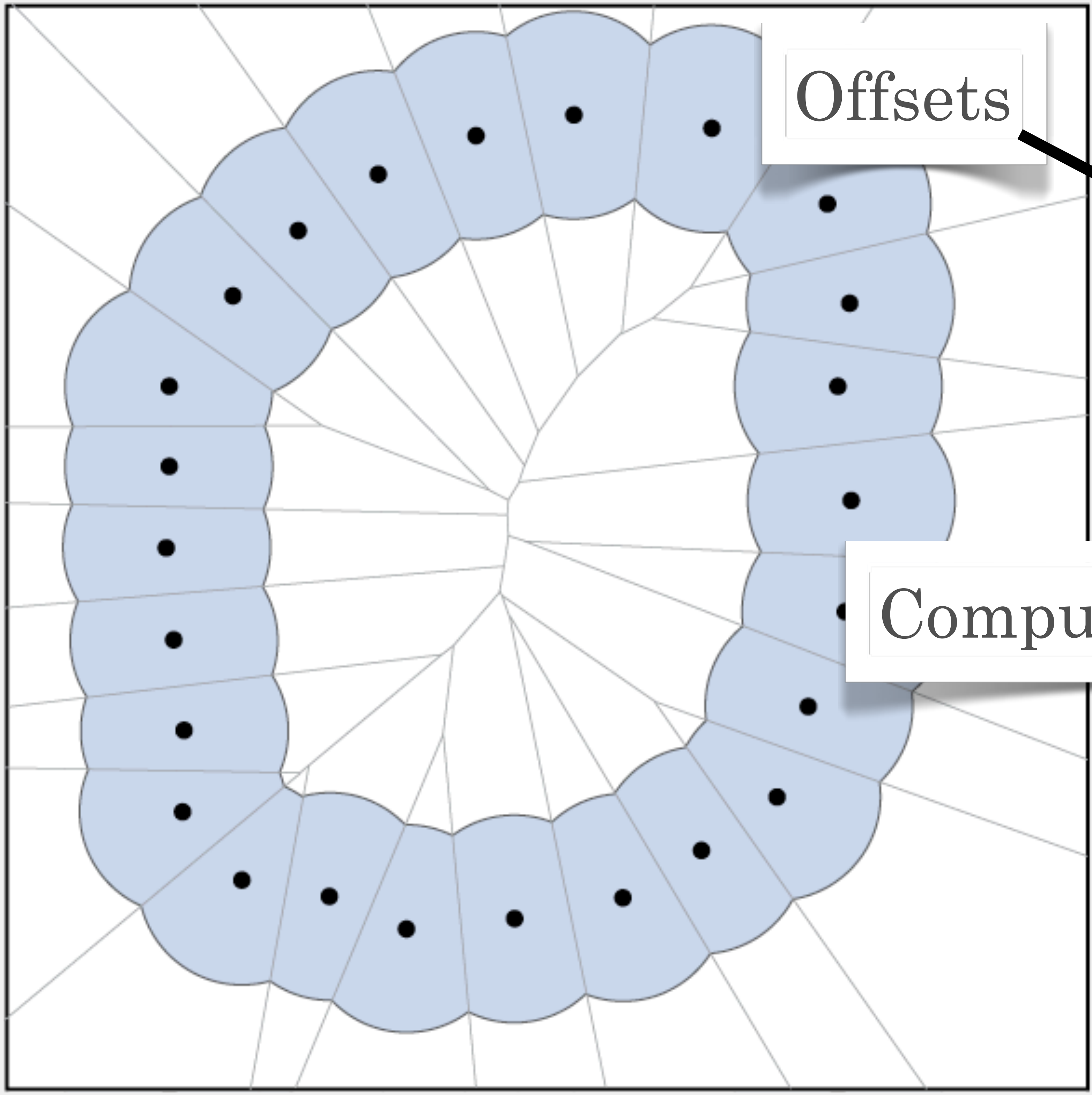
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \text{ball}(p, \alpha)$$

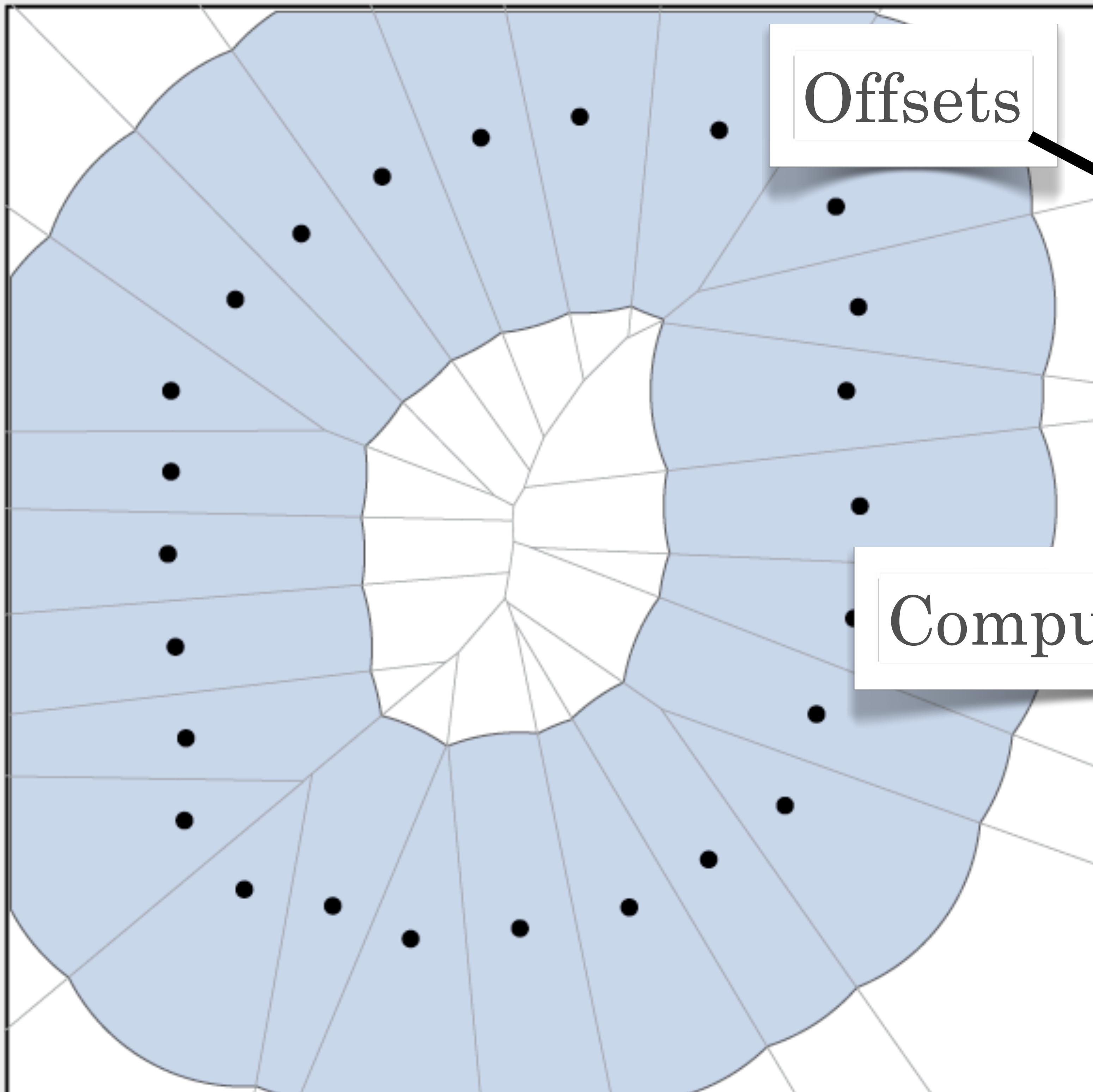
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

Points, offsets, homology, and persistence.



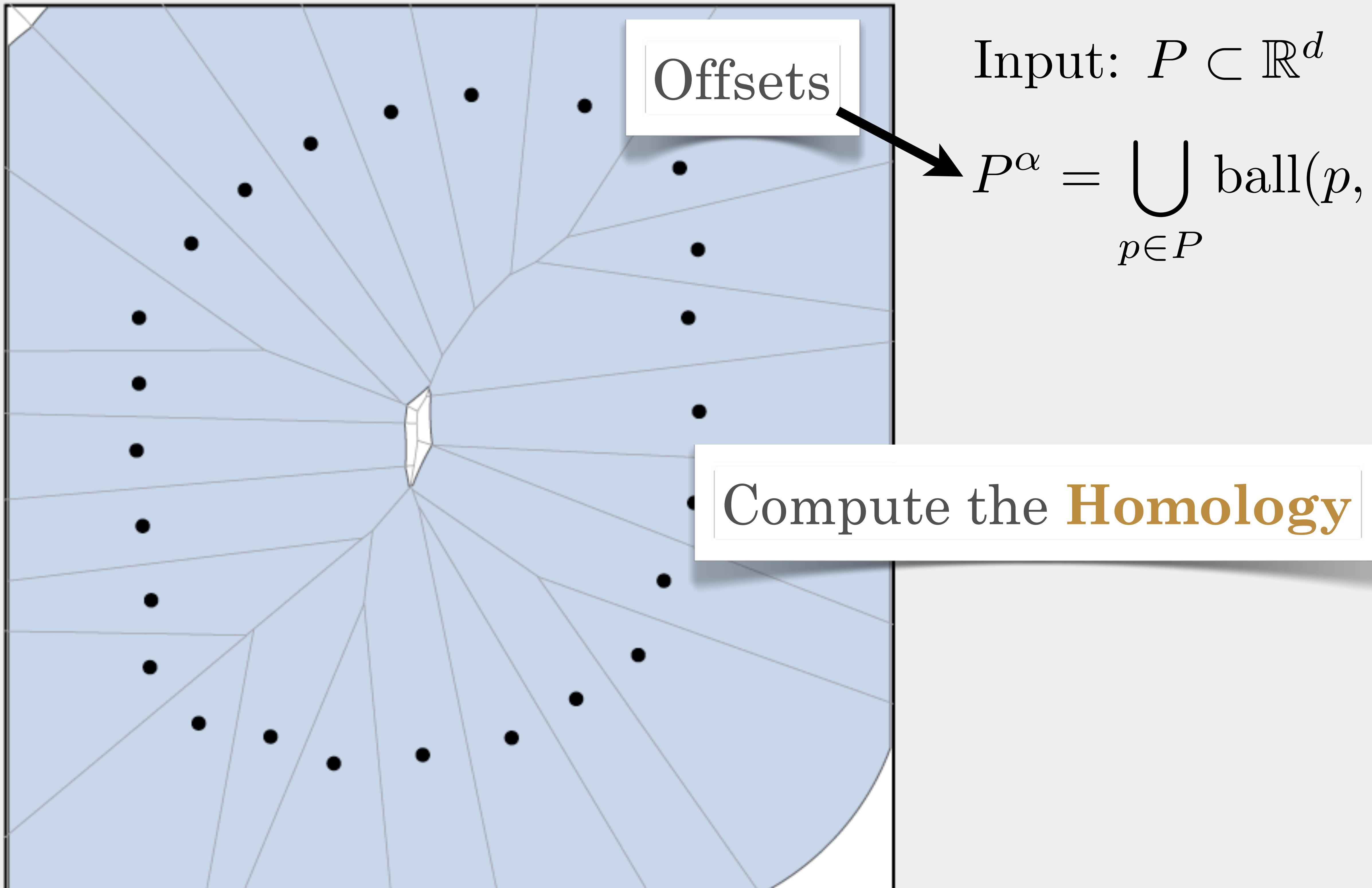
Offsets

Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

Compute the **Homology**

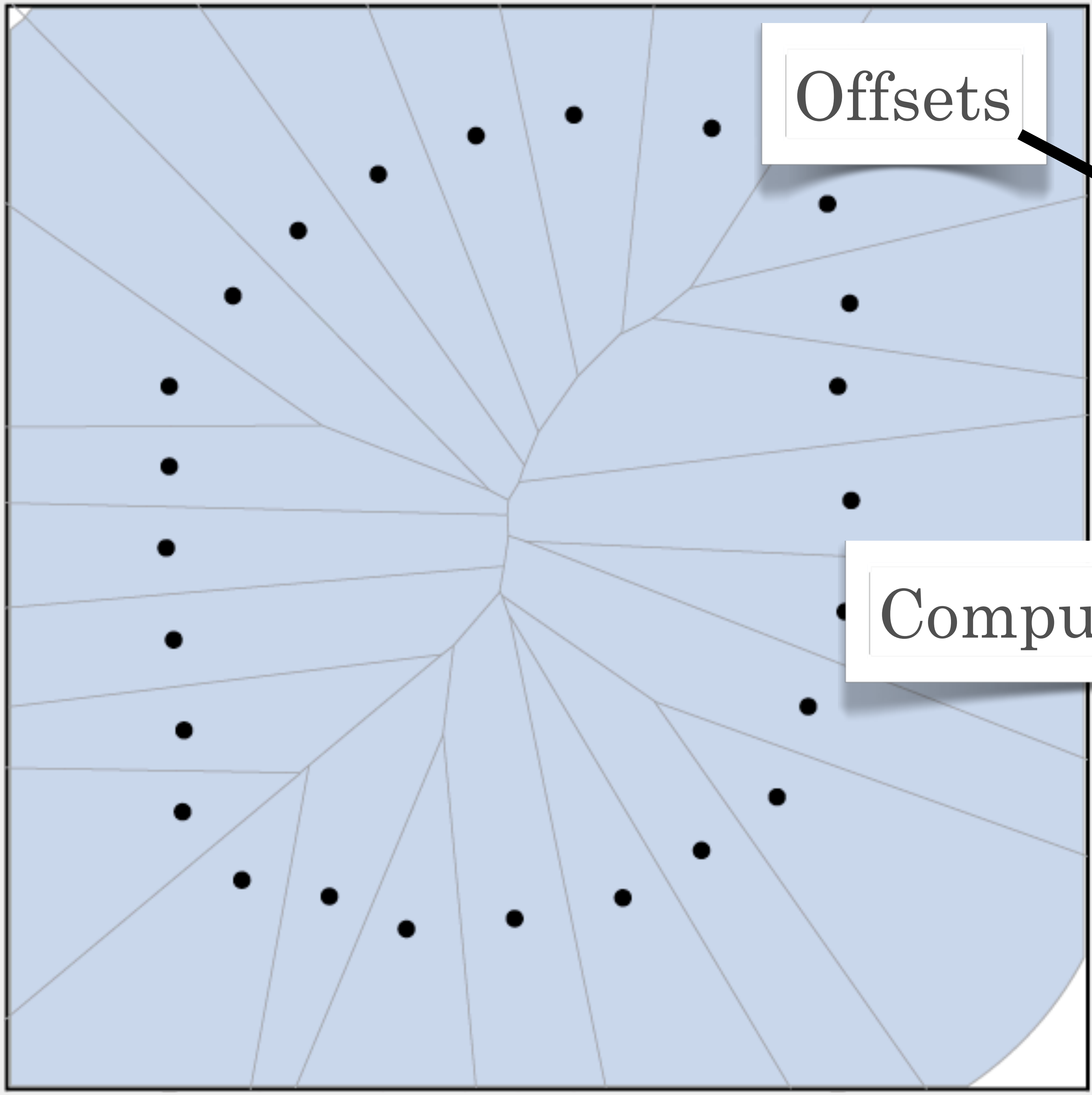
Points, offsets, homology, and persistence.



Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

Points, offsets, homology, and persistence.



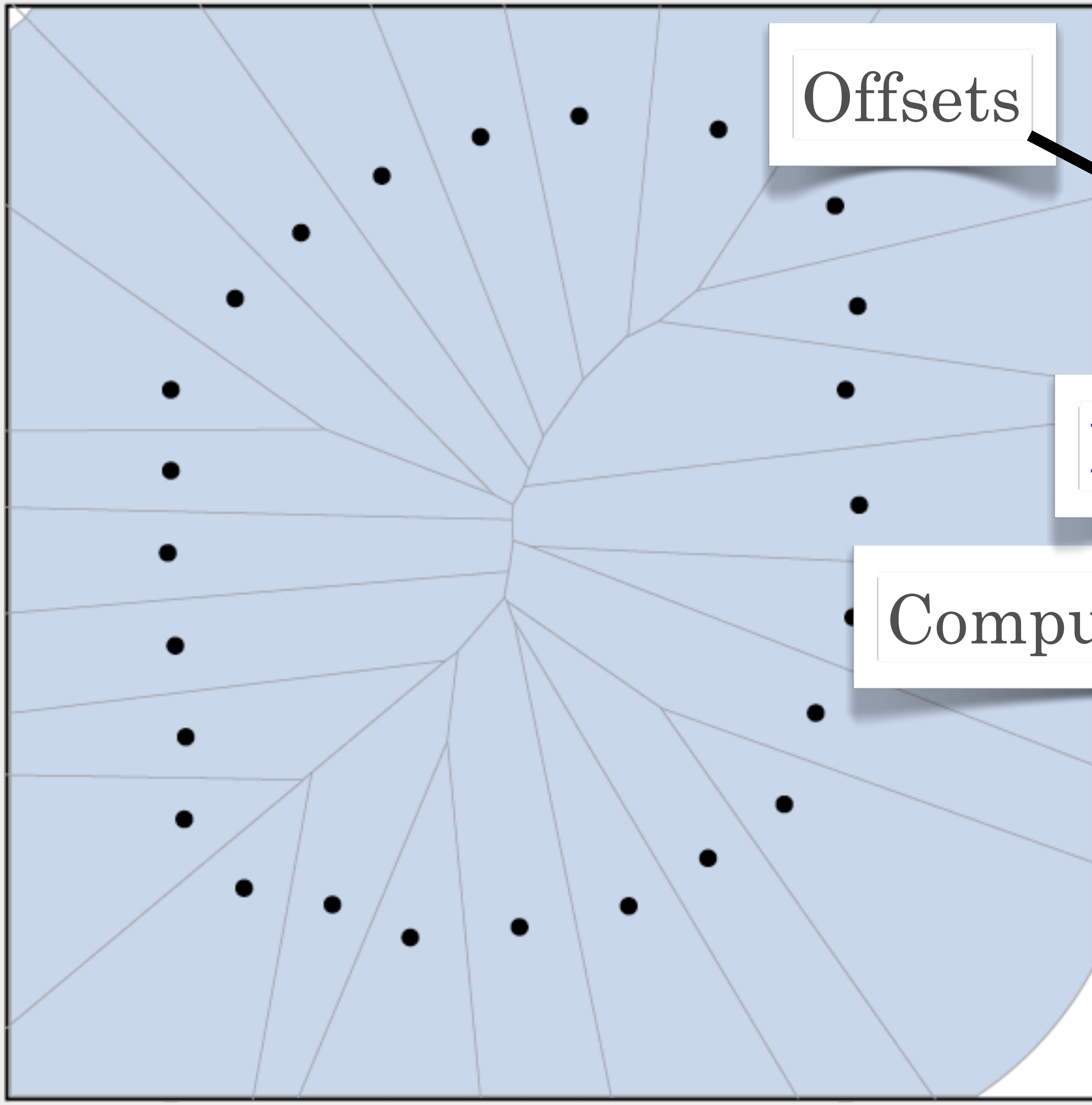
Offsets

Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \mathrm{ball}(p, \alpha)$$

Compute the **Homology**

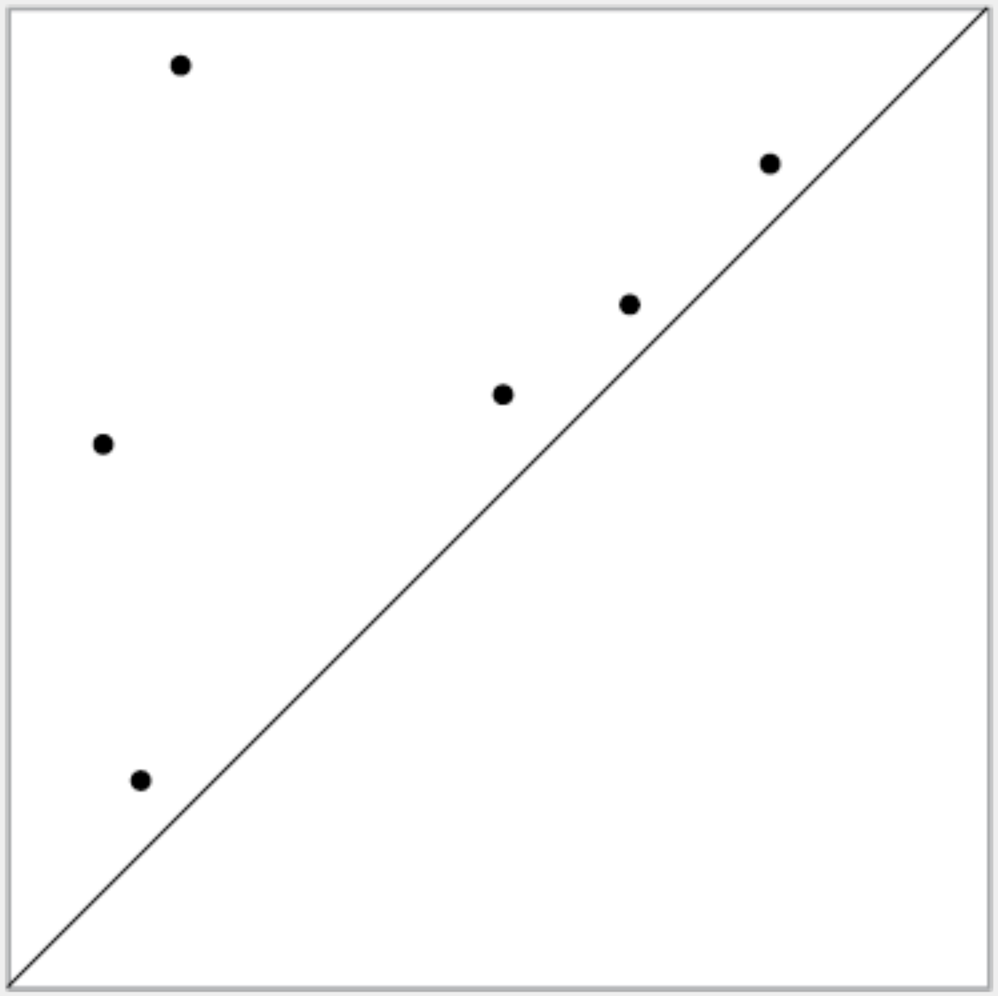
Points, offsets, homology, and persistence.



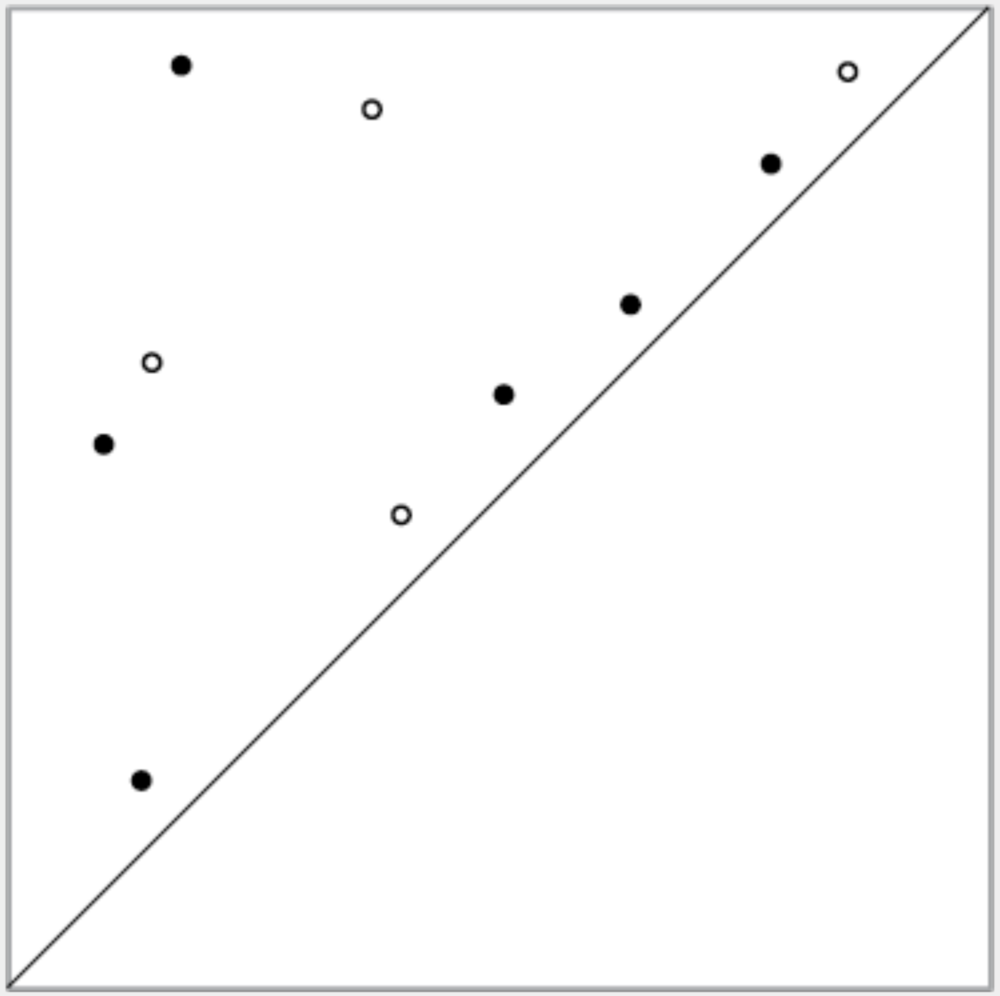
Input: $P \subset \mathbb{R}^d$

$$P^\alpha = \bigcup_{p \in P} \text{ball}(p, \alpha)$$

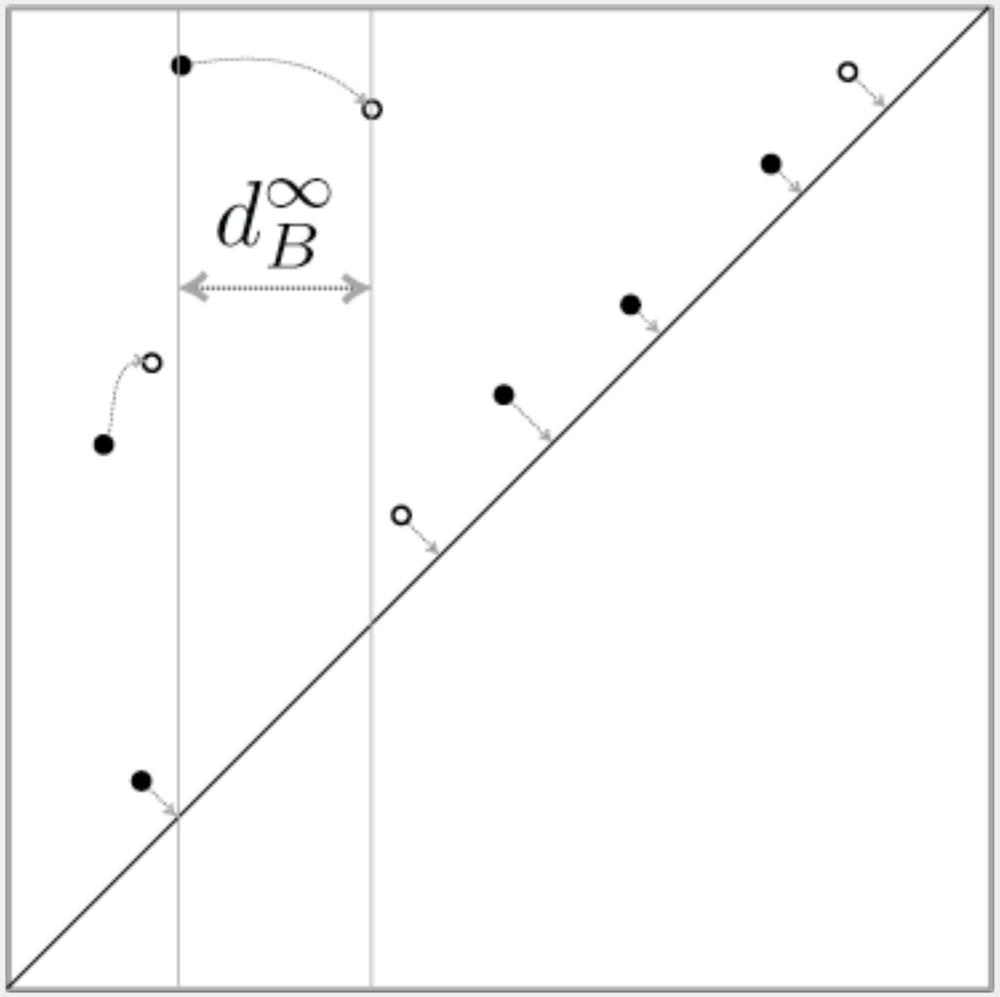
Persistence Diagrams



Persistence Diagrams



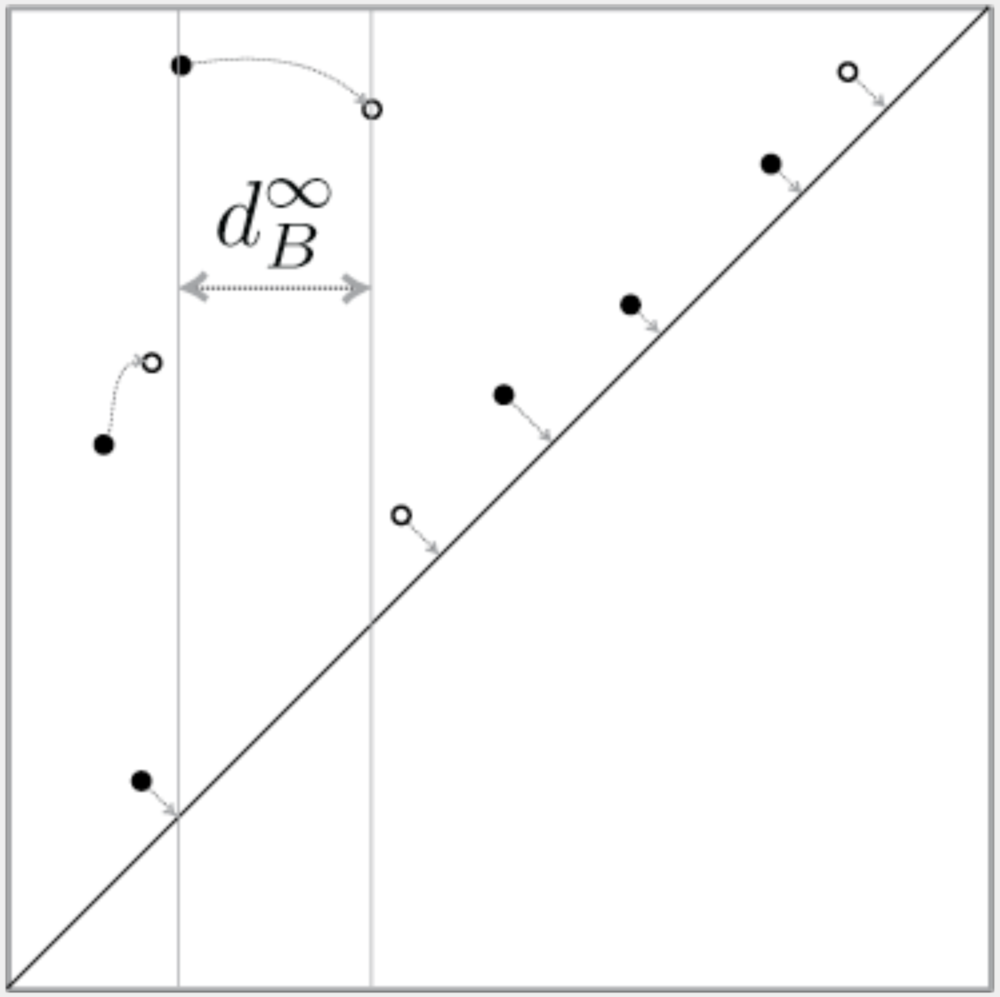
Persistence Diagrams



Bottleneck Distance

$$d_B^\infty = \max_i |p_i - q_i|_\infty$$

Approximate Persistence Diagrams

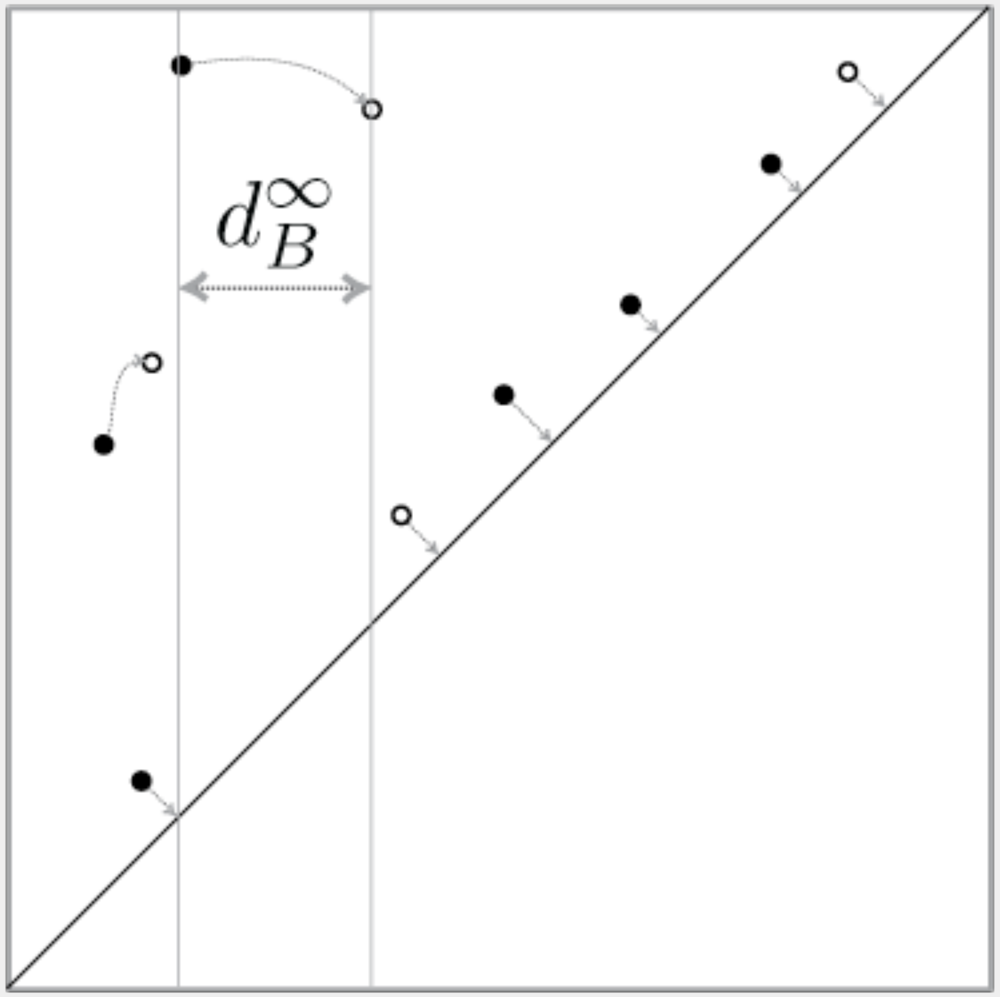


Bottleneck Distance

$$d_B^\infty = \max_i |p_i - q_i|_\infty$$

Approximate Persistence Diagrams

Birth and Death times differ by a constant factor.

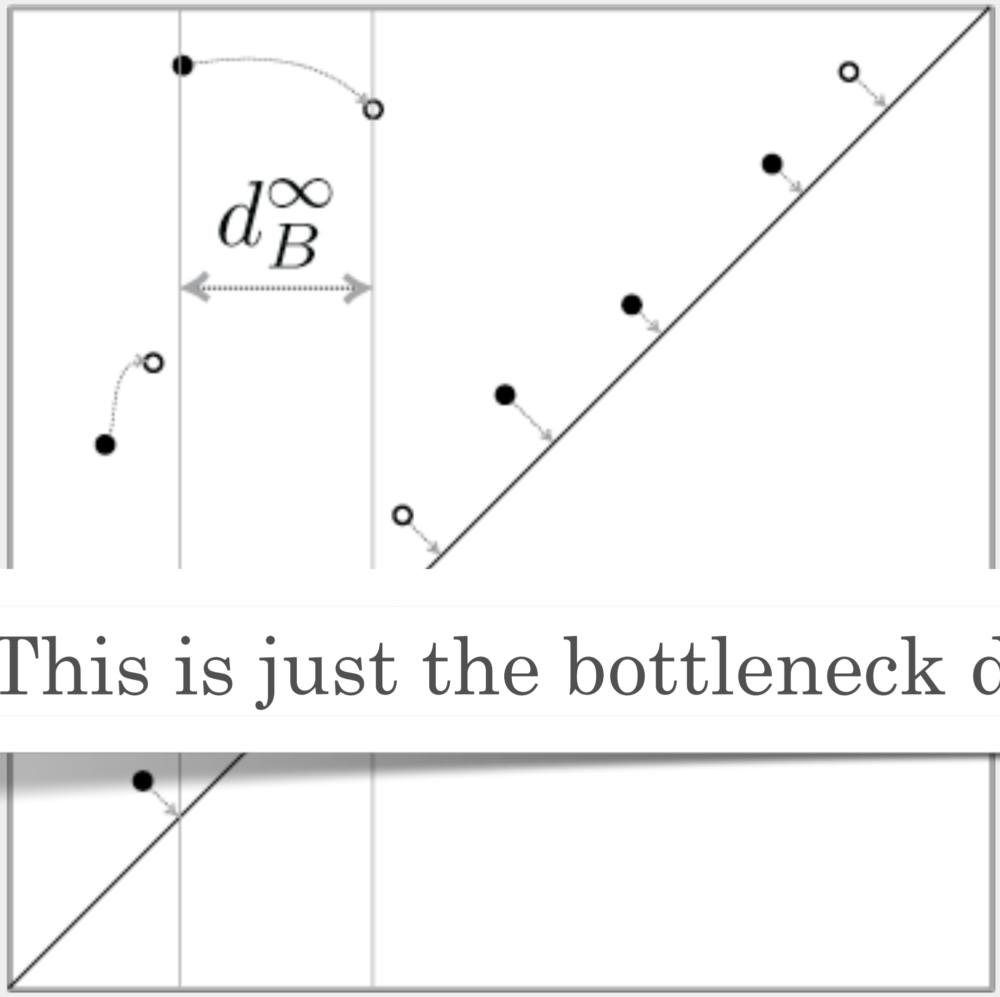


Bottleneck Distance

$$d_B^\infty = \max_i |p_i - q_i|_\infty$$

Approximate Persistence Diagrams

Birth and Death times differ by a constant factor.



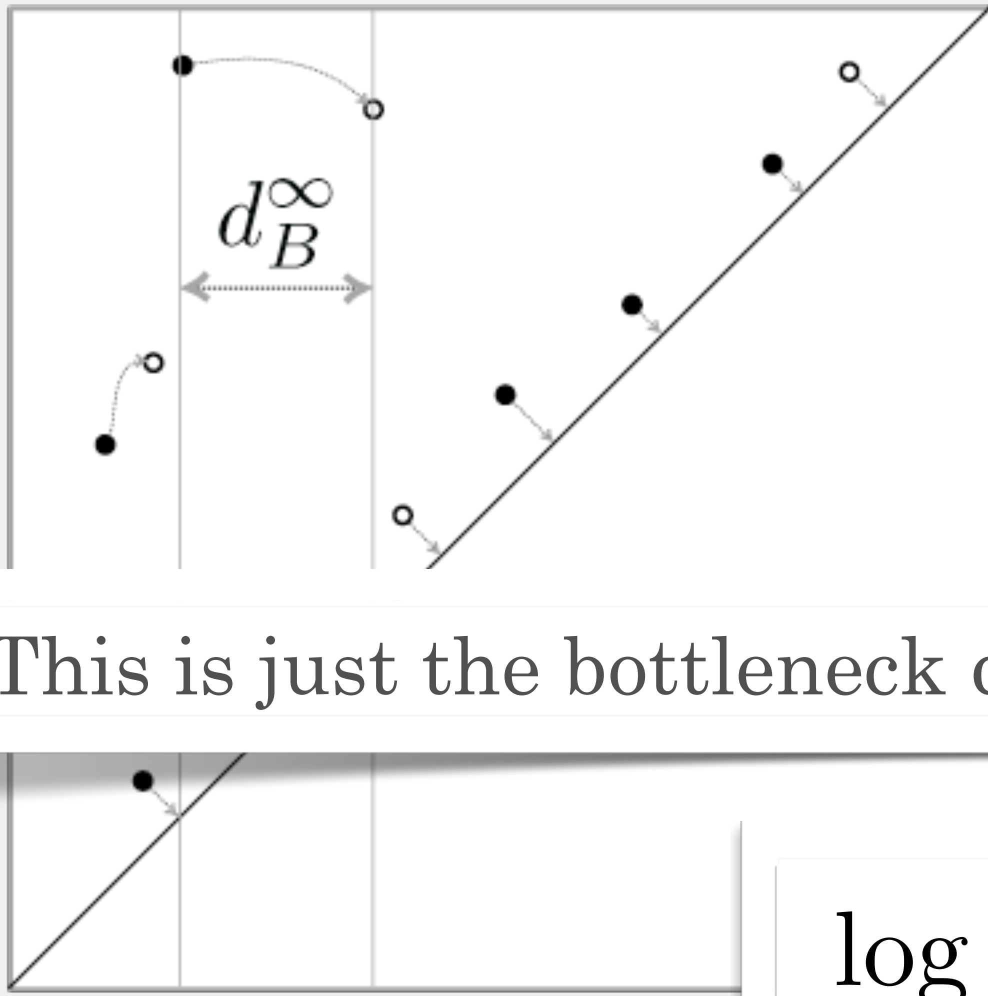
Bottleneck Distance

$$d_B^\infty = \max_i |p_i - q_i|_\infty$$

This is just the bottleneck distance of the log-scale diagrams.

Approximate Persistence Diagrams

Birth and Death times differ by a constant factor.



Bottleneck Distance

$$d_B^\infty = \max_i |p_i - q_i|_\infty$$

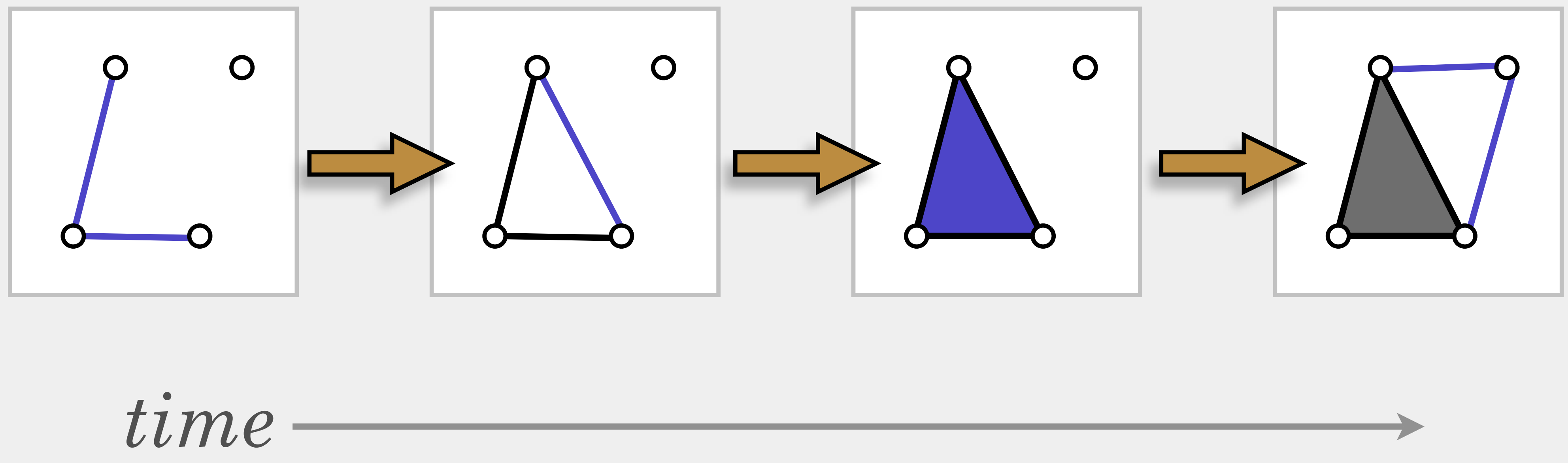
This is just the bottleneck distance of the log-scale diagrams.

$$\begin{aligned} \log a - \log b &< \varepsilon \\ \log \frac{a}{b} &< \varepsilon \\ \frac{a}{b} &< 1 + \varepsilon \end{aligned}$$

We need to build a *filtered simplicial complex*.

Associate a *birth time* with each simplex in complex K.

At time α, we have a complex K_α consisting of all simplices born at or before time α.



There are two phases, one is geometric the other is topological.

Geometry

Build a filtration, i.e. a filtered simplicial complex.

Topology
(linear algebra)

Compute the persistence diagram
(Run the Persistence Algorithm).

There are two phases, one is geometric the other is topological.

Geometry

Build a filtration, i.e. a filtered simplicial complex.

Topology
(linear algebra)

Compute the persistence diagram (Run the Persistence Algorithm).

We'll focus on this side.

There are two phases, one is geometric the other is topological.

Geometry

Build a filtration, i.e. a filtered simplicial complex.

Topology
(linear algebra)

Compute the persistence diagram (Run the Persistence Algorithm).

We'll focus on this side.

Running time: **$O(N^3)$**.
N is the **size** of the complex.

Idea 1: Use the Delaunay Triangulation

Idea 1: Use the Delaunay Triangulation

Good: It works, (alpha-complex filtration).

Idea 1: Use the Delaunay Triangulation

Good: It works, (alpha-complex filtration).

Bad: It can have size $\mathbf{n}^{\mathbf{O(d)}}$.

Idea 2: Connect up everything close.

Idea 2: Connect up everything close.

Čech Filtration: Add a k-simplex for every $k+1$ points that have a smallest enclosing ball of radius at most α.

Idea 2: Connect up everything close.

Čech Filtration: Add a k-simplex for every $k+1$ points that have a smallest enclosing ball of radius at most α.

Rips Filtration: Add a k-simplex for every $k+1$ points that have all pairwise distances at most α.

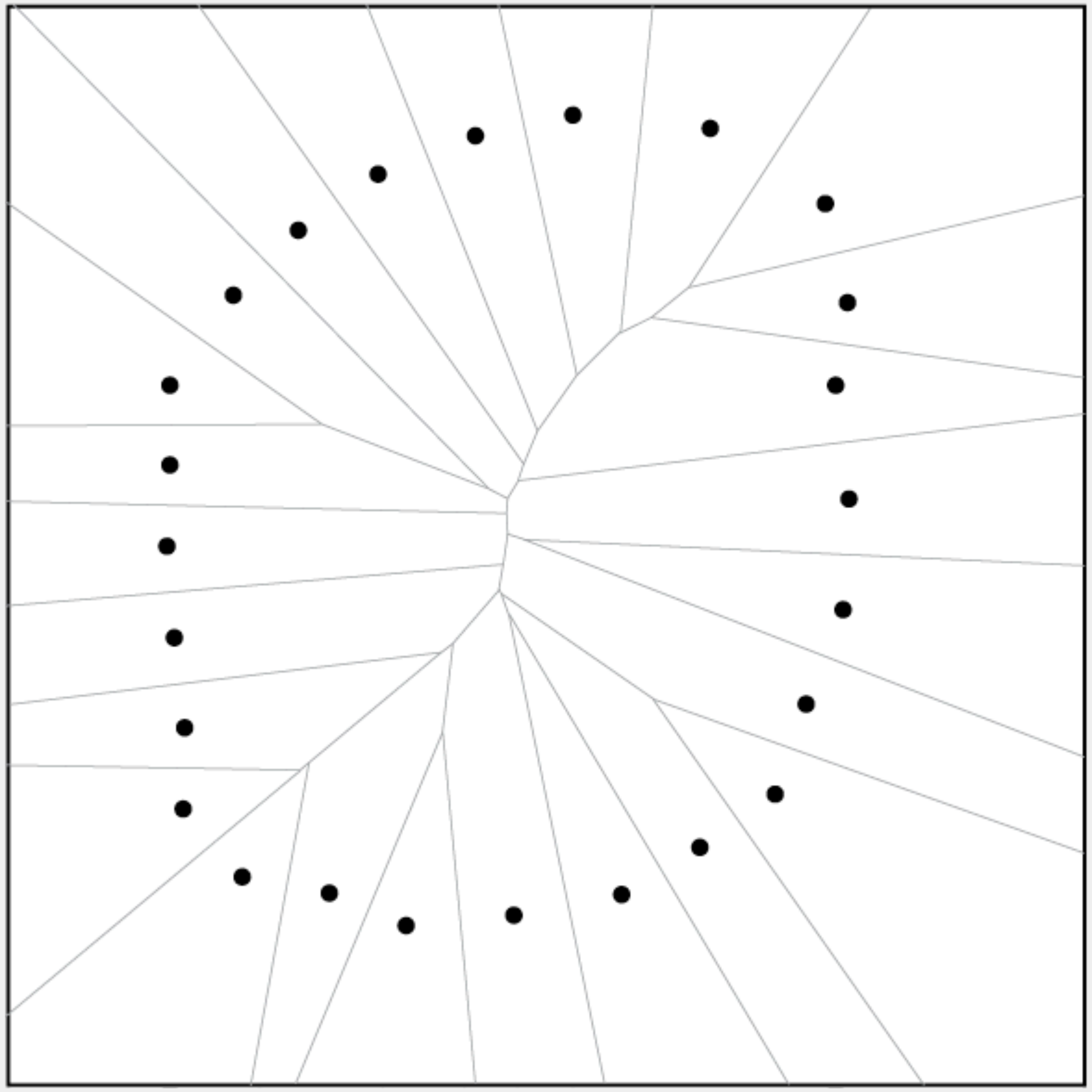
Idea 2: Connect up everything close.

Čech Filtration: Add a k-simplex for every $k+1$ points that have a smallest enclosing ball of radius at most α.

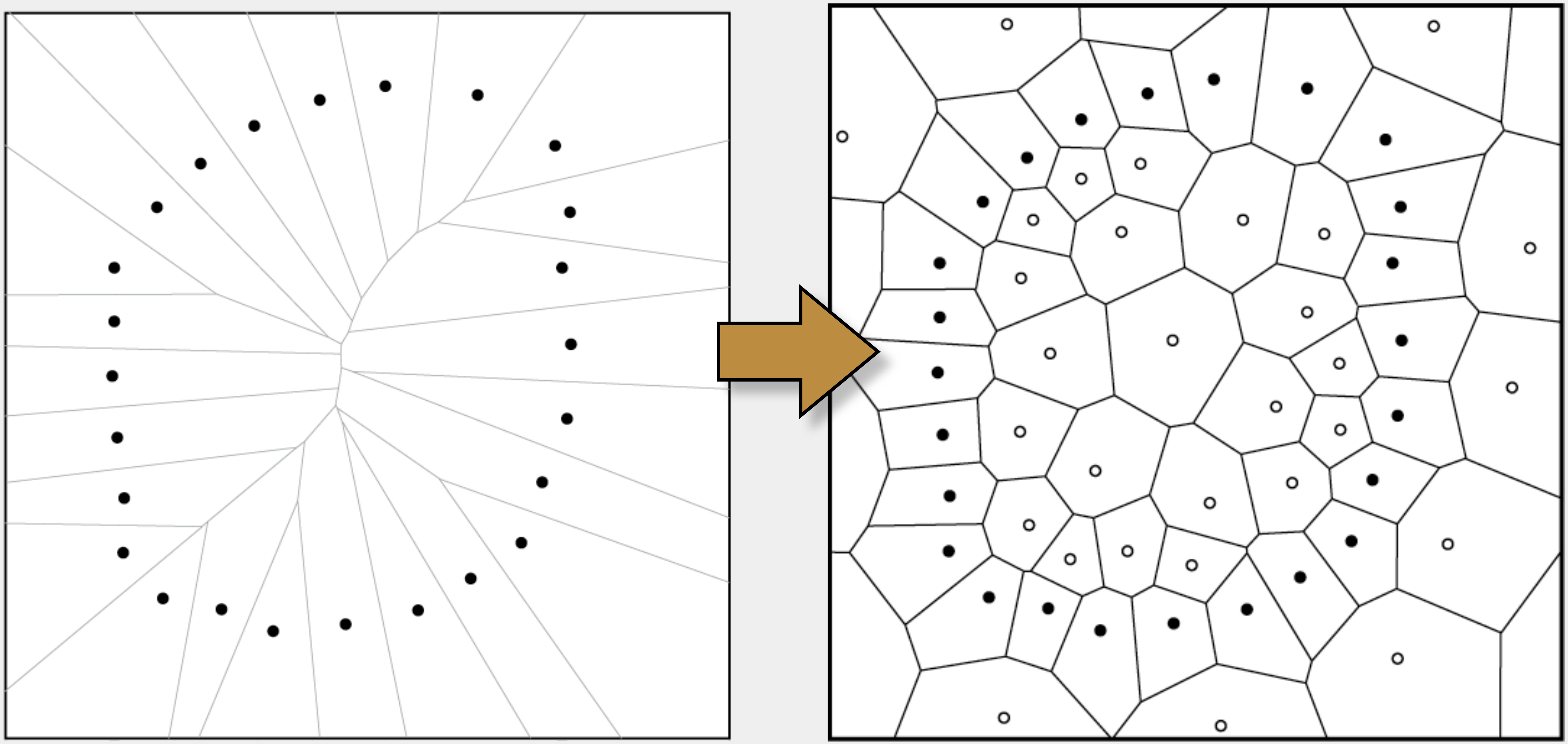
Rips Filtration: Add a k-simplex for every $k+1$ points that have all pairwise distances at most α.

Still $\mathbf{n^d}$, but we can quit early.

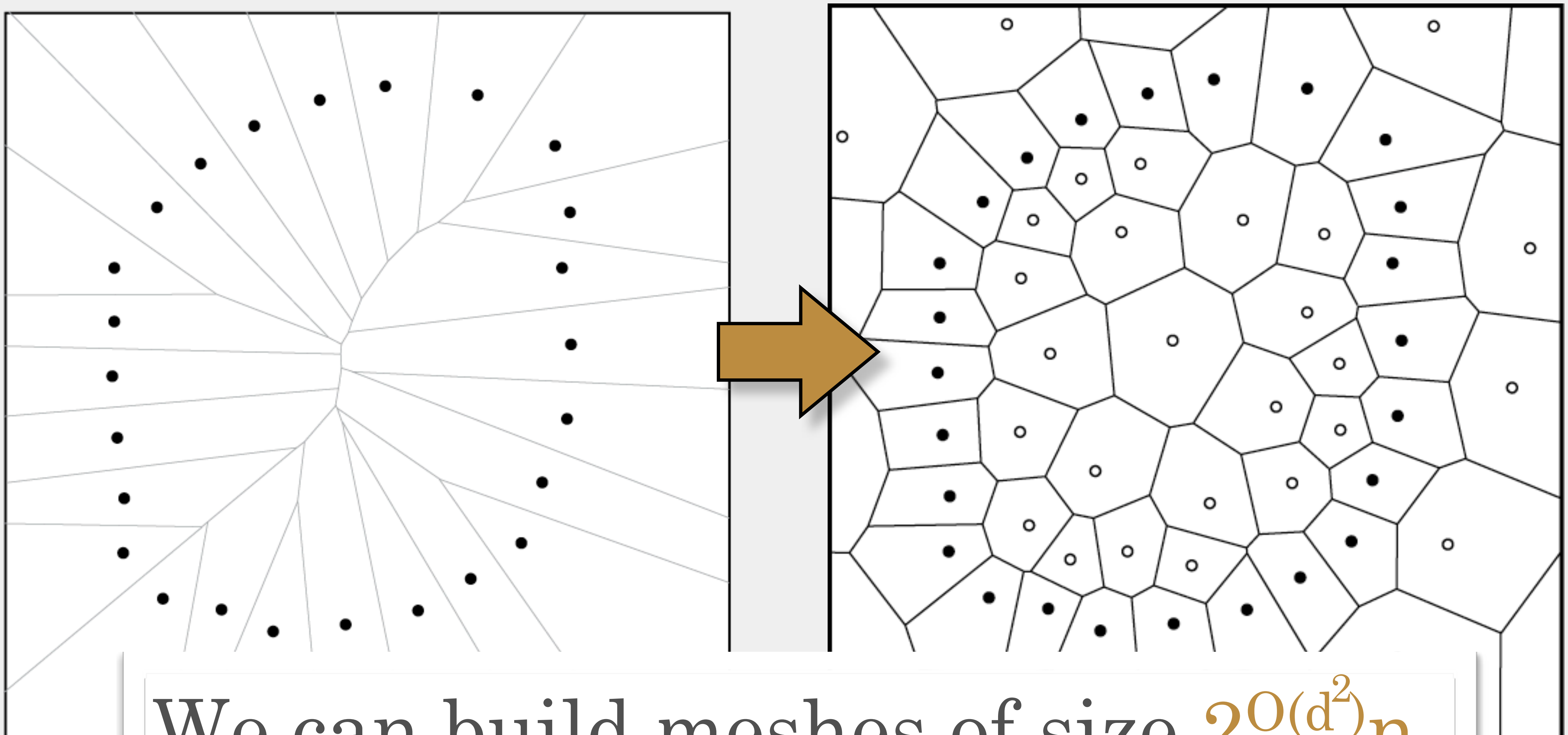
Our Idea: Build a **quality mesh**.



Our Idea: Build a **quality mesh**.



Our Idea: Build a **quality mesh**.



We can build meshes of size $2^{O(d^2)}n$.

Meshing Counter-intuition

Delaunay Refinement
can take ***less time and space*** than
Delaunay Triangulation.

Meshing Counter-intuition

Delaunay Refinement
can take ***less time and space*** than
Delaunay Triangulation.

Theorem [Hudson, Miller, Phillips, '06]:

A quality mesh of a point set can be constructed in $O(n \log \Delta)$ time, where Δ is the spread.

Meshing Counter-intuition

Delaunay Refinement
can take ***less time and space*** than
Delaunay Triangulation.

Theorem [Hudson, Miller, Phillips, '06]:

A quality mesh of a point set can be constructed in $O(n \log \Delta)$ time, where Δ is the spread.

Theorem [Miller, Phillips, Sheehy, '08]:

A quality mesh of a *well-paced* point set has size $O(n)$.

The α-mesh filtration

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

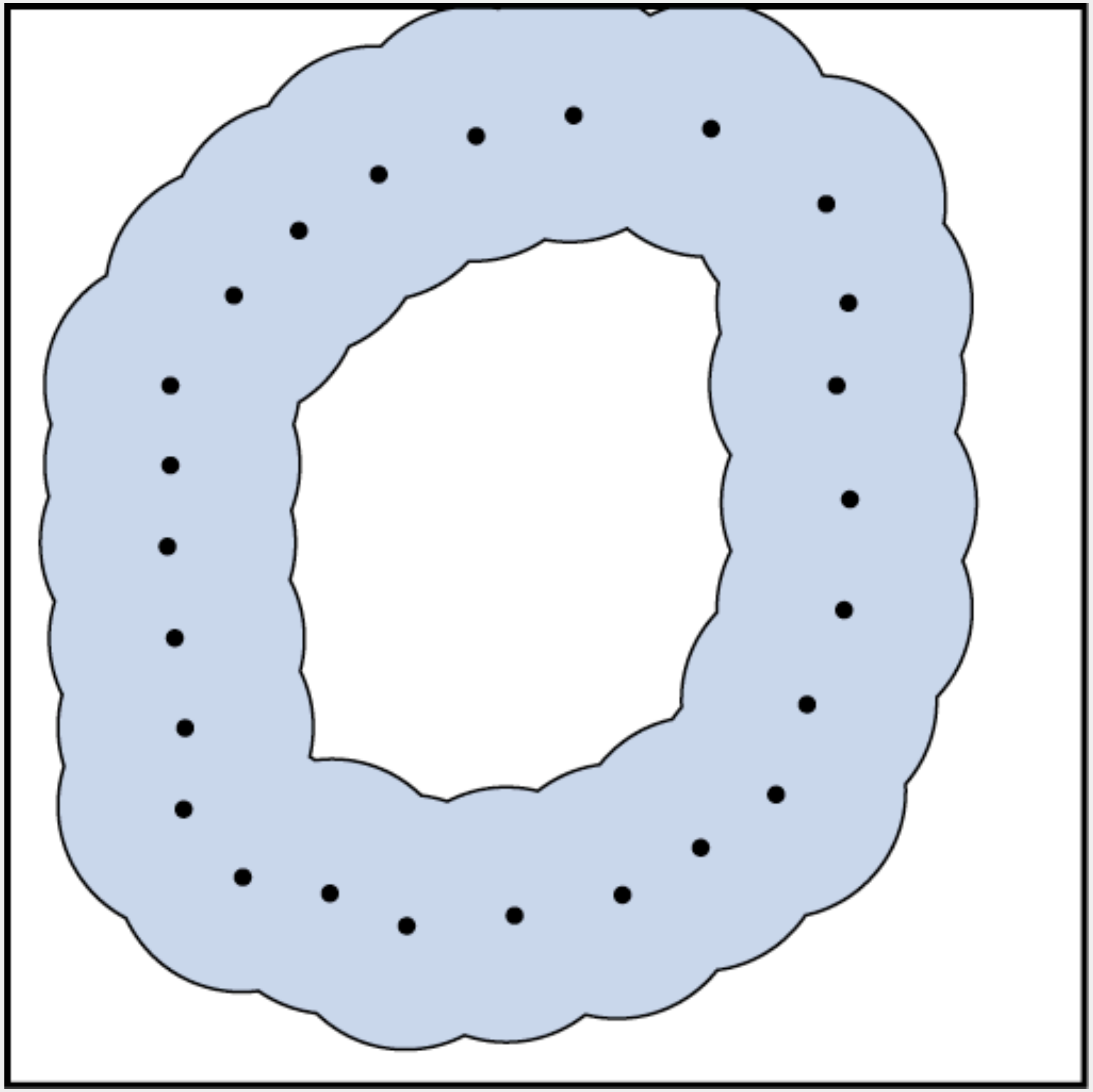
The α-mesh filtration

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



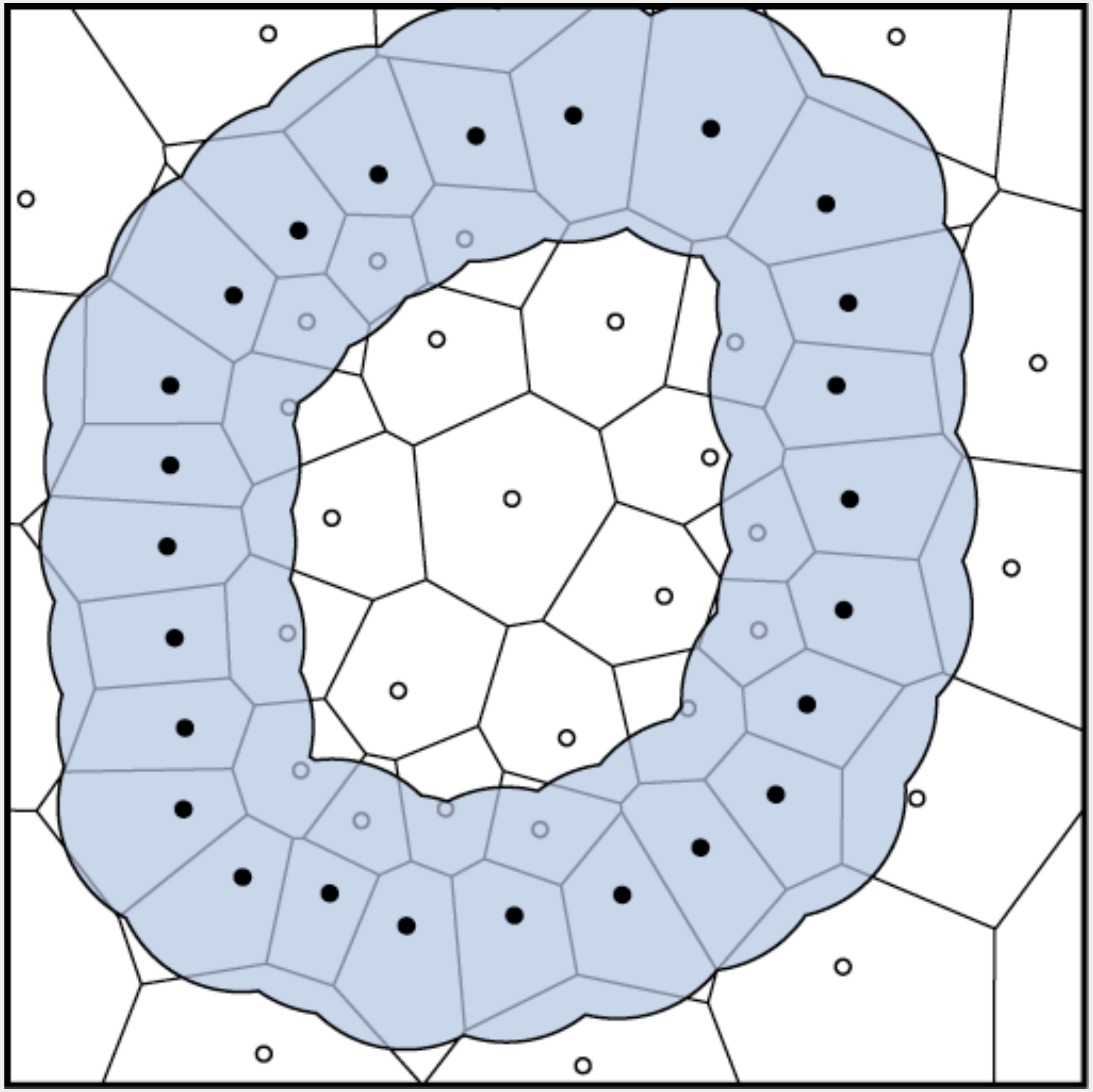
The α-mesh filtration

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



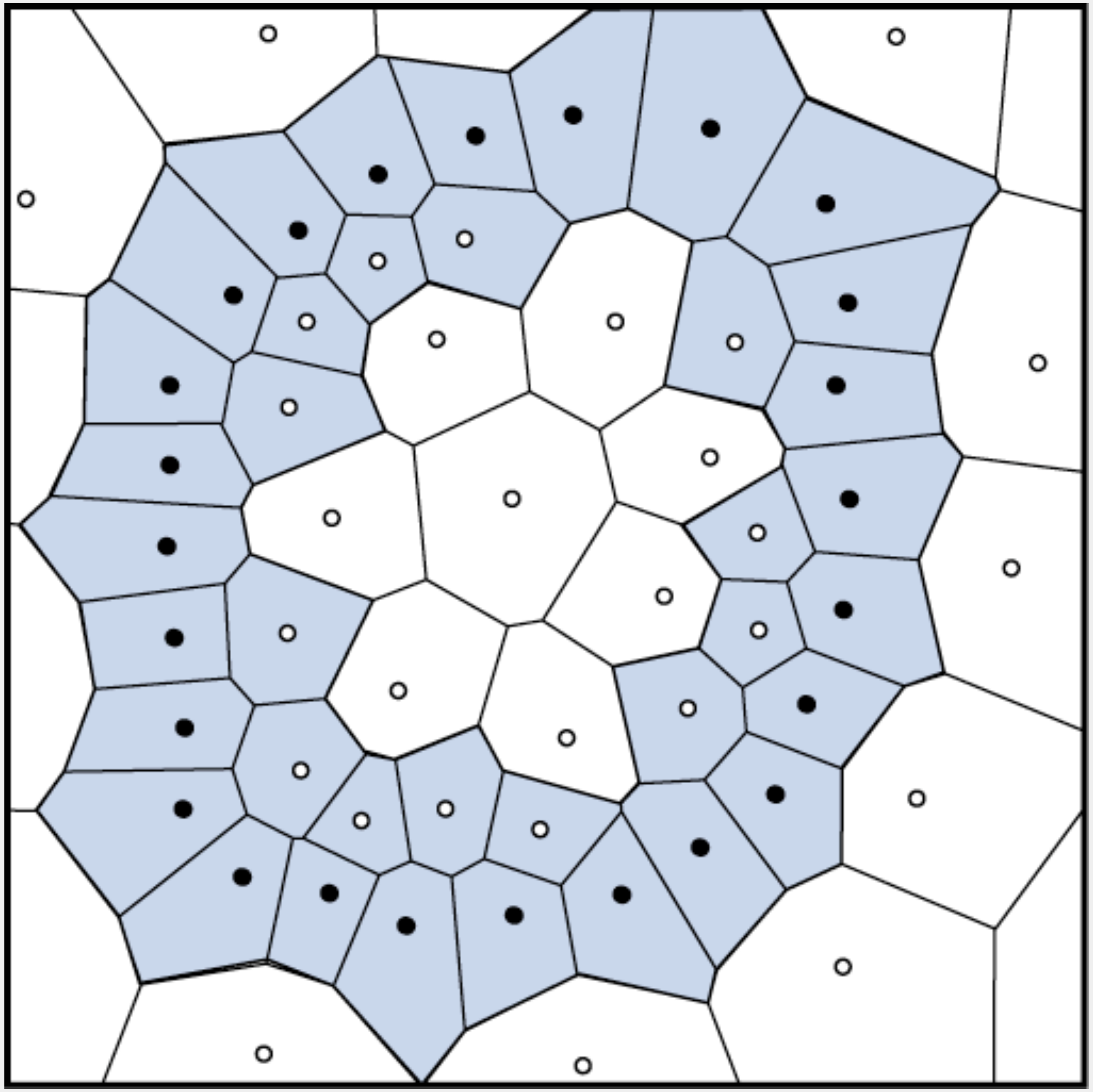
The α-mesh filtration

1. Build a mesh *M*.

2. Assign birth times to vertices based on distance to *P* (special case points very close to *P*).

3. For each simplex *s* of Del(*M*), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



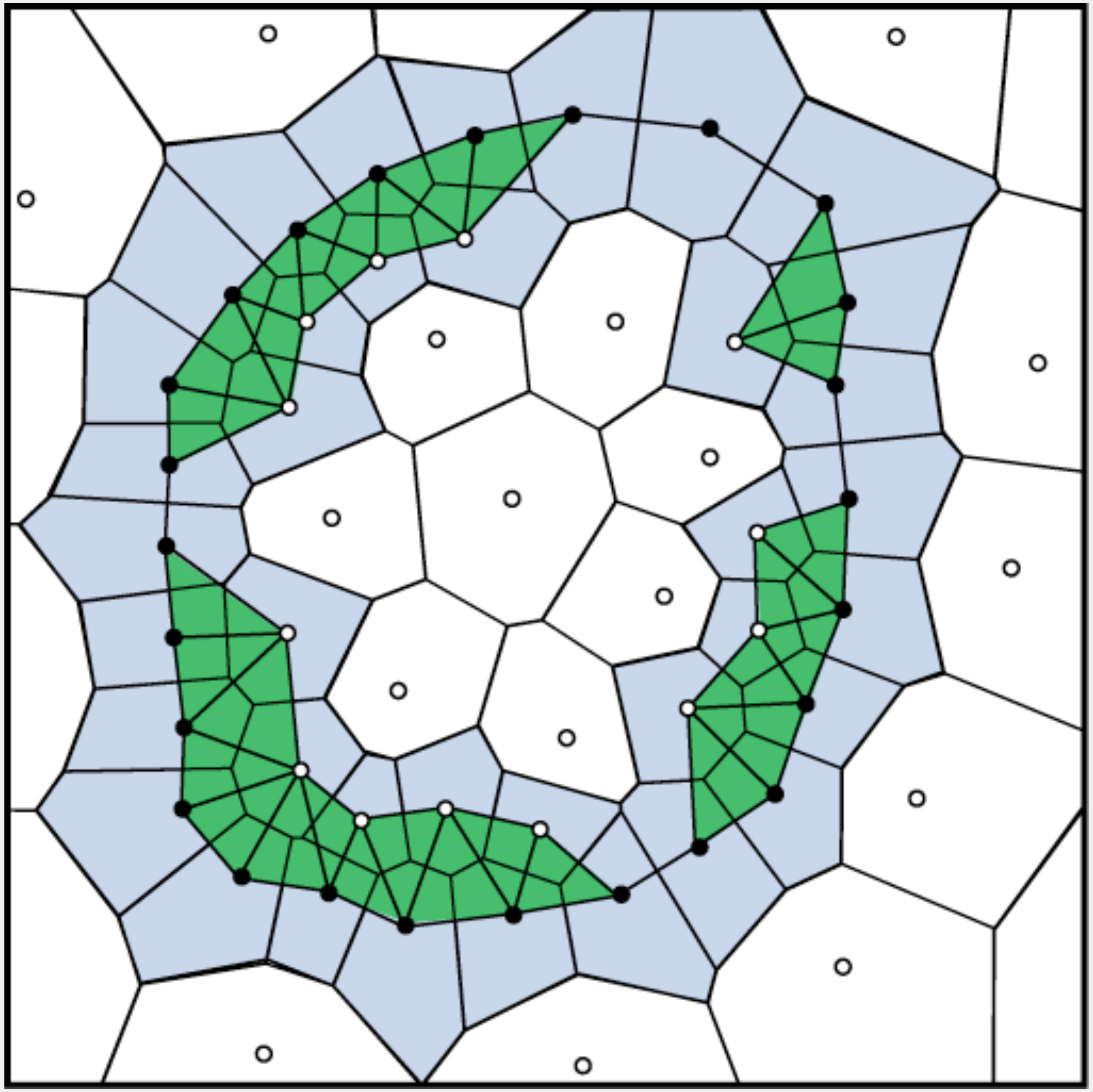
The α-mesh filtration

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



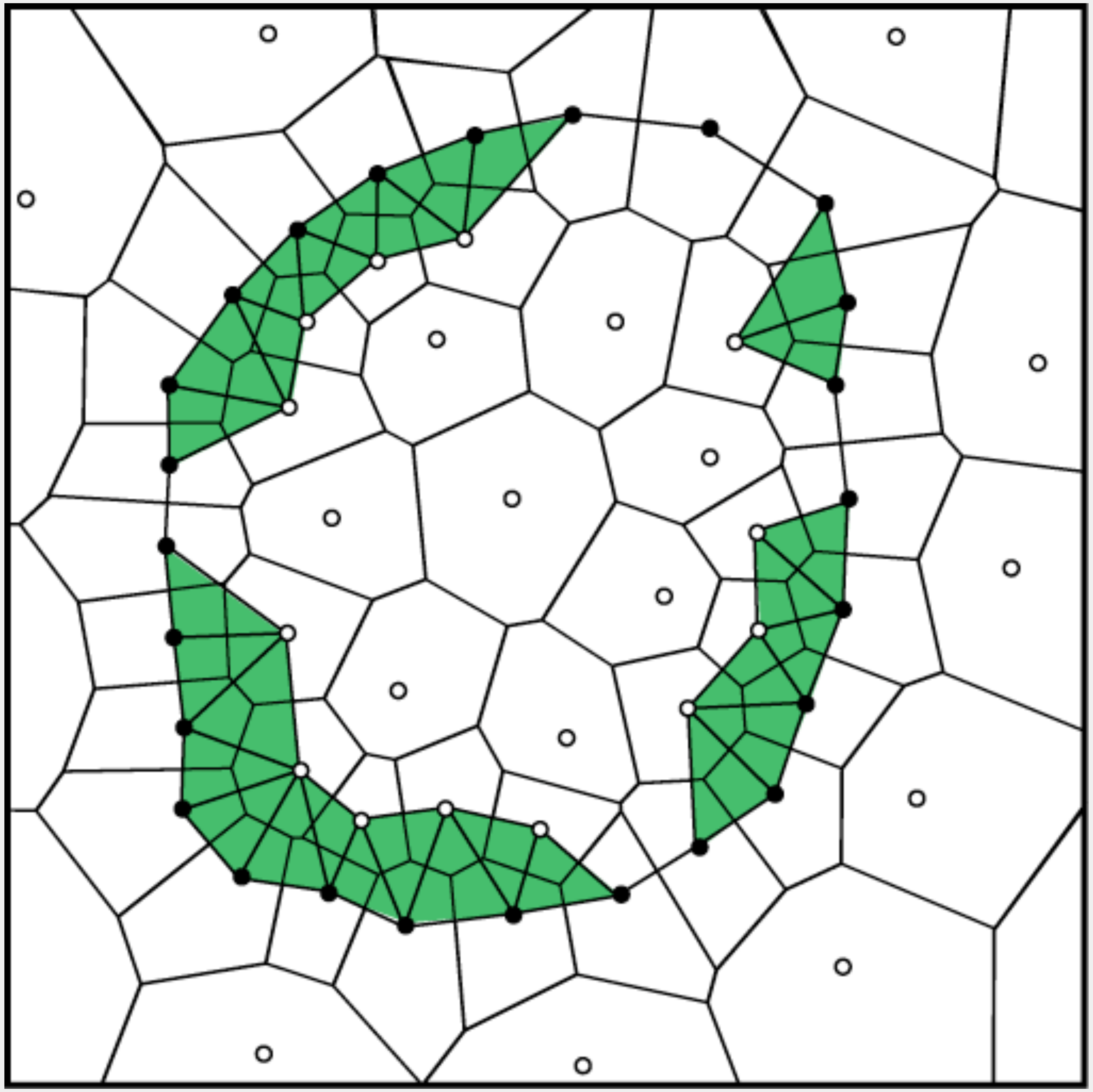
The α-mesh filtration

1. Build a mesh *M*.

2. Assign birth times to vertices based on distance to *P* (special case points very close to *P*).

3. For each simplex *s* of Del(*M*), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



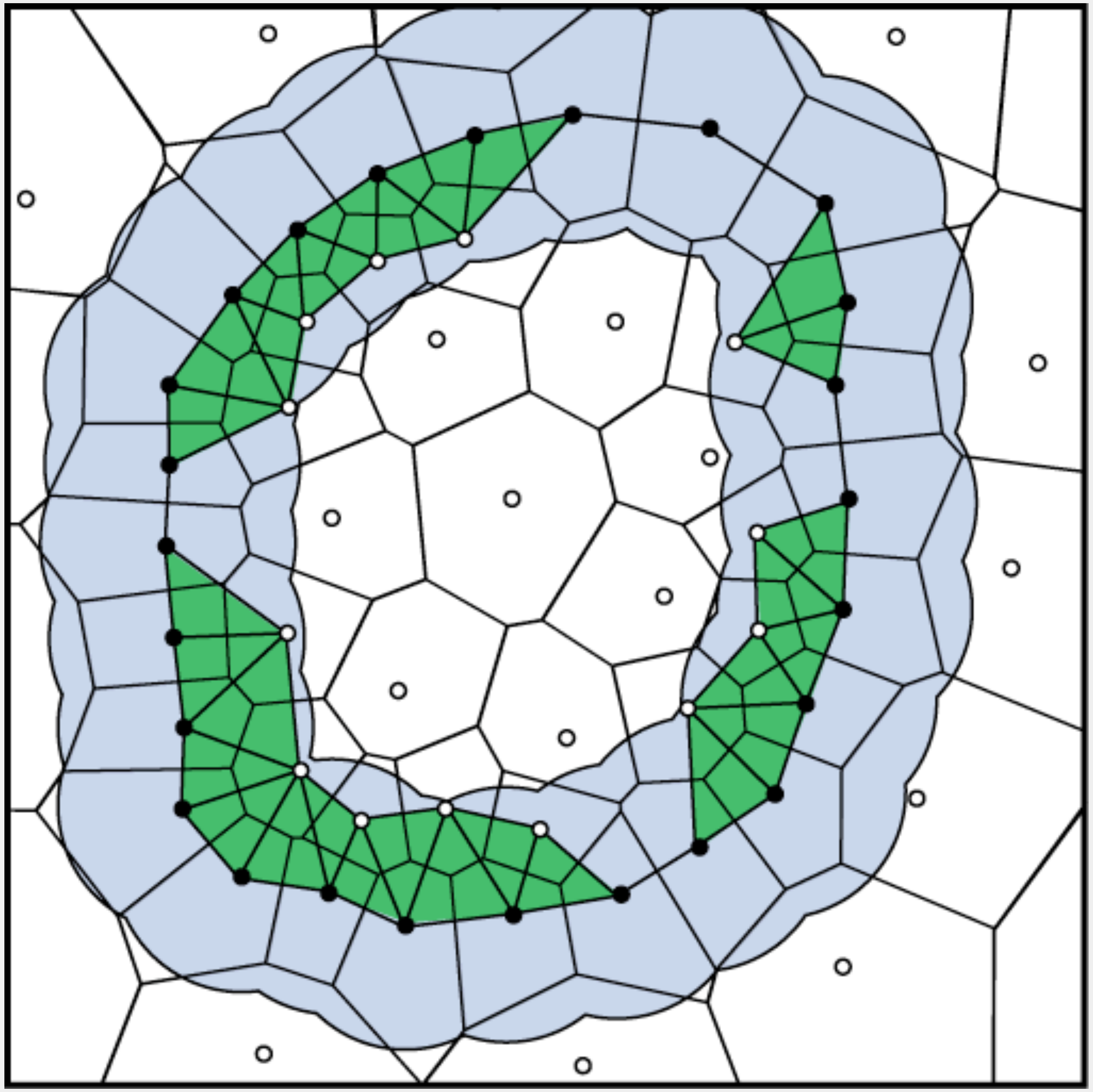
The α-mesh filtration

1. Build a mesh *M*.

2. Assign birth times to vertices based on distance to *P* (special case points very close to *P*).

3. For each simplex *s* of Del(*M*), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.



Approximation via interleaving.

Approximation via interleaving.

Definition:

Two filtrations, $\{P_\alpha\}$ and $\{Q_\alpha\}$ are ε-*interleaved* if $P_{\alpha-\varepsilon} \subseteq Q_\alpha \subseteq P_{\alpha+\varepsilon}$ for all α.

Approximation via interleaving.

Definition:

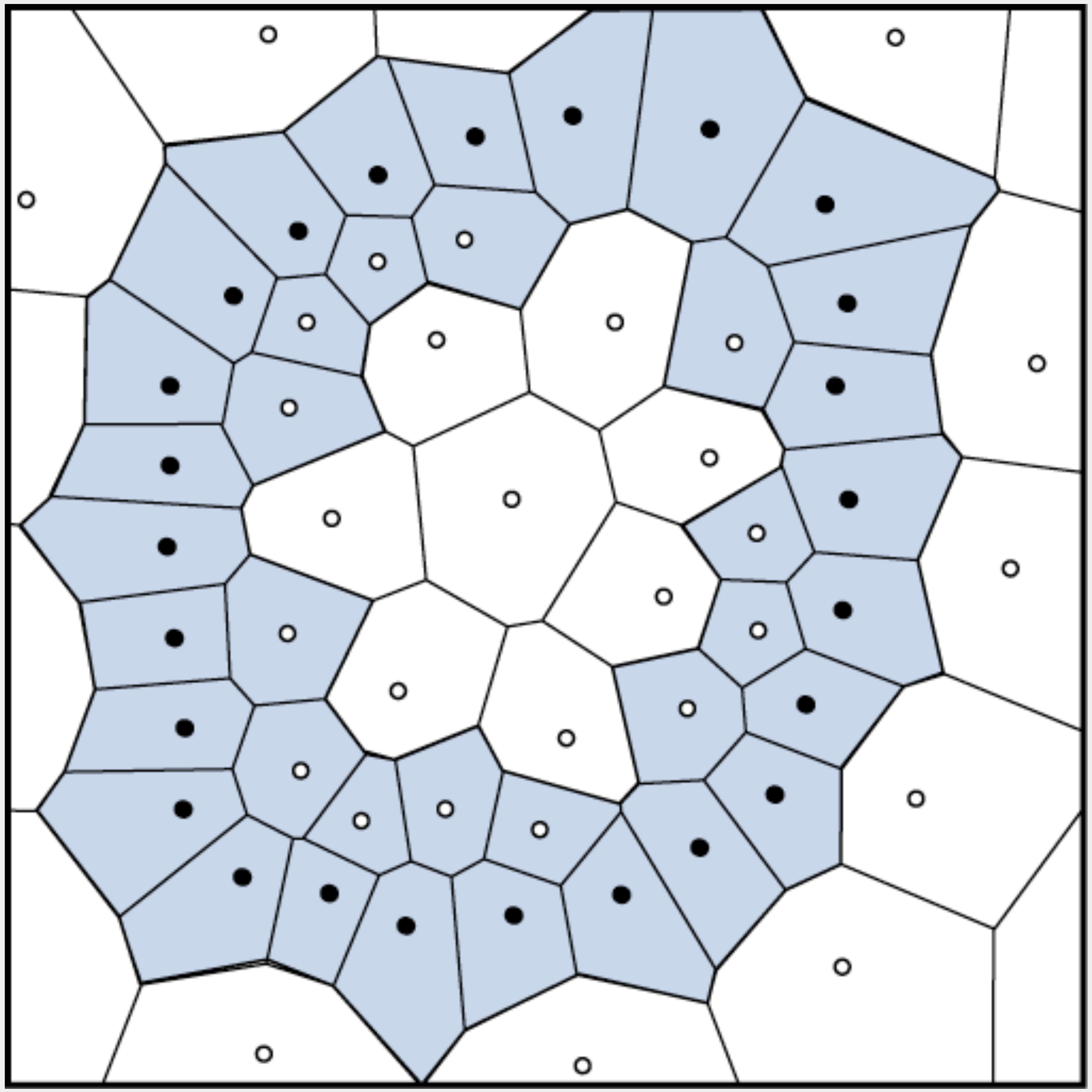
Two filtrations, $\{P_\alpha\}$ and $\{Q_\alpha\}$ are ε-*interleaved* if $P_{\alpha-\varepsilon} \subseteq Q_\alpha \subseteq P_{\alpha+\varepsilon}$ for all α.

Theorem [Chazal et al, '09]:

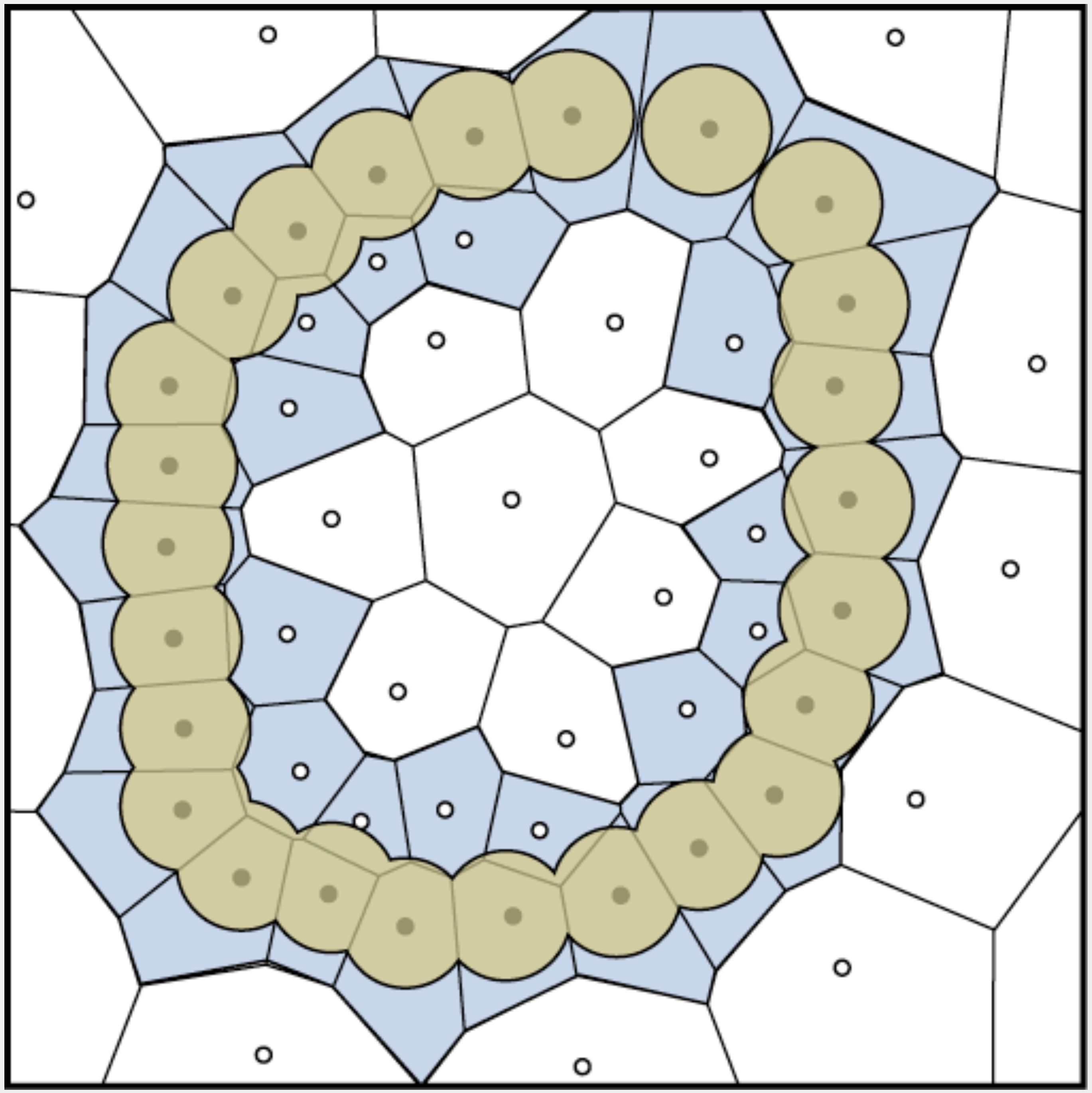
If $\{P_\alpha\}$ and $\{Q_\alpha\}$ are ε-interleaved then their persistence diagrams are ε-close in the bottleneck distance.

The Voronoi filtration interleaves with the offset filtration.

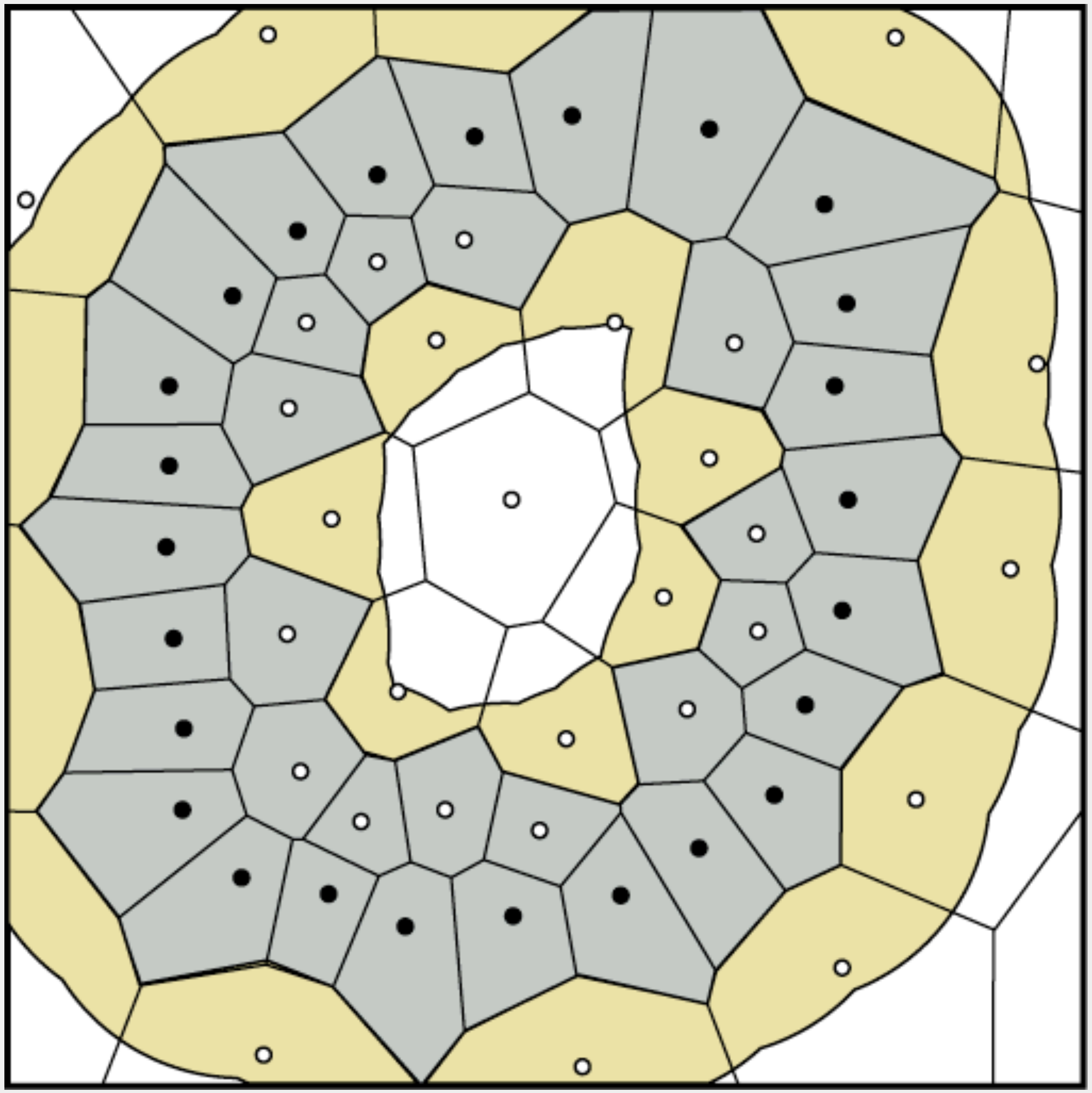
The Voronoi filtration interleaves with the offset filtration.



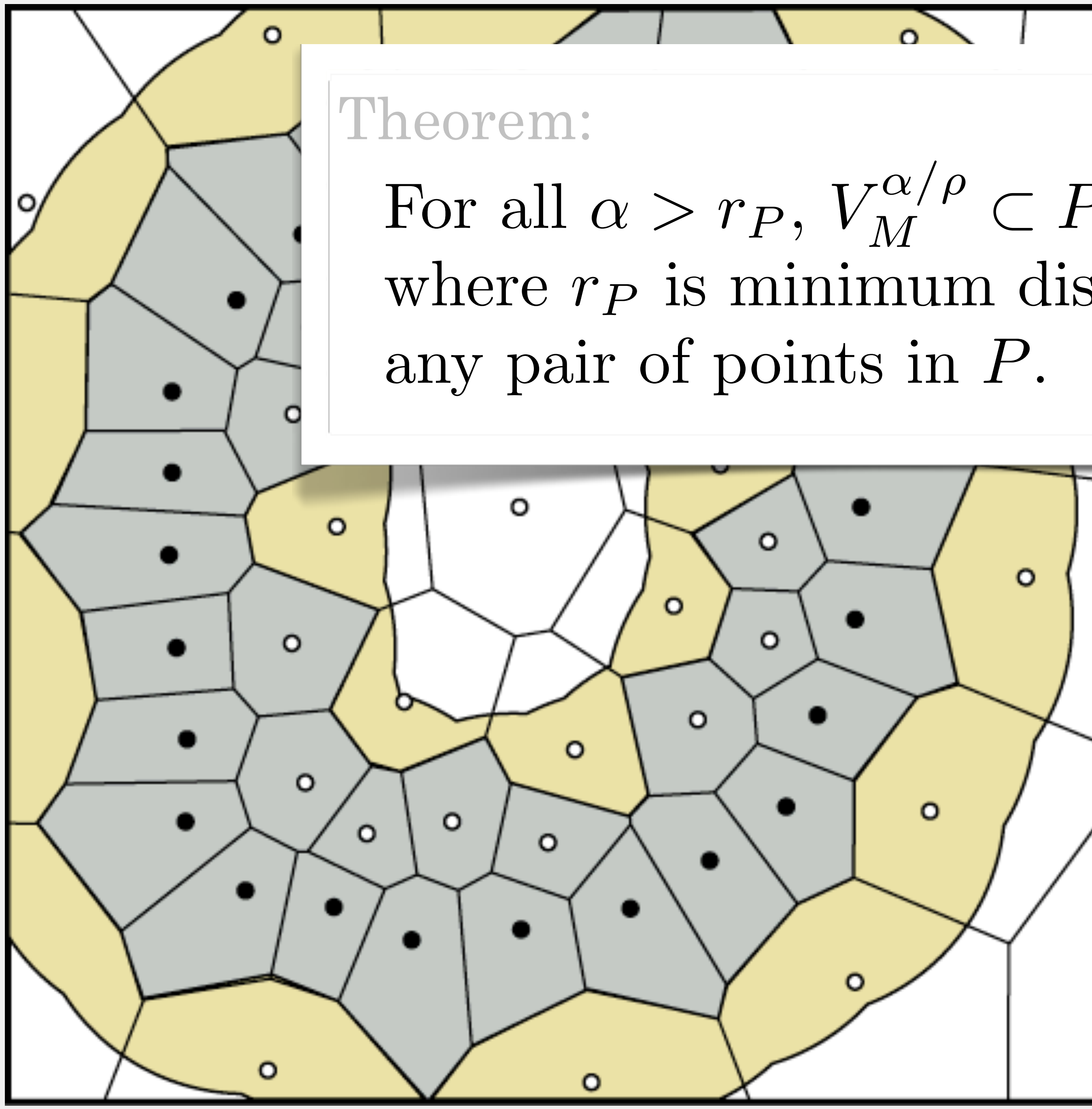
The Voronoi filtration interleaves with the offset filtration.



The Voronoi filtration interleaves with the offset filtration.



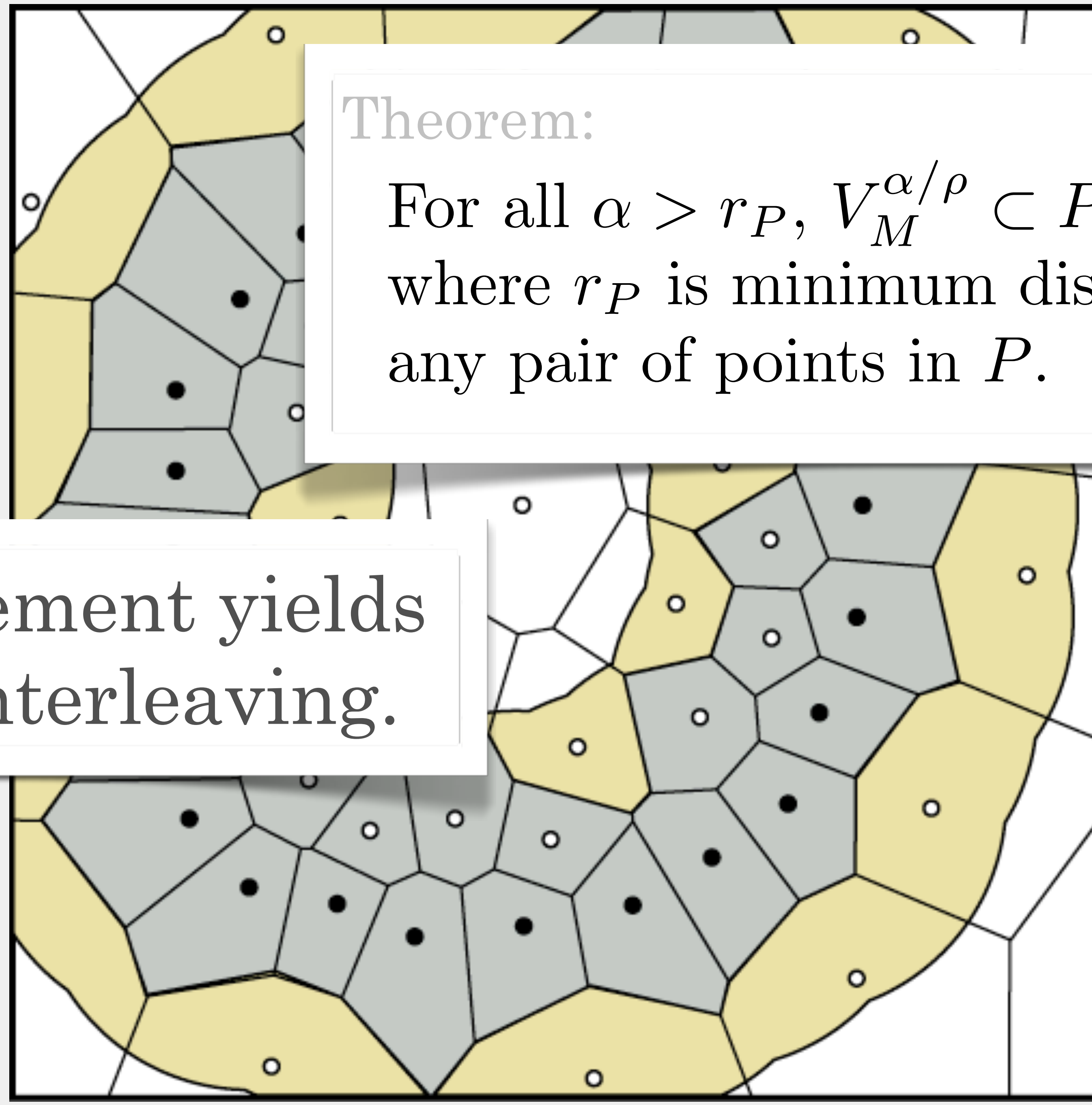
The Voronoi filtration interleaves with the offset filtration.



Theorem:

For all $\alpha > r_P$, $V_M^{\alpha/\rho} \subset P^\alpha \subset V_M^{\alpha\rho}$, where r_P is minimum distance between any pair of points in P.

The Voronoi filtration interleaves with the offset filtration.

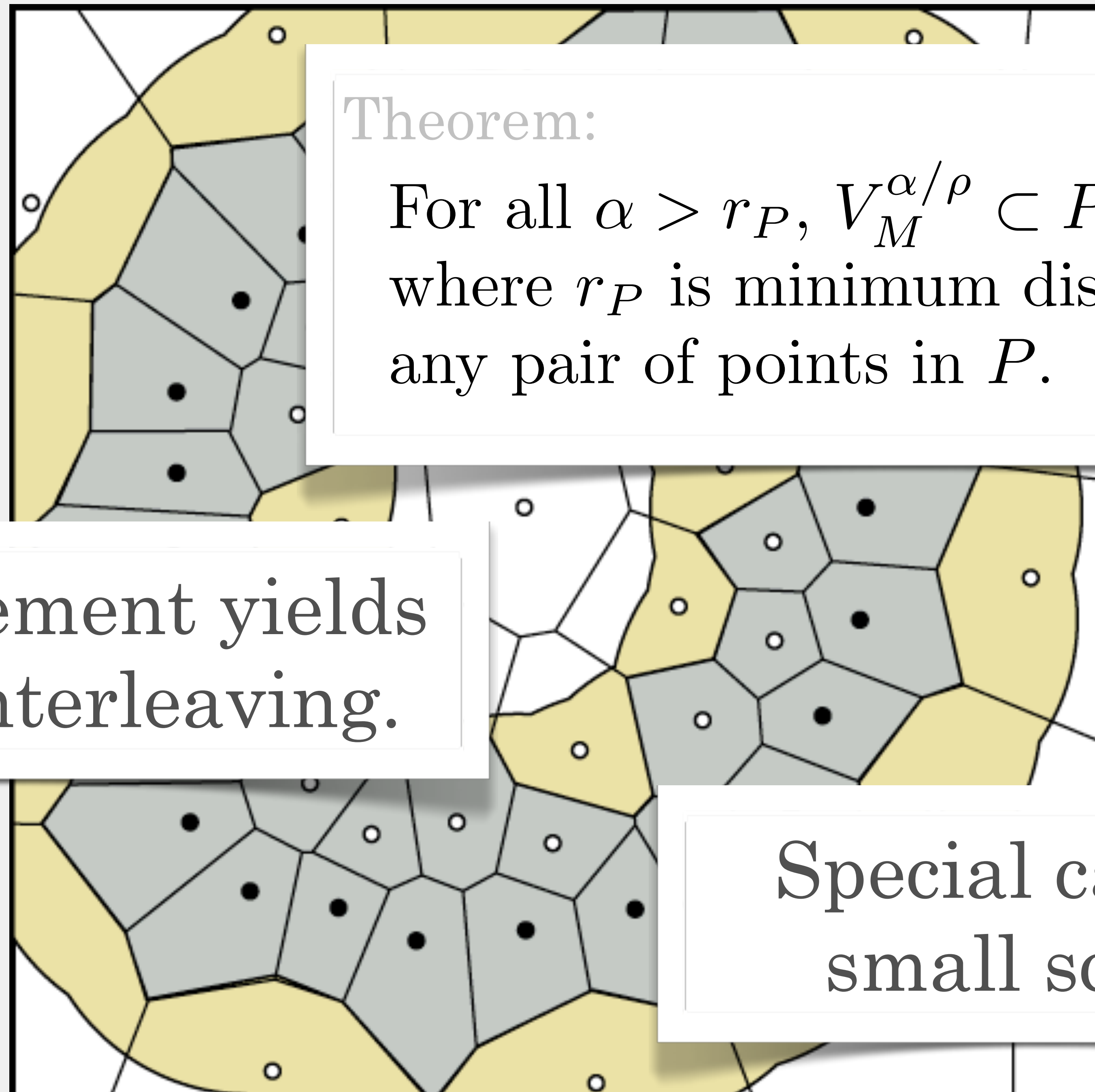


Theorem:

For all $\alpha > r_P$, $V_M^{\alpha/\rho} \subset P^\alpha \subset V_M^{\alpha\rho}$, where r_P is minimum distance between any pair of points in P.

Finer refinement yields a tighter interleaving.

The Voronoi filtration interleaves with the offset filtration.



Theorem:

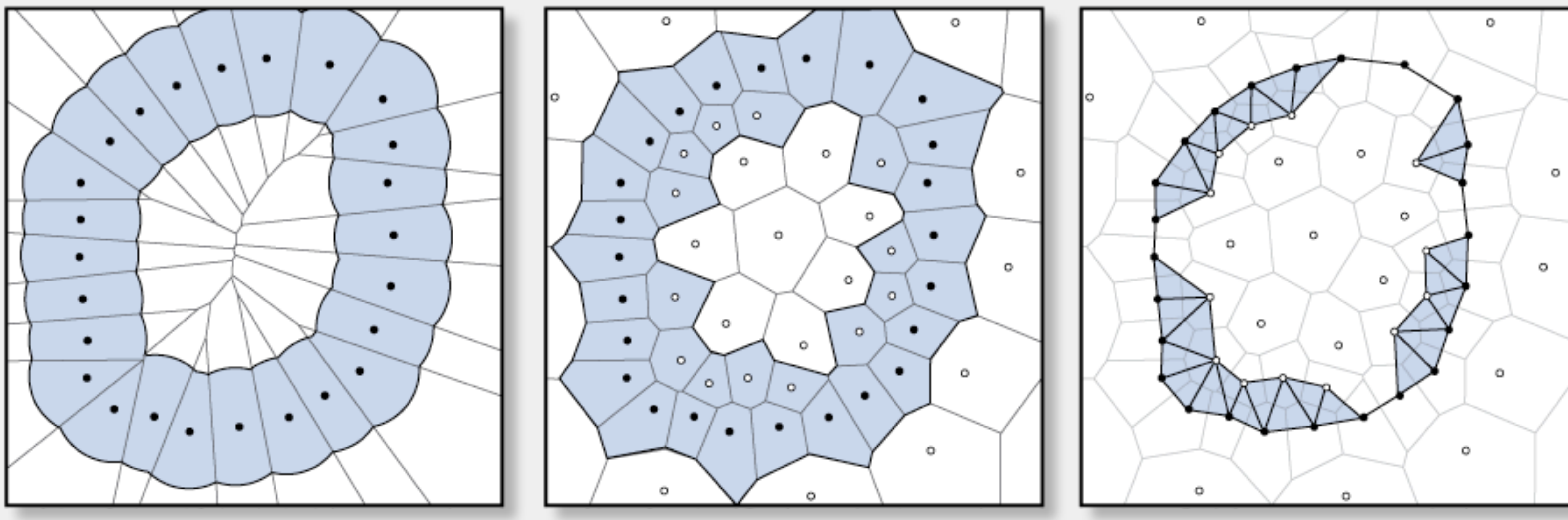
For all $\alpha > r_P$, $V_M^{\alpha/\rho} \subset P^\alpha \subset V_M^{\alpha\rho}$, where r_P is minimum distance between any pair of points in P.

Finer refinement yields a tighter interleaving.

Special case for small scales.

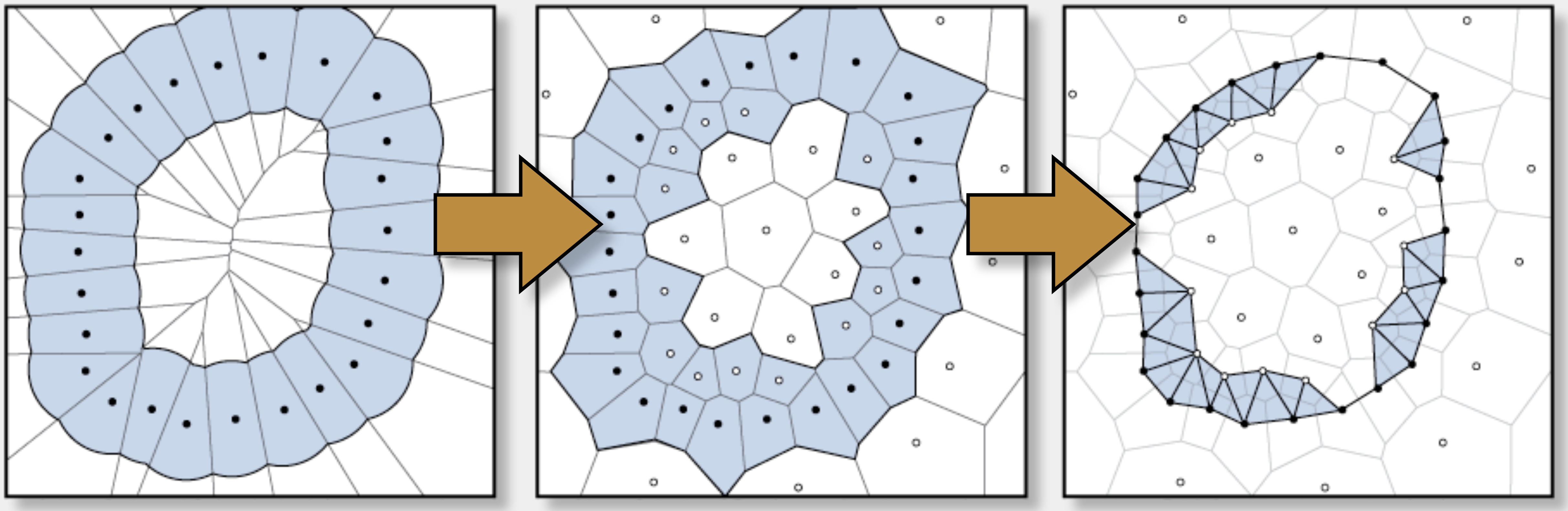
Geometric Approximation

Topologically Equivalent



Geometric Approximation

Topologically Equivalent



The Results

1. Build a mesh *M*.

2. Assign birth times to vertices based on distance to *P* (special case points very close to *P*).**

3. For each simplex *s* of Del(*M*), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

	Approximation ratio	Complex Size
Previous Work	1	$n^{O(d)}$
Simple mesh filtration		
Over-refine the mesh		
Linear-Size Meshing		

The Results

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).**

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

	Approximation ratio	Complex Size
Previous Work	1	$n^{O(d)}$
Simple mesh filtration	ρ	$2^{O(d^2)} n \log \Delta$
Over-refine the mesh		
Linear-Size Meshing		

The Results

1. Build a mesh M.

2. Assign birth times to vertices based on distance to P (special case points very close to P).**

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

	Approximation ratio	Complex Size
Previous Work	1	$n^{O(d)}$
Simple mesh filtration	$\rho = \sim 3$	$2^{O(d^2)} n \log \Delta$
Over-refine the mesh		
Linear-Size Meshing		

The Results

1. Build a mesh M.
   Over-refine it.

2. Assign birth times to vertices based on distance to P (special case points very close to P).**

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

	Approximation ratio	Complex Size
Previous Work	1	$n^{O(d)}$
Simple mesh filtration	$\rho =\sim 3$	$2^{O(d^2)} n \log \Delta$
Over-refine the mesh	$1 + \varepsilon$	$\varepsilon^{-O(d^2)} n \log \Delta$
Linear-Size Meshing		

The Results

1. Build a mesh M.
    Over-refine it.
    Use linear-size meshing.

2. Assign birth times to vertices based on distance to P (special case points very close to P).**

3. For each simplex s of Del(M), let birth(s) be the min birth time of its vertices.

4. Feed this filtered complex to the persistence algorithm.

	Approximation ratio	Complex Size
Previous Work	1	$n^{O(d)}$
Simple mesh filtration	$\rho =\sim 3$	$2^{O(d^2)} n \log \Delta$
Over-refine the mesh	$1 + \varepsilon$	$\varepsilon^{-O(d^2)} n \log \Delta$
Linear-Size Meshing	$1 + \varepsilon + 3\theta$	$(\varepsilon\theta)^{-O(d^2)} n$

Thank you.