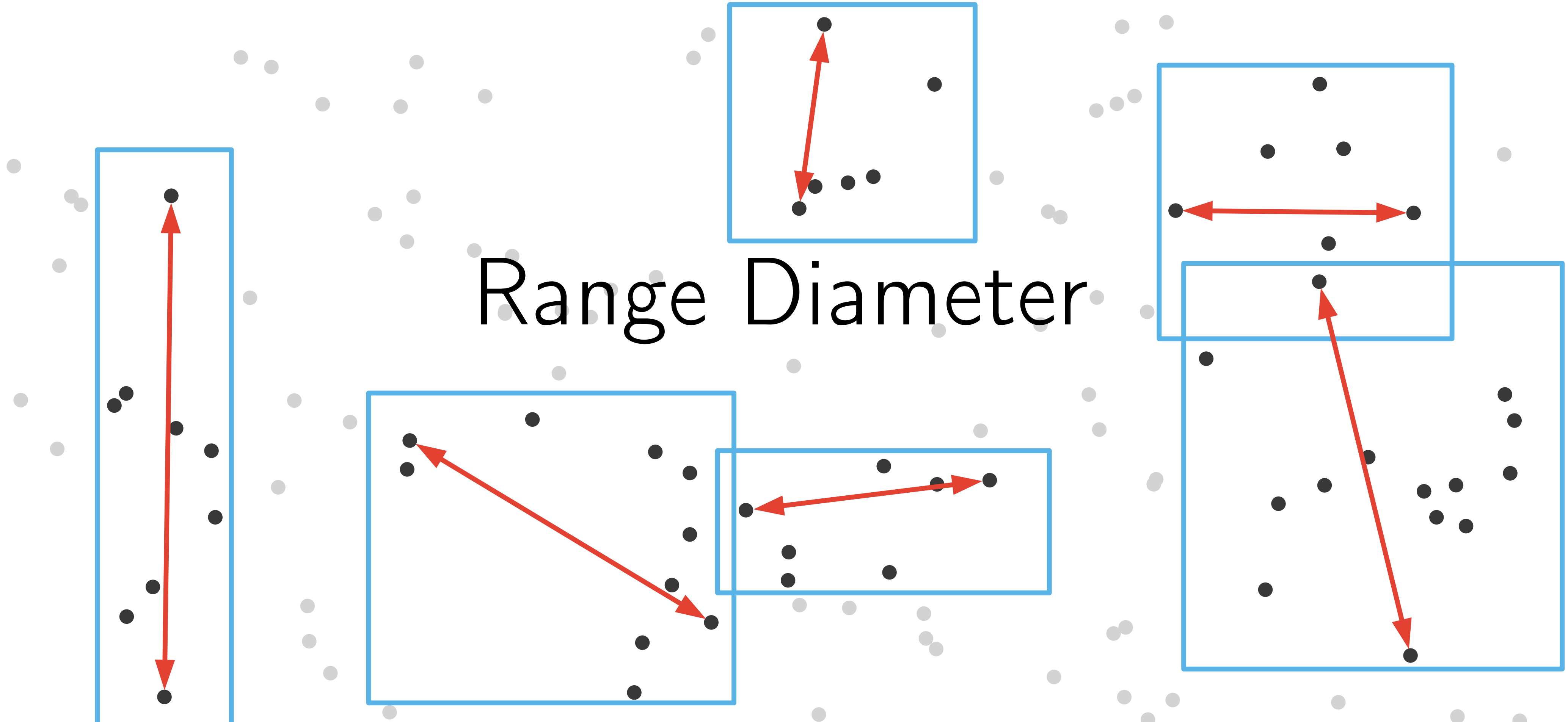
Range Diameter



Pooya Davoodi

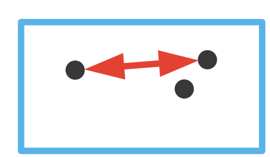
Aarhus University
Denmark
(Now at polytechnic
institute of NYU.)

Michiel Smid

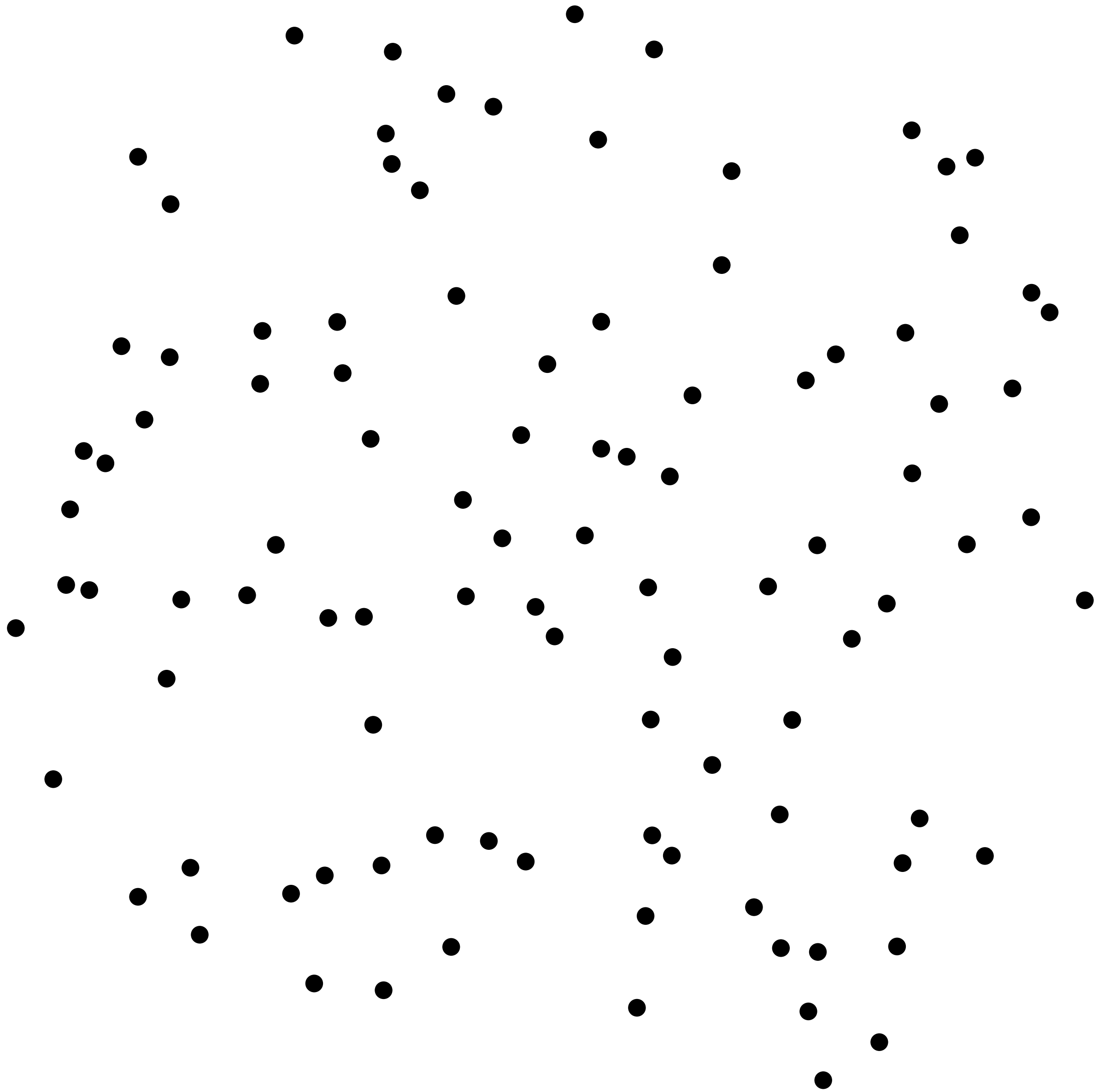
Carleton University
Ottawa, Canada

Freek van Walderveen

Aarhus University
Denmark

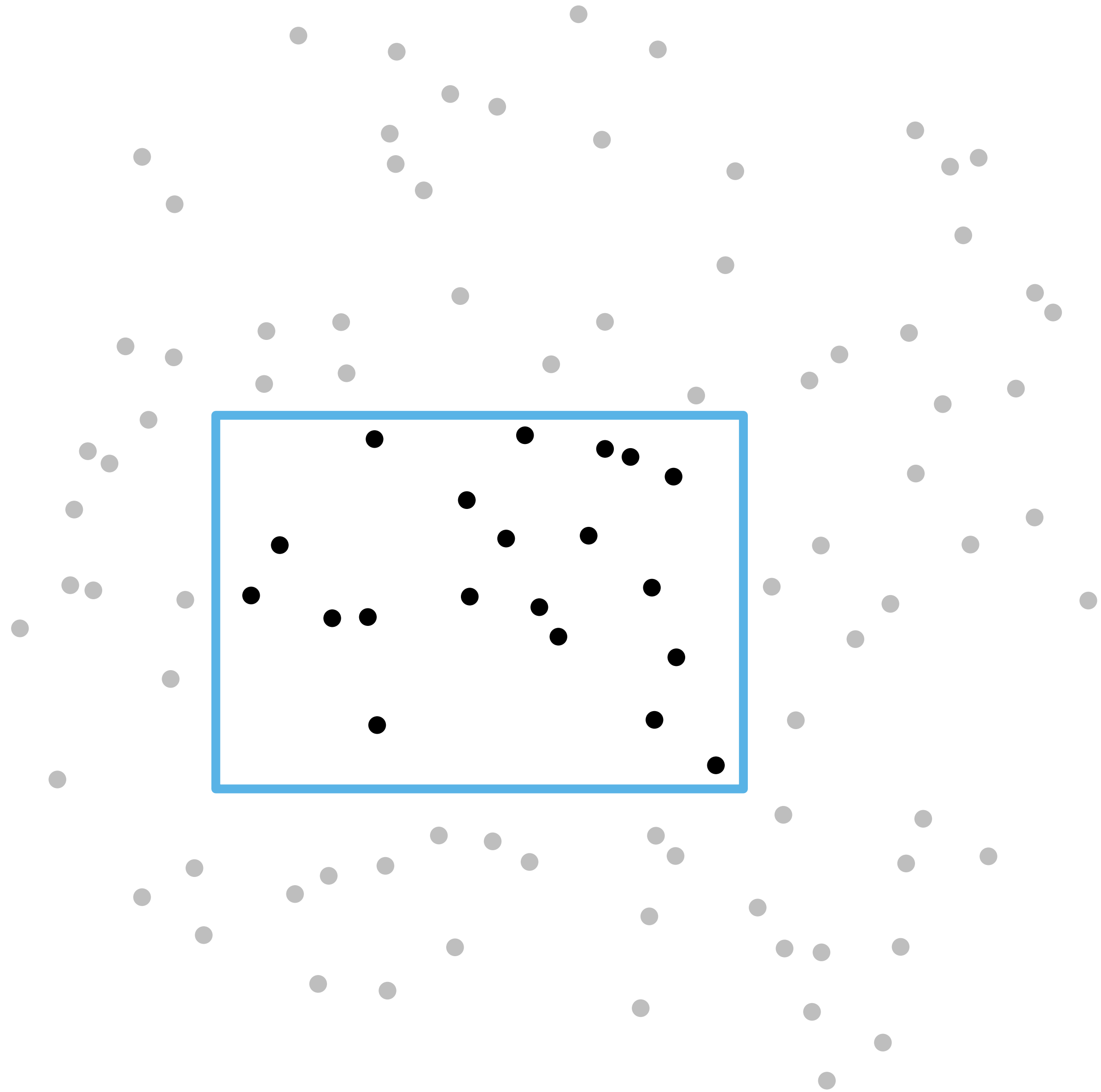


Range diameter



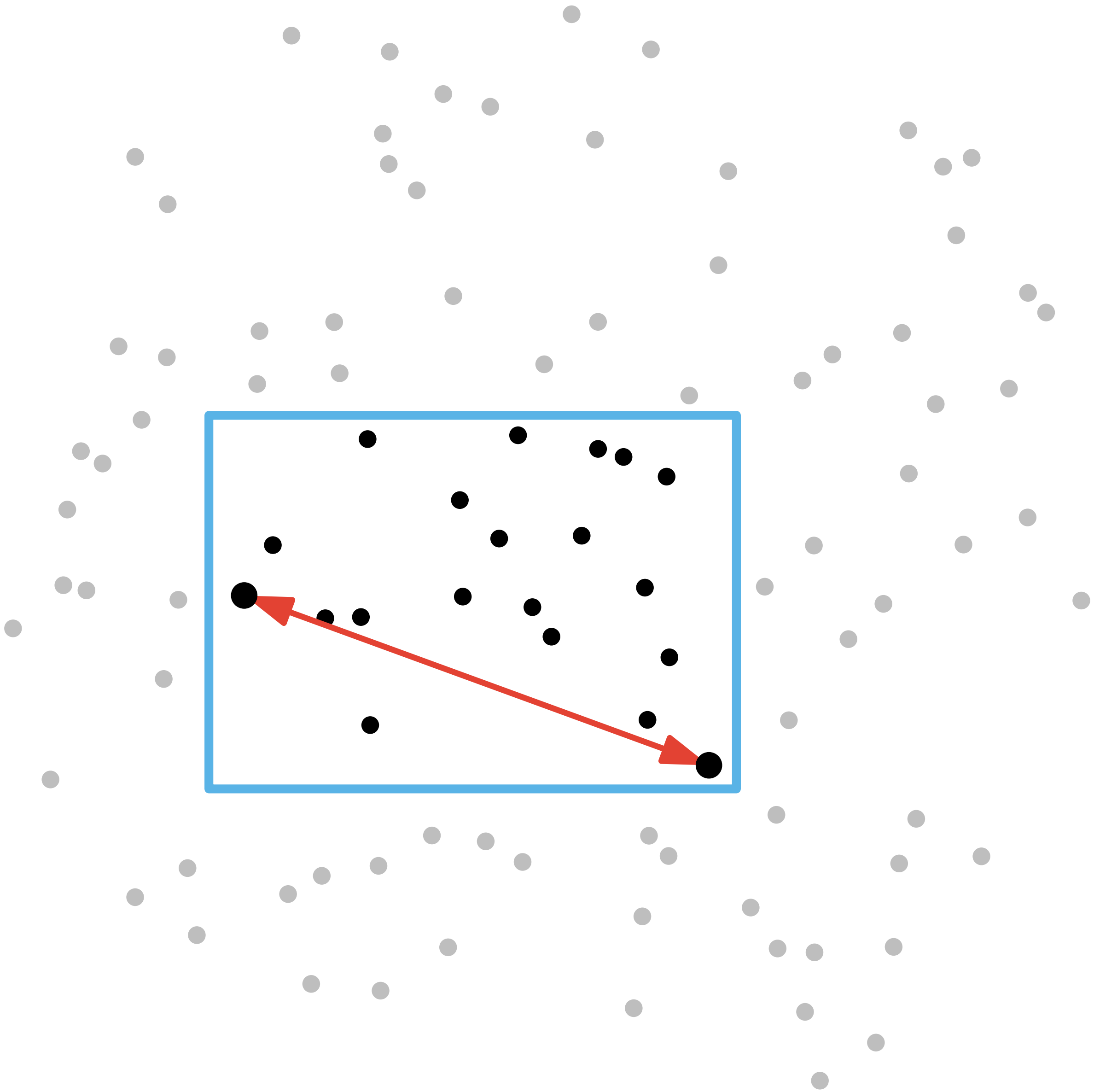
Range diameter

furthest pair distance in range

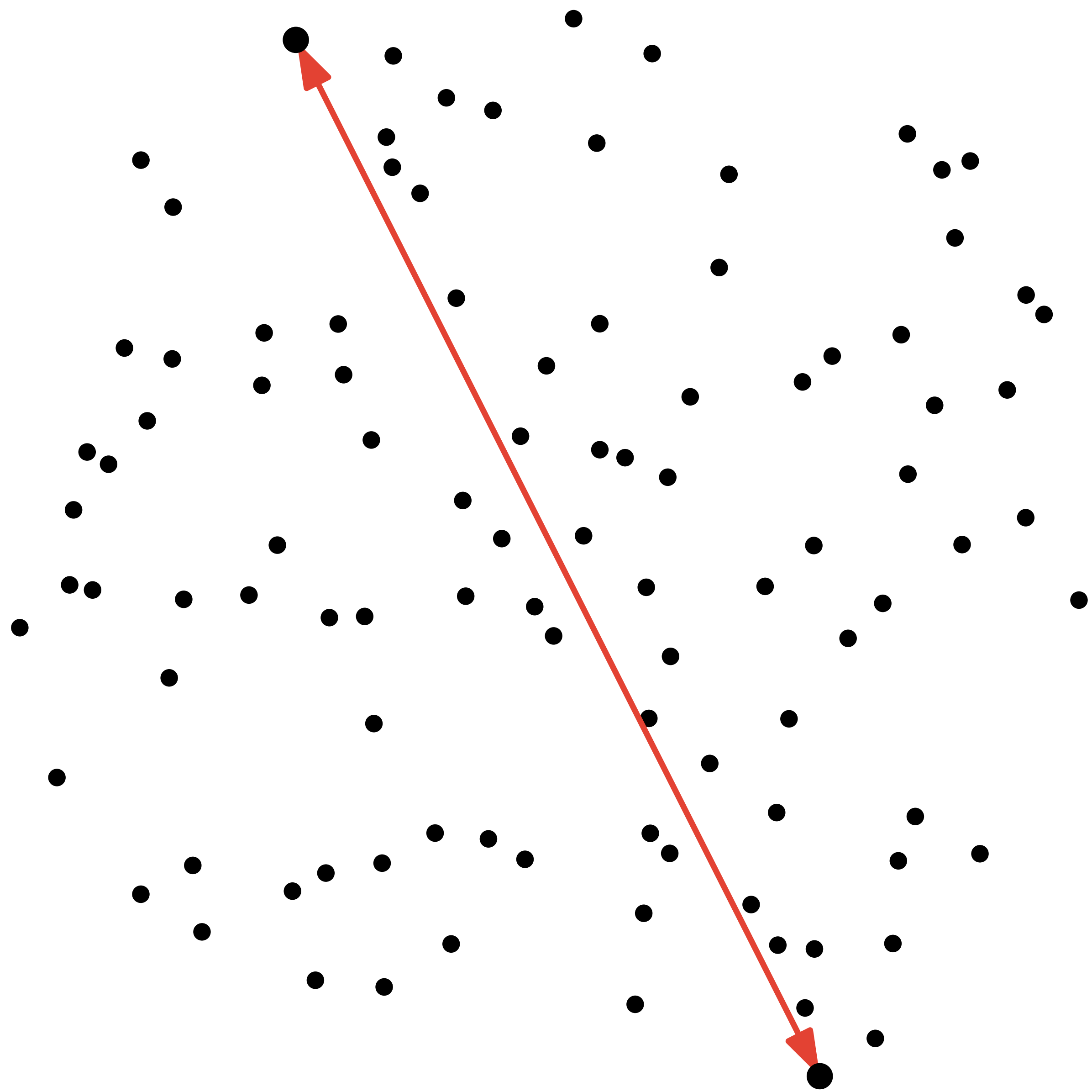


Range diameter

furthest pair distance in range

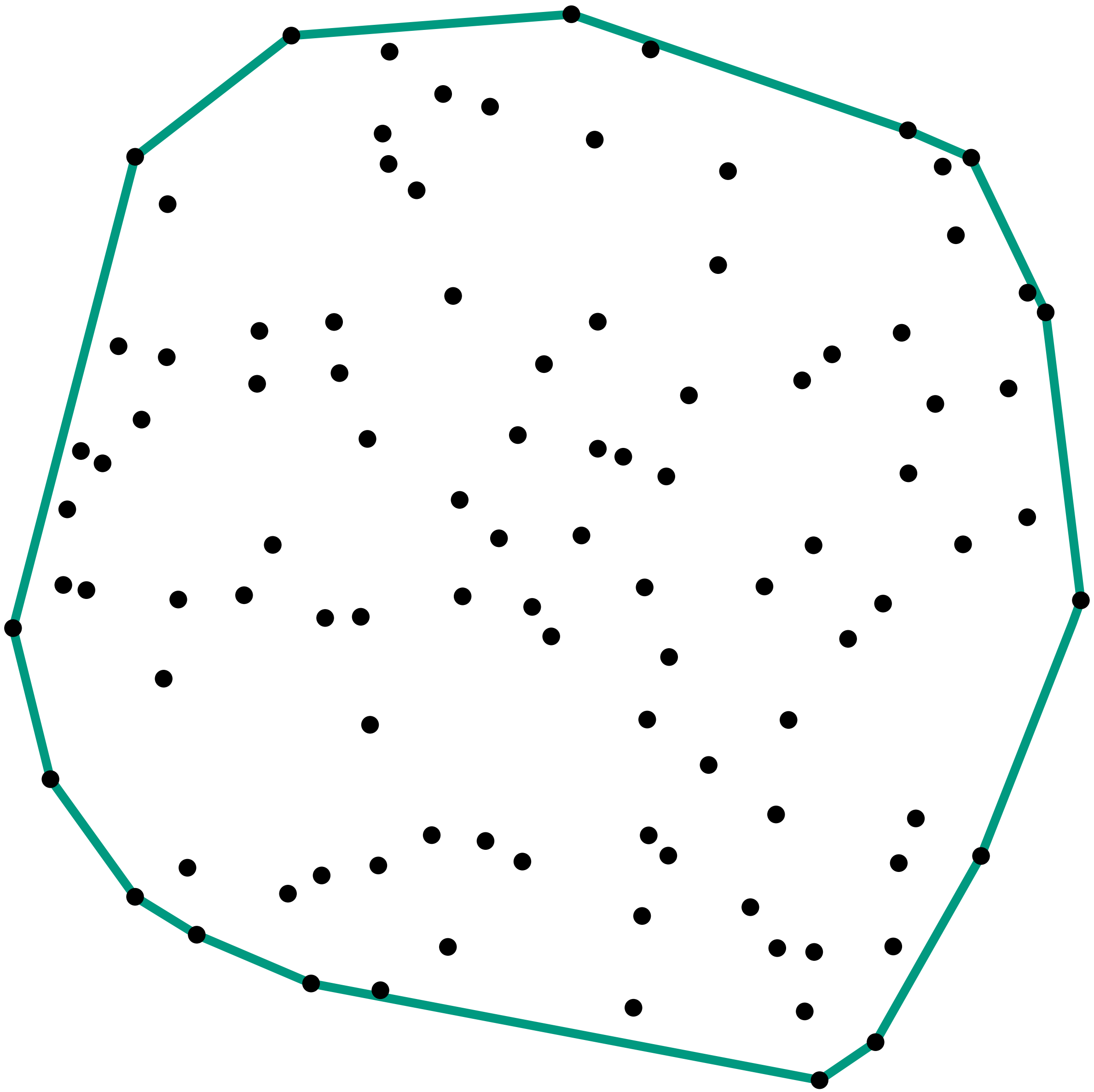


Diameter of a point set



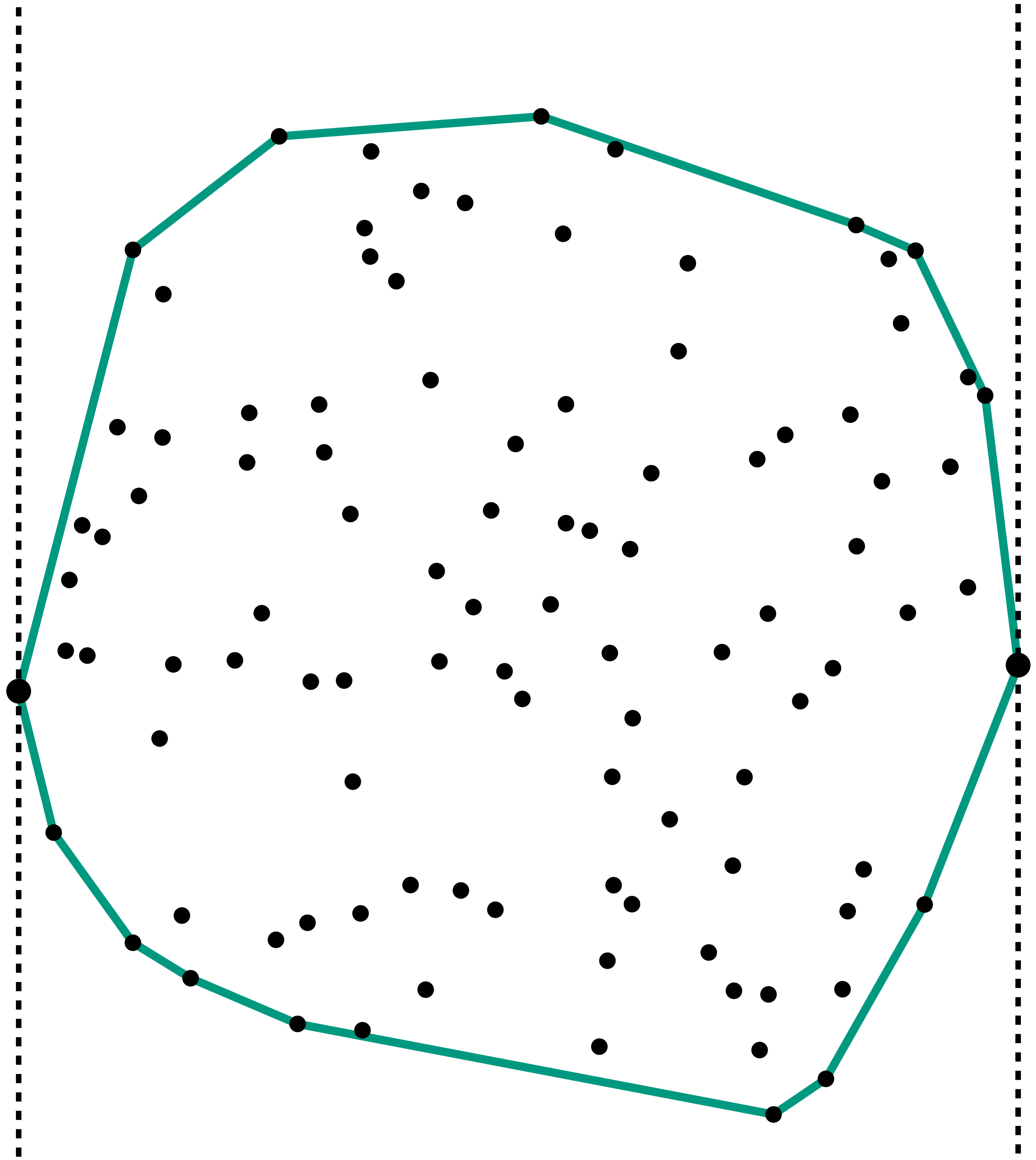
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



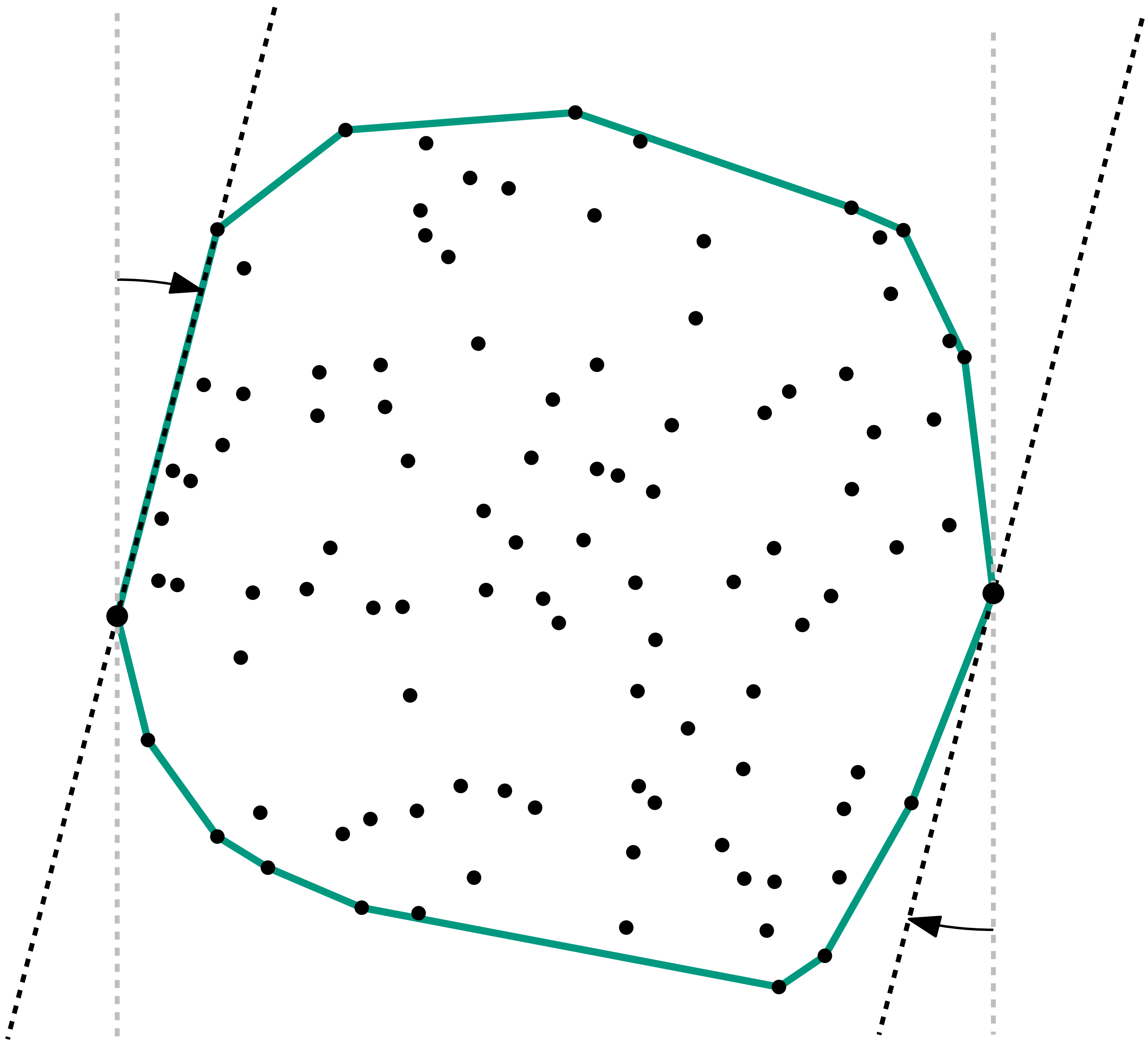
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



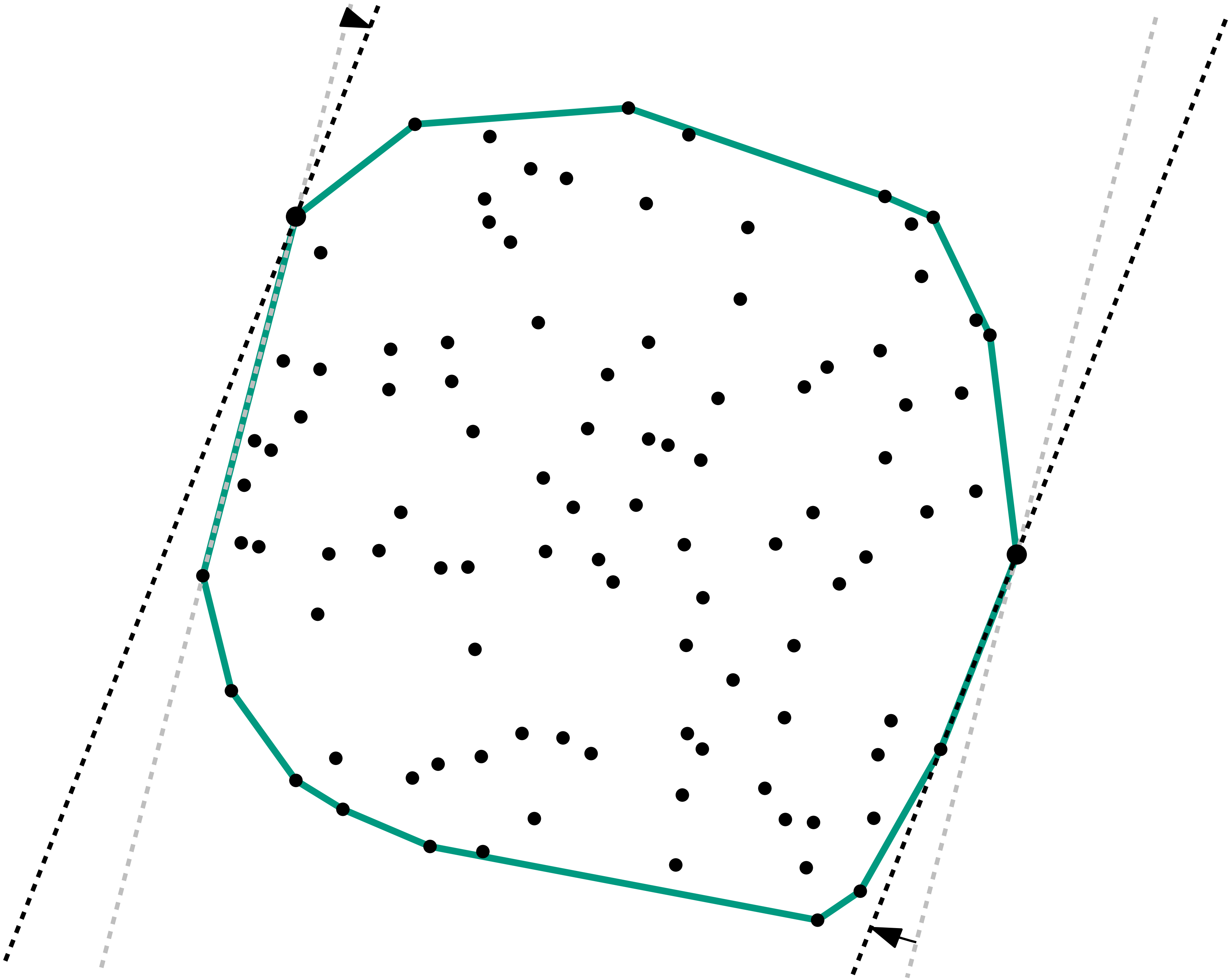
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



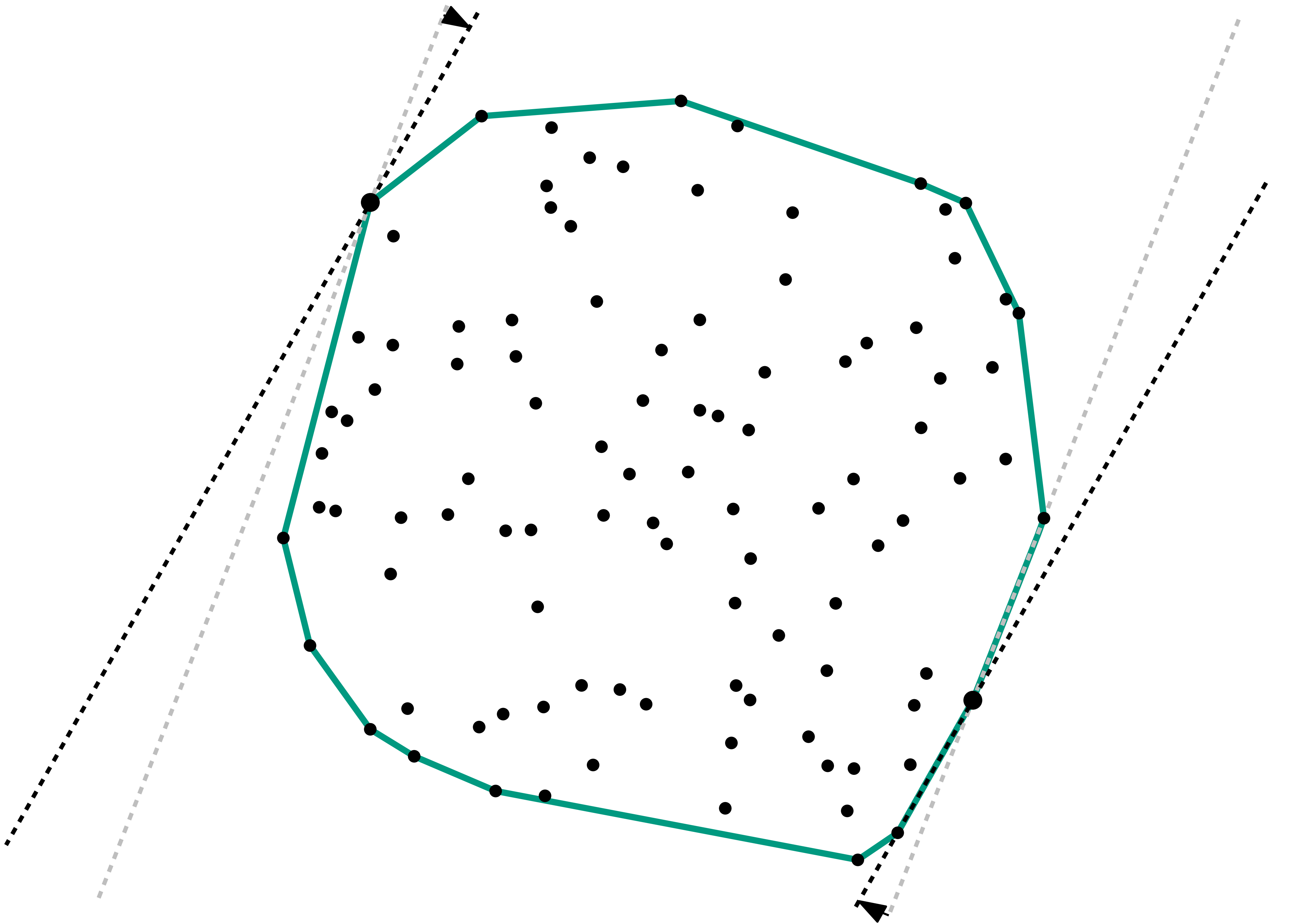
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



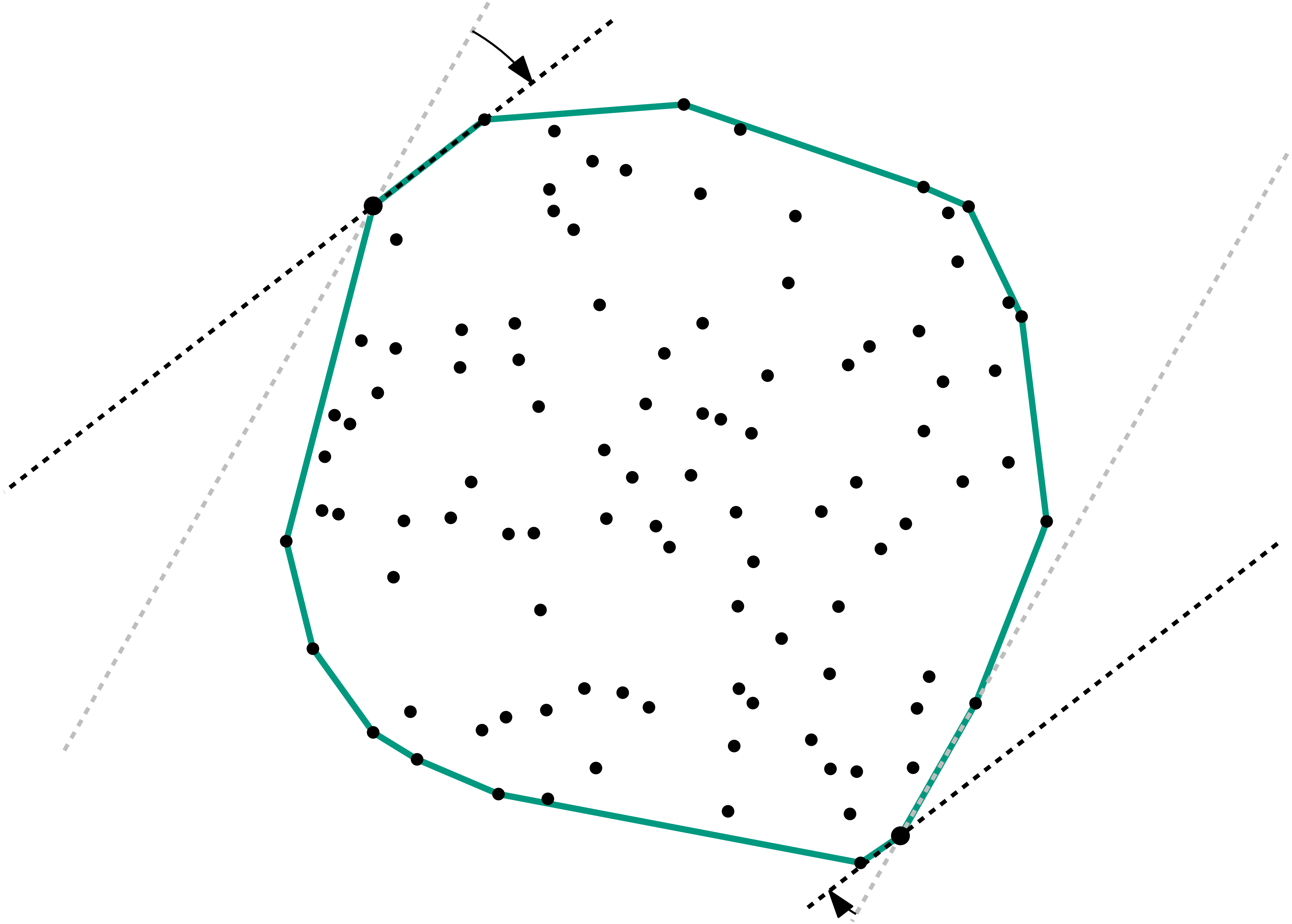
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



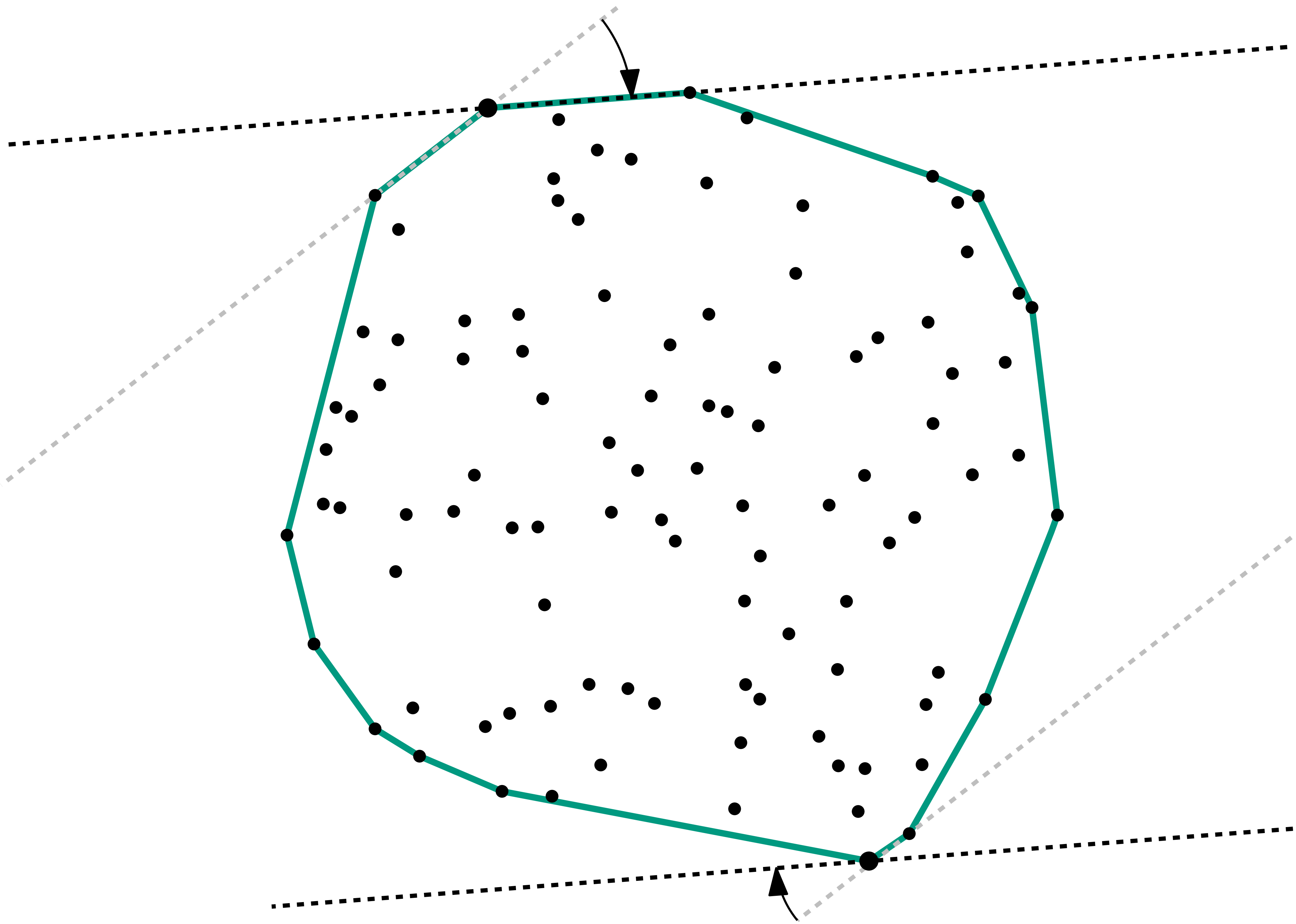
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



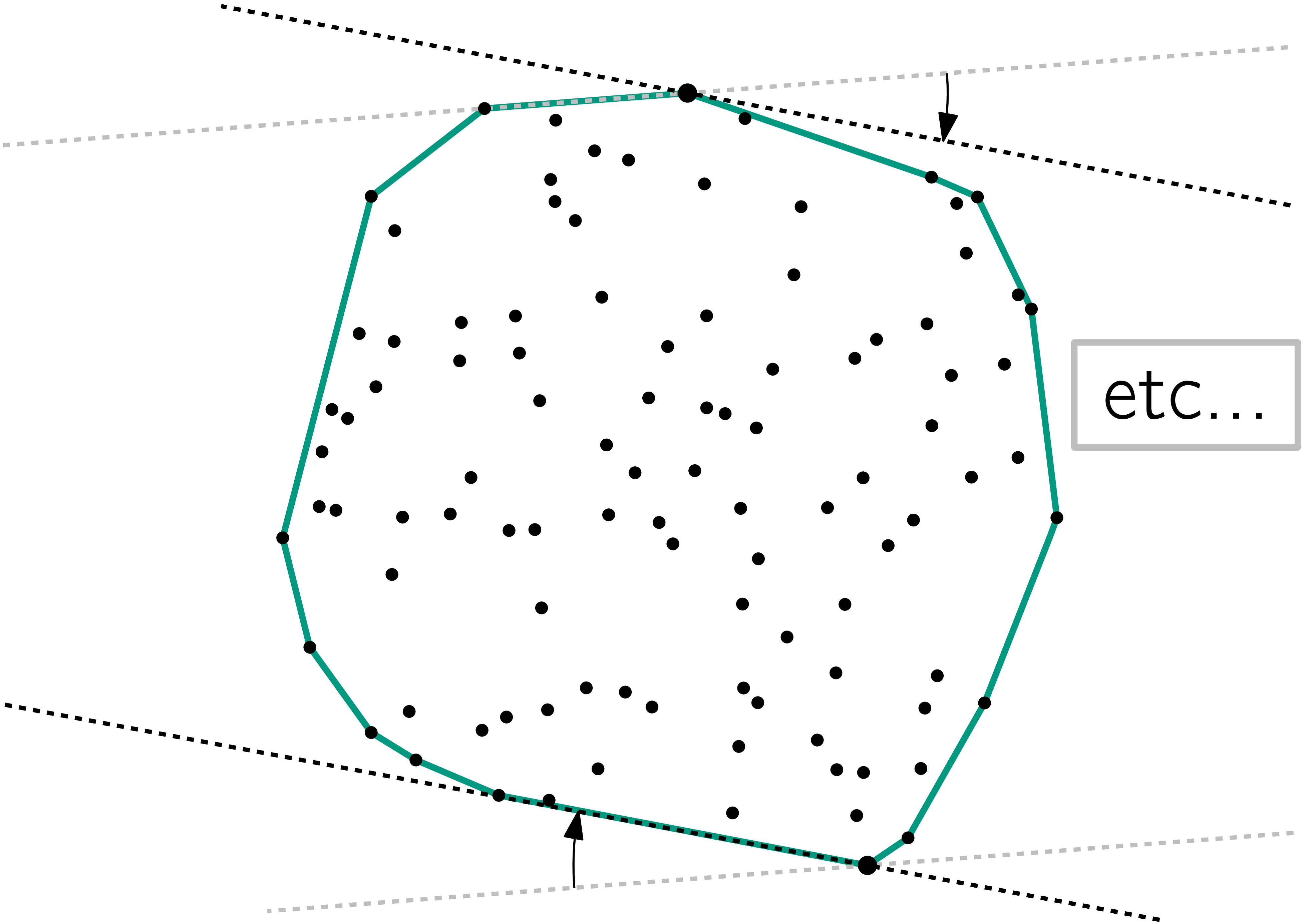
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



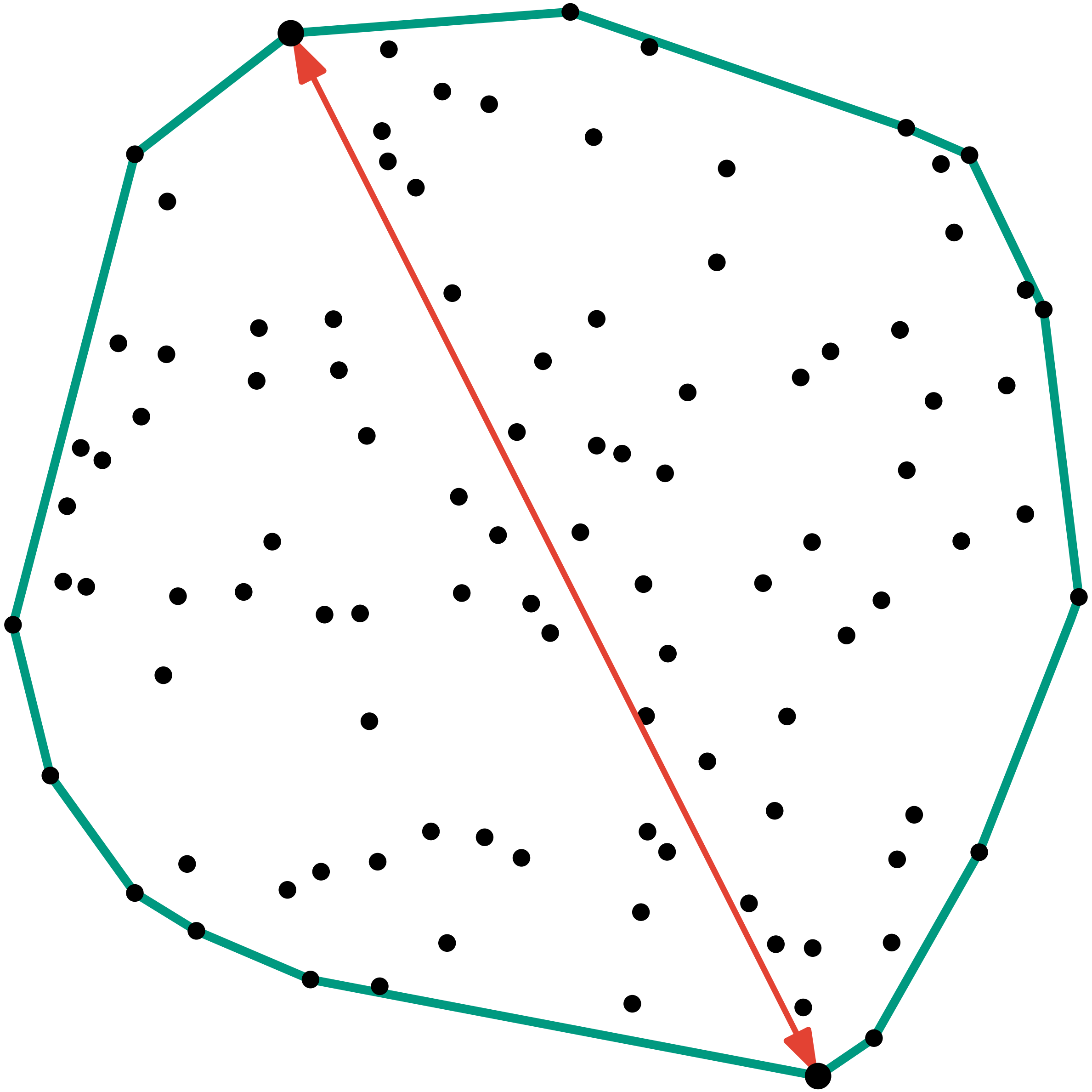
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



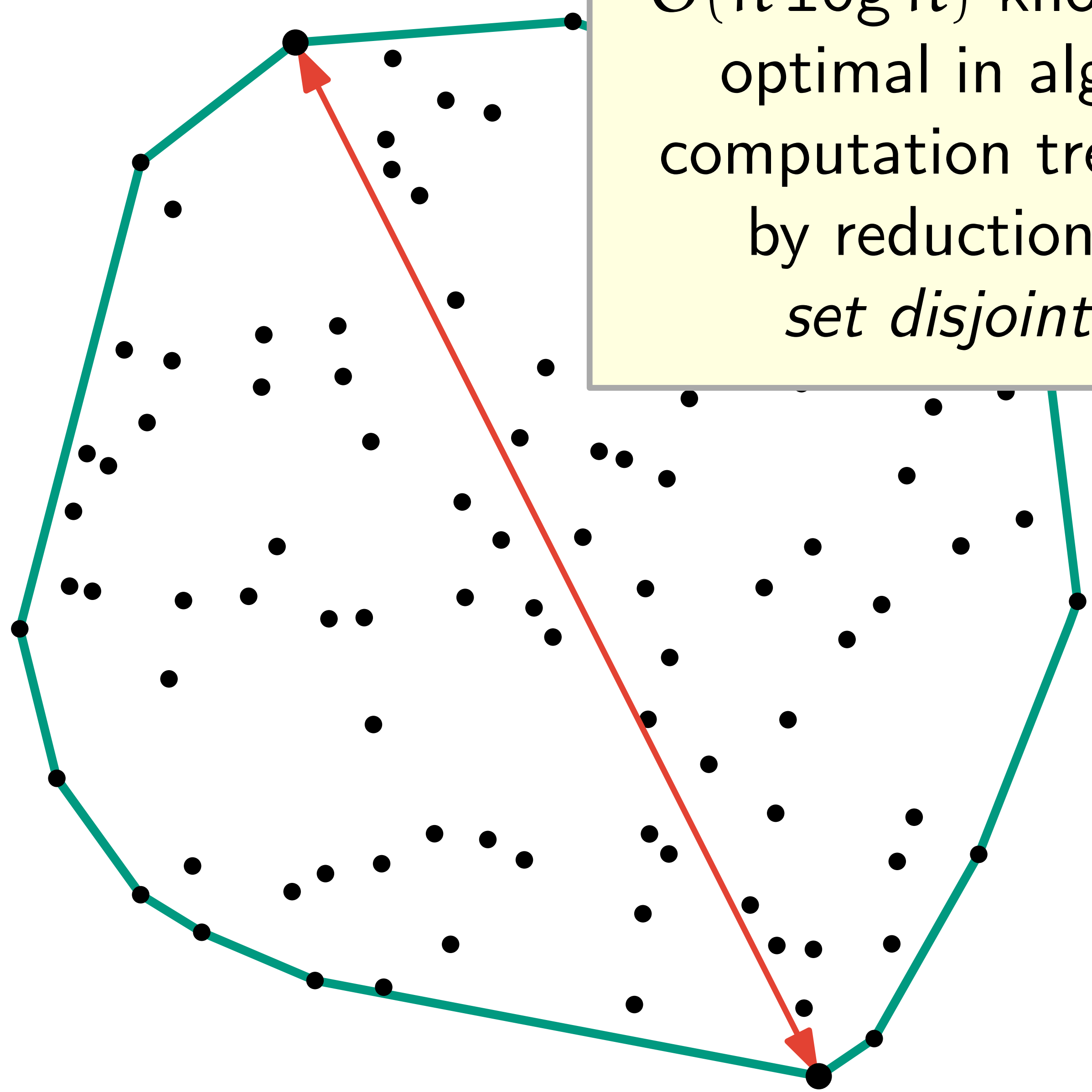
Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



Diameter of a point set

Use classical **rotating calipers** algorithm around convex hull



$O(n \log n)$ known to be optimal in algebraic computation tree model by reduction from *set disjointness*

Previous work

on non-decomposable geometric range aggregate queries

- Range diameter query [Gupta et al. 2009]:
  $O((n + (n/k)^2) \log^2 n)$ space, $O(k \log^5 n)$ query time

k	space	query time
1	$O(n^2 \log^2 n)$	$O(\log^5 n)$
$\sqrt{n}$	$O(n \log^2 n)$	$O(\sqrt{n} \log^5 n)$

Previous work

on non-decomposable geometric range aggregate queries

- Range diameter query [Gupta et al. 2009]:
$O((n + (n/k)^2) \log^2 n)$ space, $O(k \log^5 n)$ query time

k	space	query time
1	$O(n^2 \log^2 n)$	$O(\log^5 n)$
$\sqrt{n}$	$O(n \log^2 n)$	$O(\sqrt{n} \log^5 n)$

- Closest pair query [Shan et al. 2003, Gupta 2005, Sharathkumar and Gupta 2007, Gupta et al. 2009]
Main result: $O(n \log^5 n)$ space, $O(\log^2 n)$ query time

- Width query (narrowest strip enclosing points) [Gupta et al. 2009]:
$(1 + \delta)$ approximation in $O(\frac{1}{\sqrt{\delta}} n \log^2 n)$ space,
$O(\frac{1}{\sqrt{\delta}} \log^3 n)$ query time

Previous work

on non-decomposable geometric range aggregate queries

- Range diameter query [Gupta et al. 2009]:
  $O((n+(n/k)^2)\log^2 n)$ space, $O(k\log^5 n)$ query time

k	space	query time
1	$O(n^2\log^2 n)$	$O(\log^5 n)$
$\sqrt{n}$	$O($	$\log^5 n)$

Quadratic space necessary for polylog query time?

- Closest pair qu , Gupta 2005,
  Sharathkumar ta et al. 2009]
  Main result: $O(n\log^5 n)$ space, $O(\log^2 n)$ query time

- Width query (narrowest strip enclosing points) [Gupta et al. 2009]:
  $(1+\delta)$ approximation in $O(\frac{1}{\sqrt{\delta}}n\log^2 n)$ space,
  $O(\frac{1}{\sqrt{\delta}}\log^3 n)$ query time

Outline

- Reduction from *set intersection* to range diameter
  $\rightarrow$ strong evidence for hardness

Outline

- Reduction from *set intersection* to range diameter
  $\rightarrow$ strong evidence for hardness

What about special cases?

Outline

- Reduction from *set intersection* to range diameter
  $\rightarrow$ strong evidence for hardness

What about special cases?

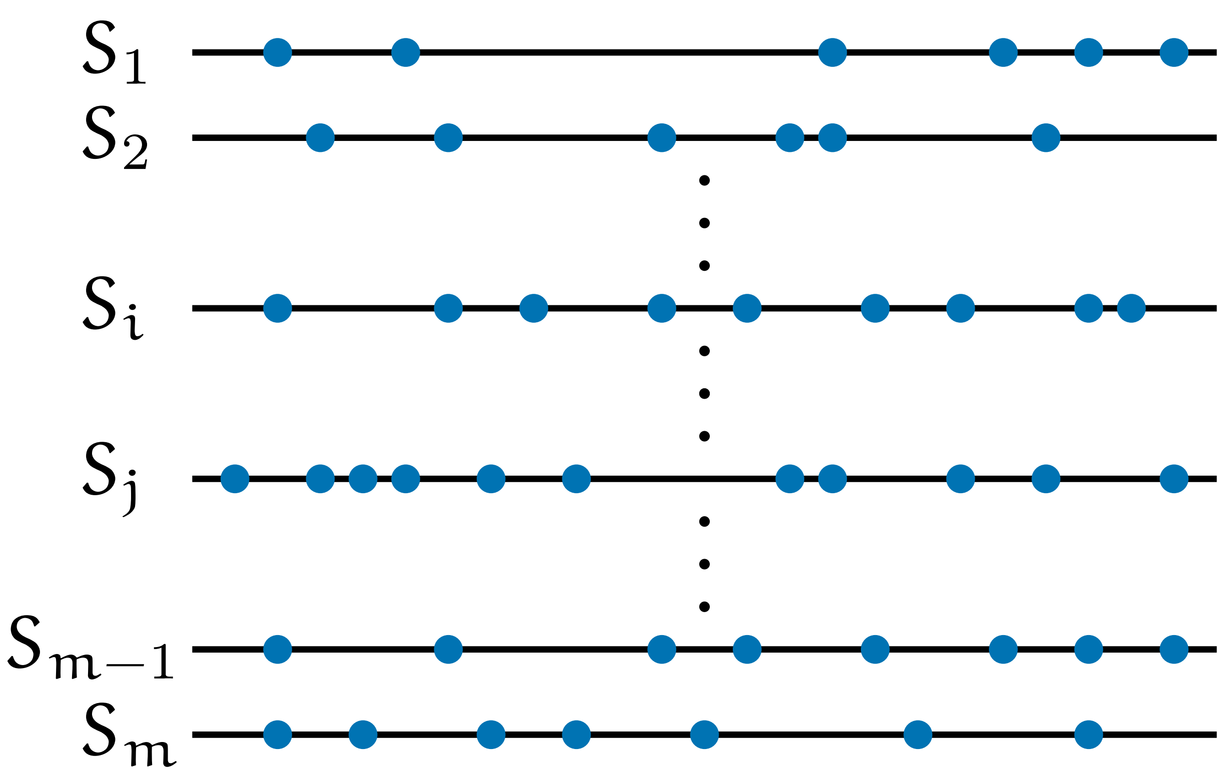
- Data structure for range diameter queries on points in convex position:
  $O(n \log n)$ space, $O(\log n)$ query time

Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$? $\qquad n = \sum_i |S_i|$



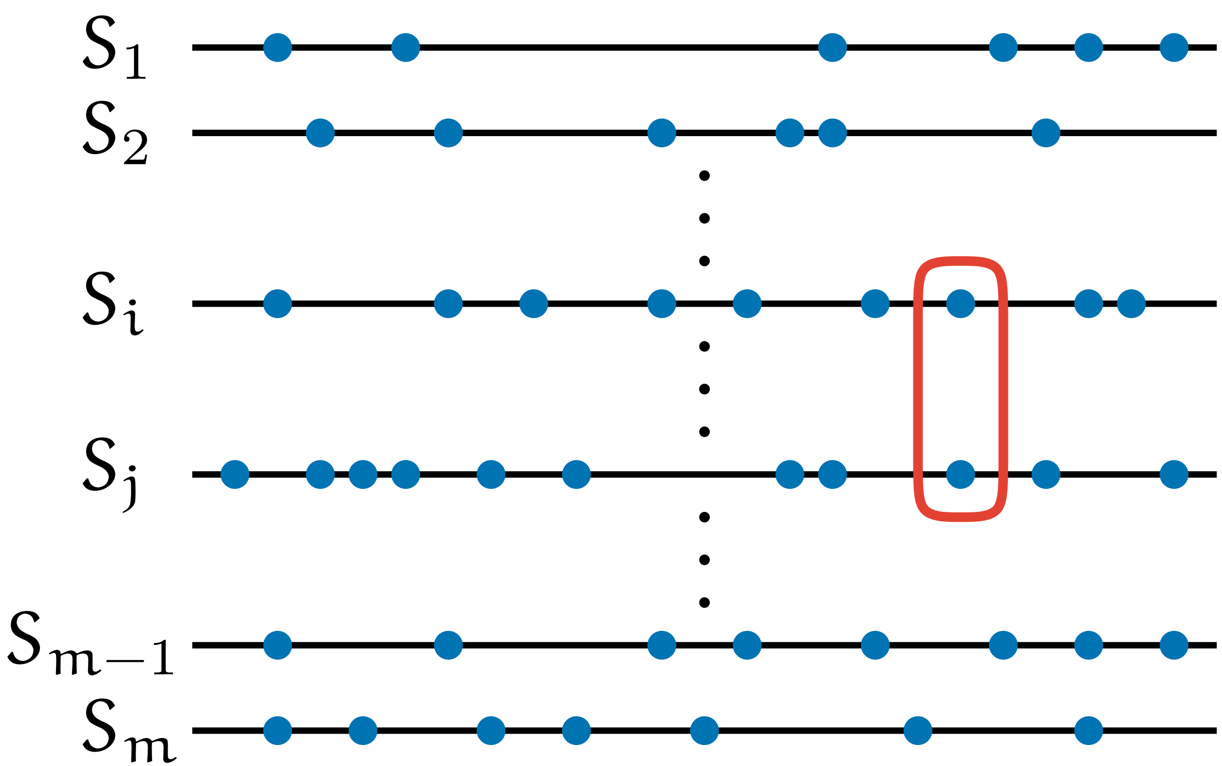
Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

$n = \sum_i |S_i|$



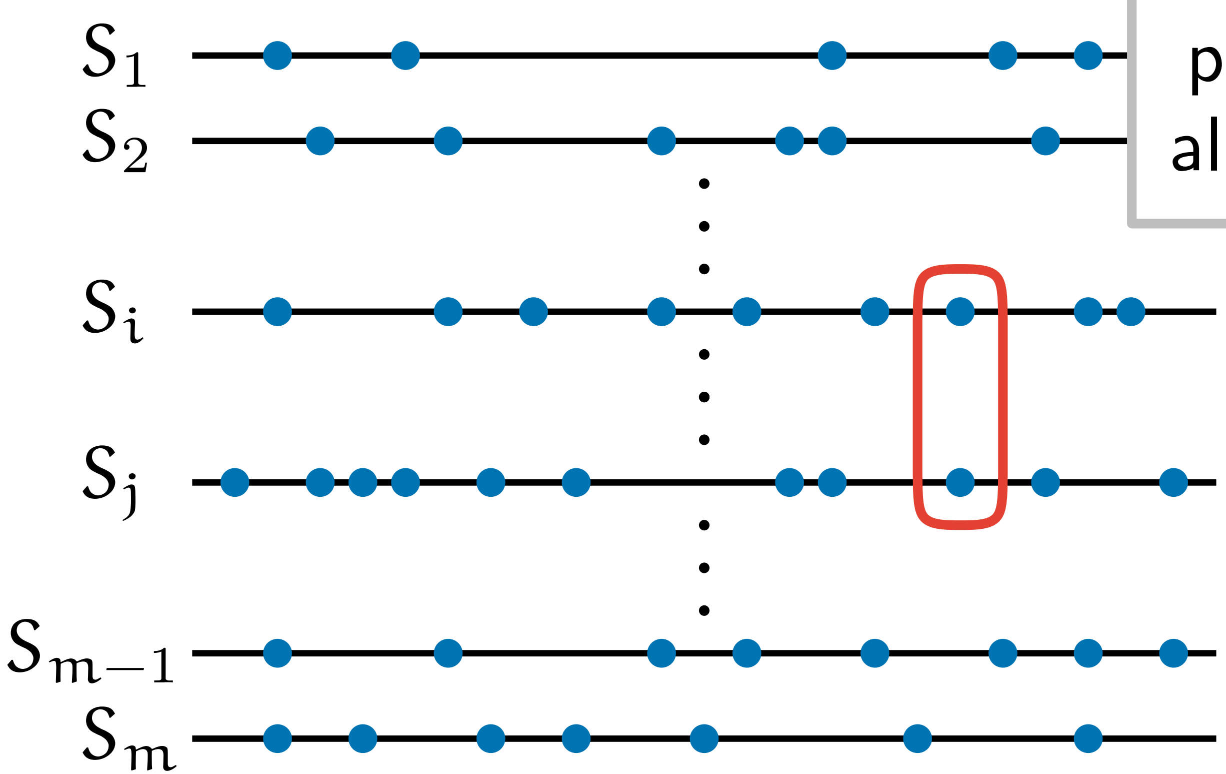
Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

$n = \sum_i |S_i|$



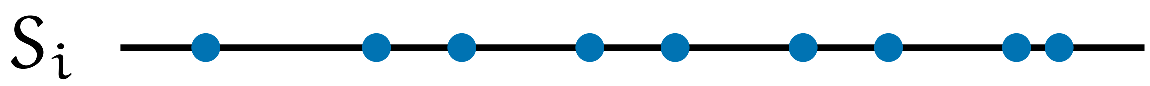
Conjecture:
polylog query time $\Rightarrow$
almost quadratic space

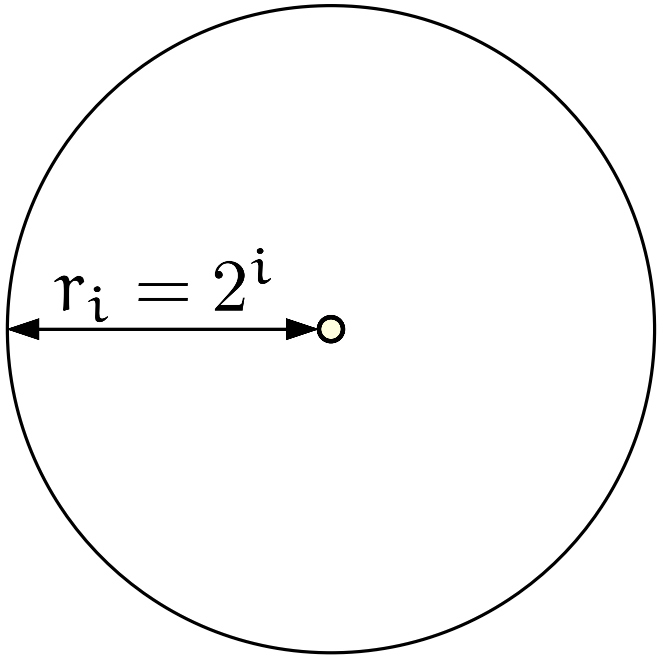
Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?



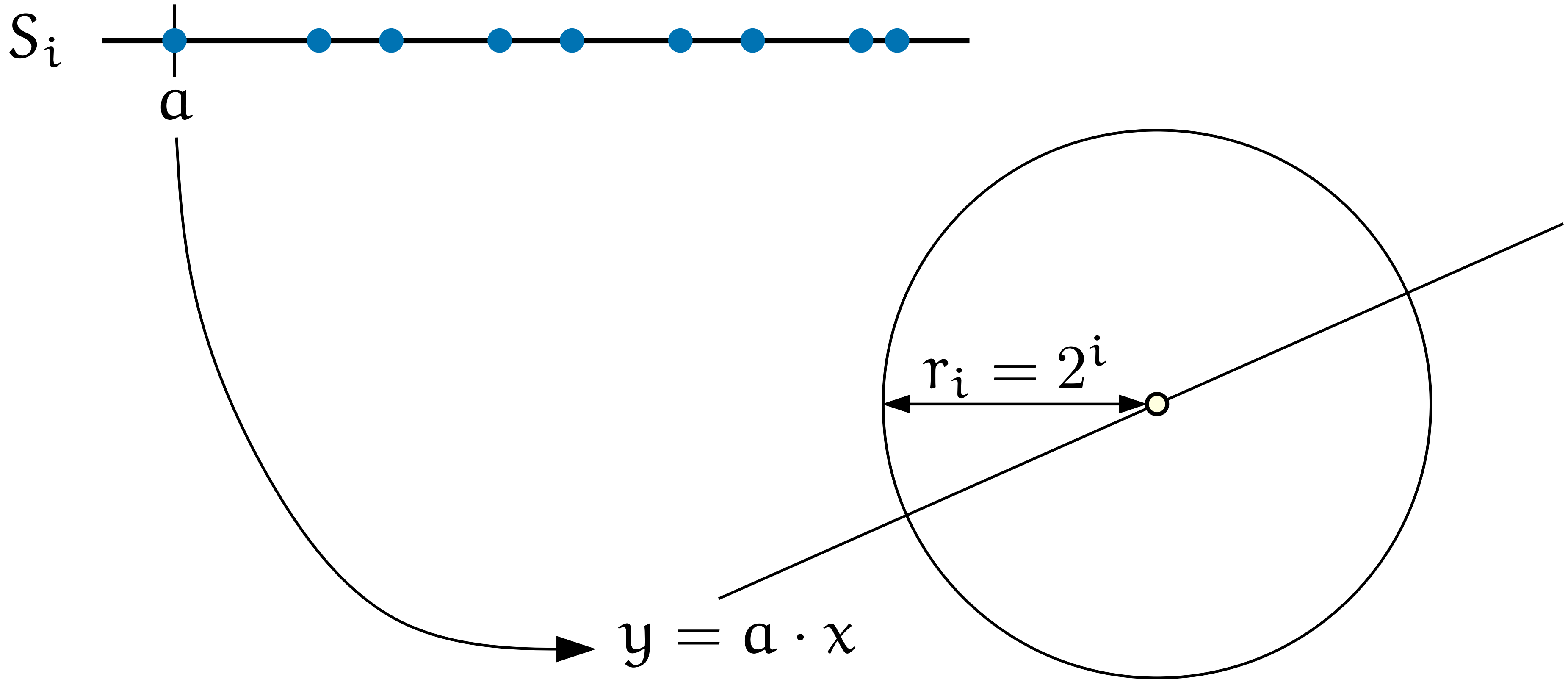


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

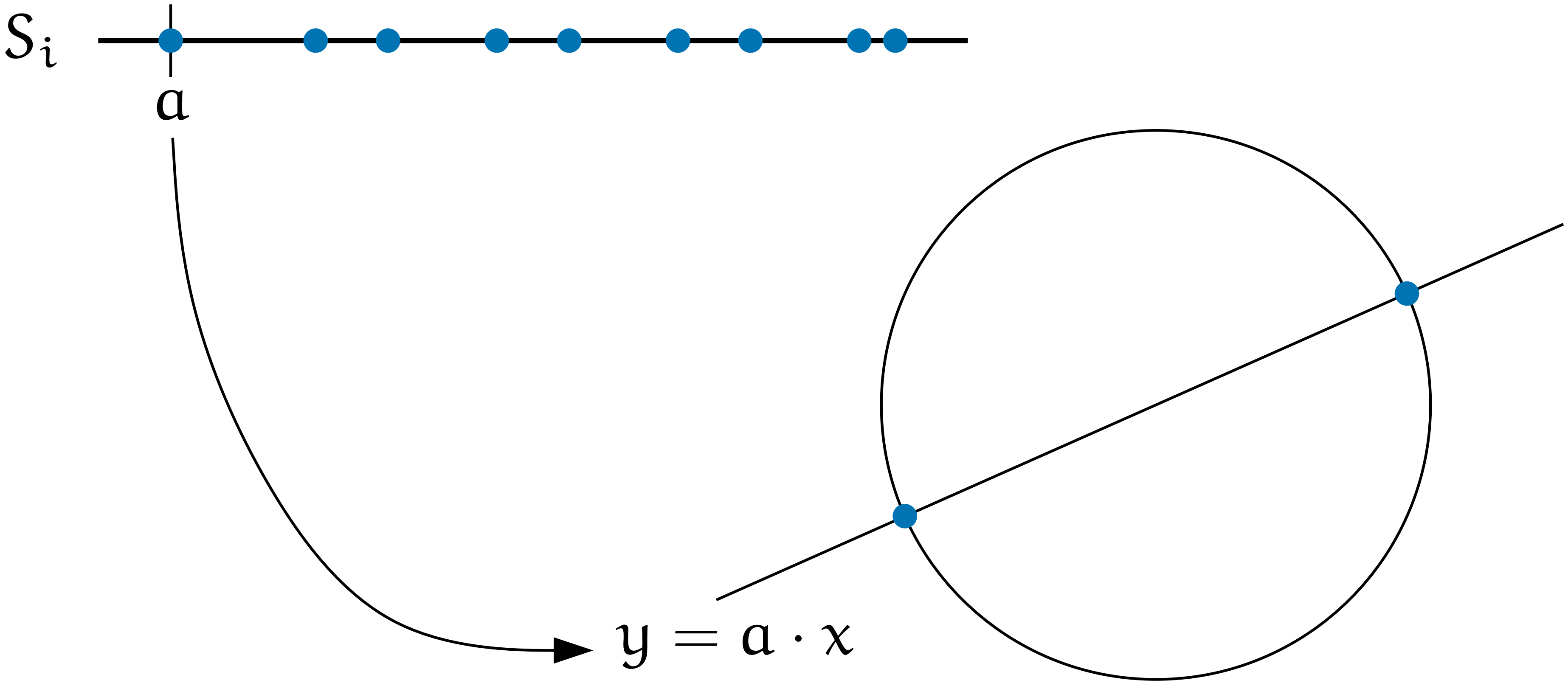


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

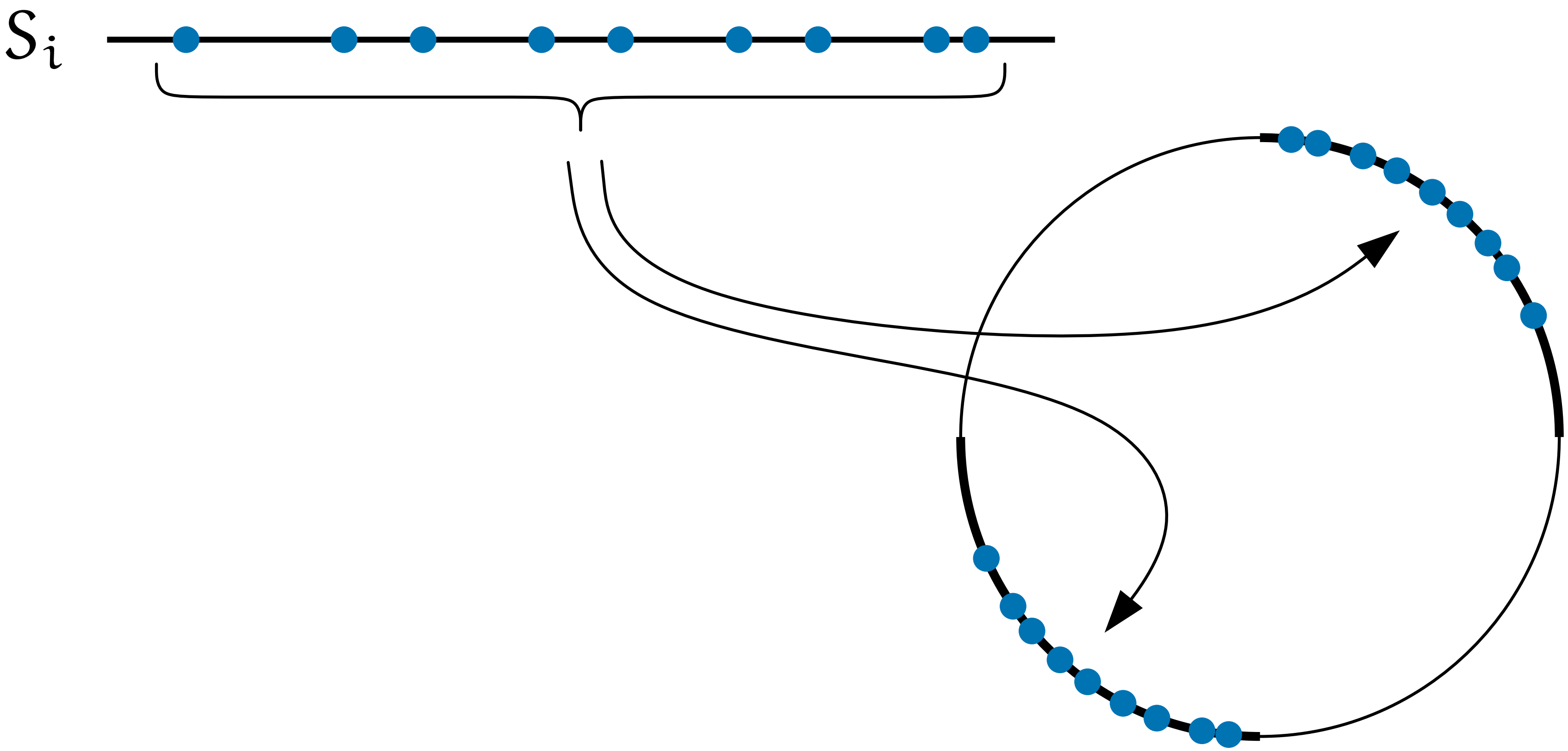


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

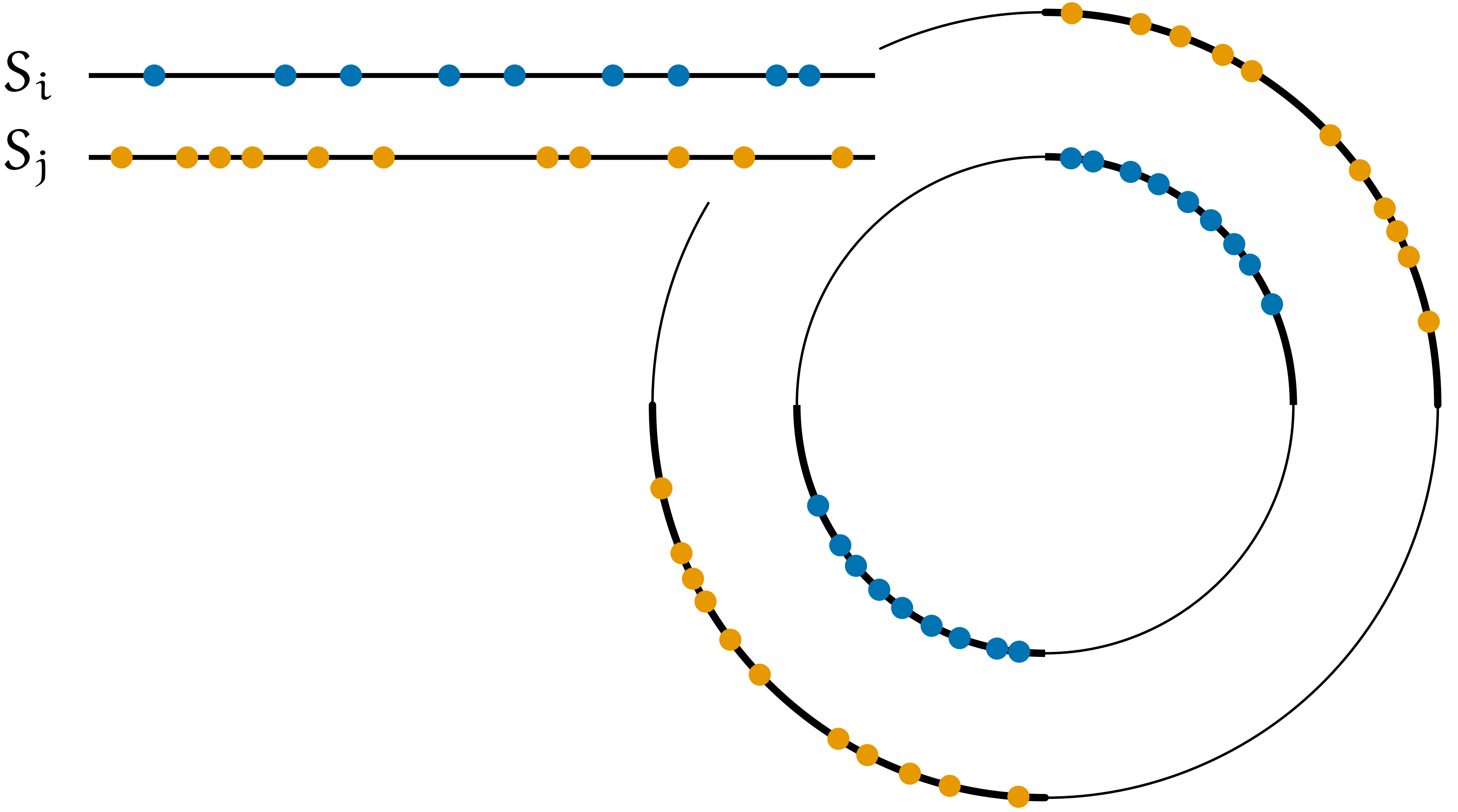


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

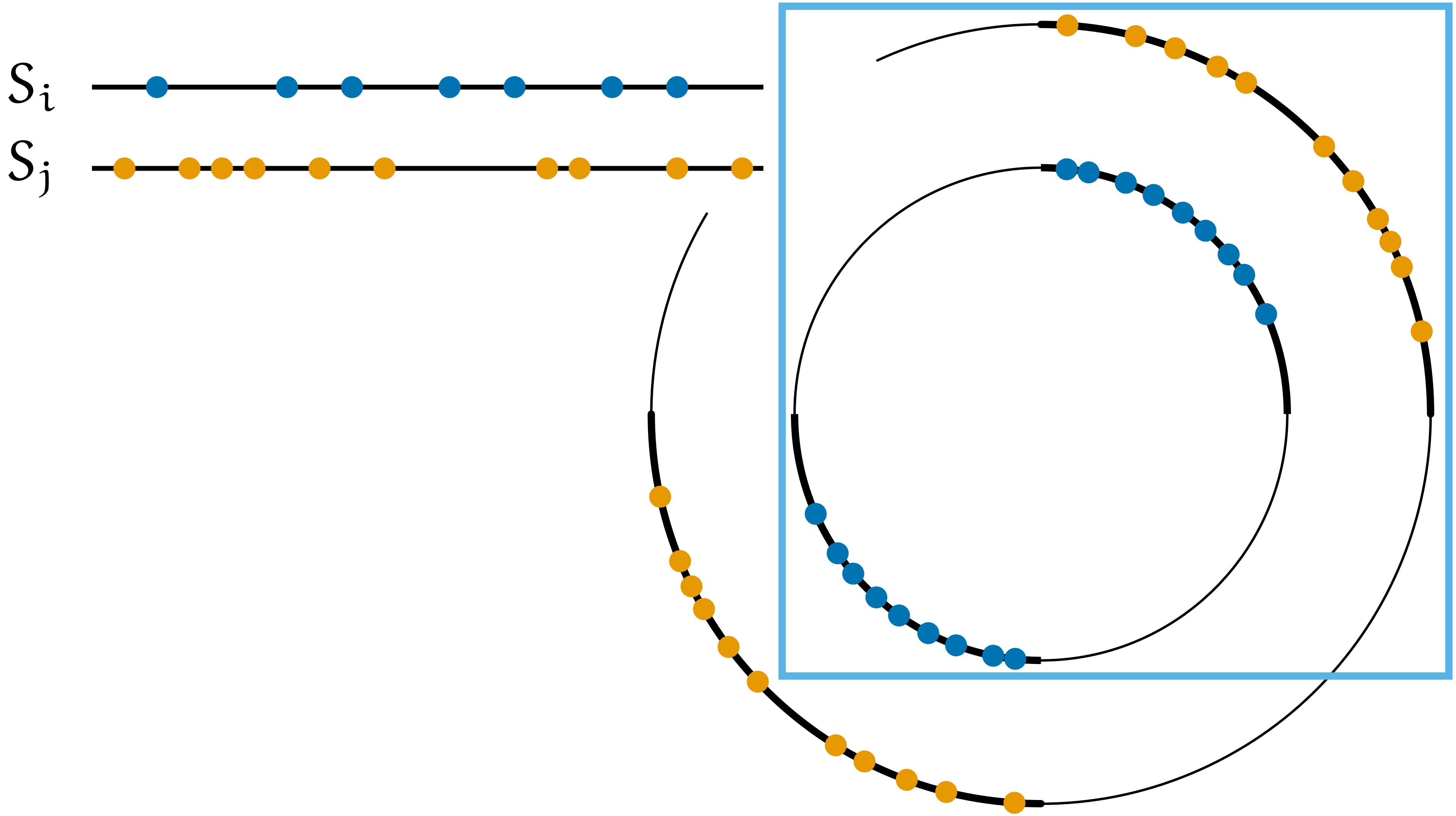


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

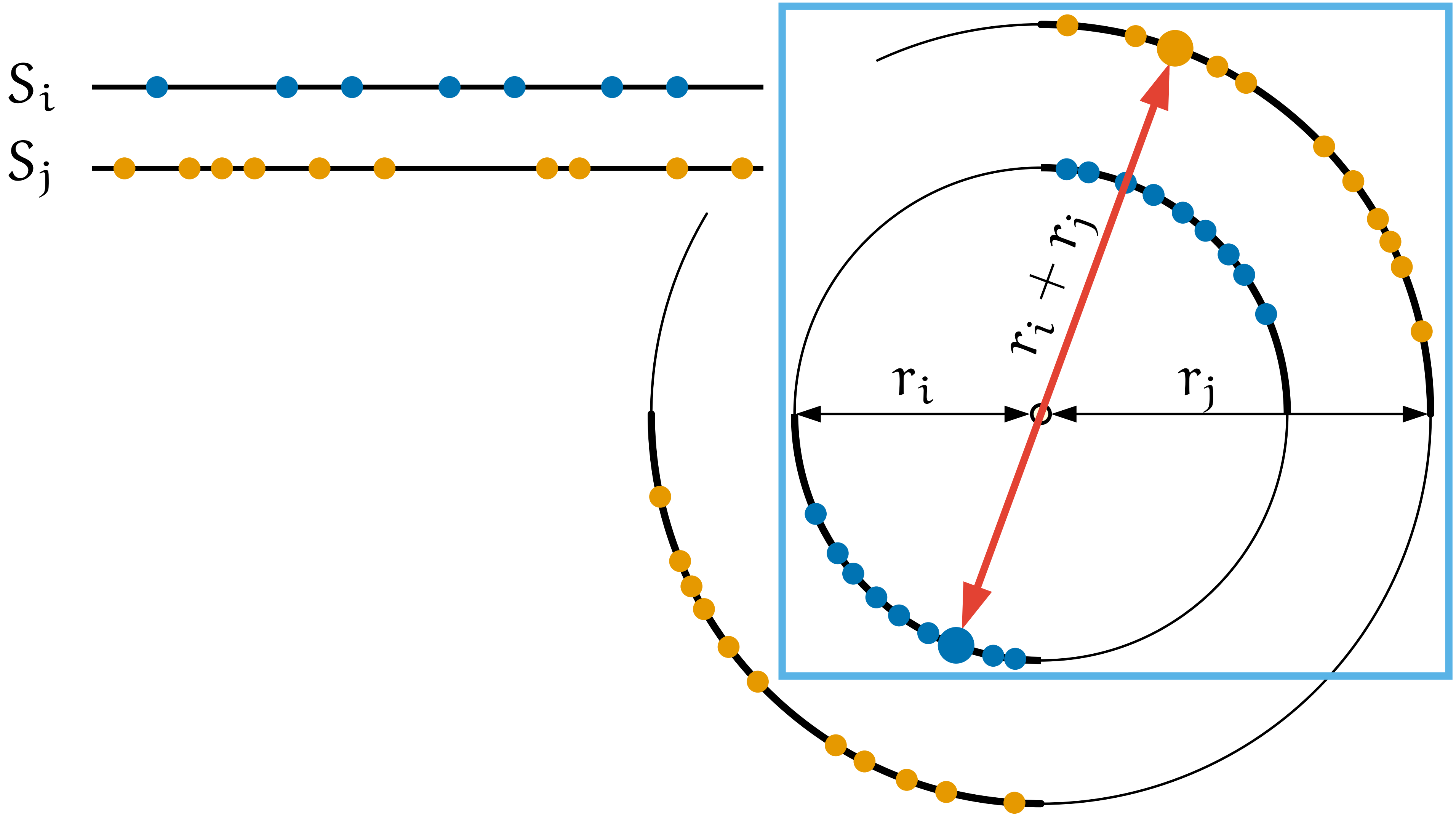


Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?

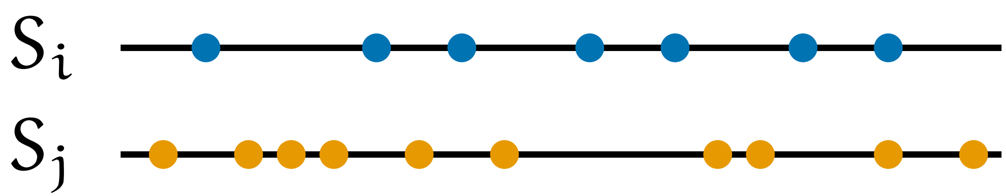


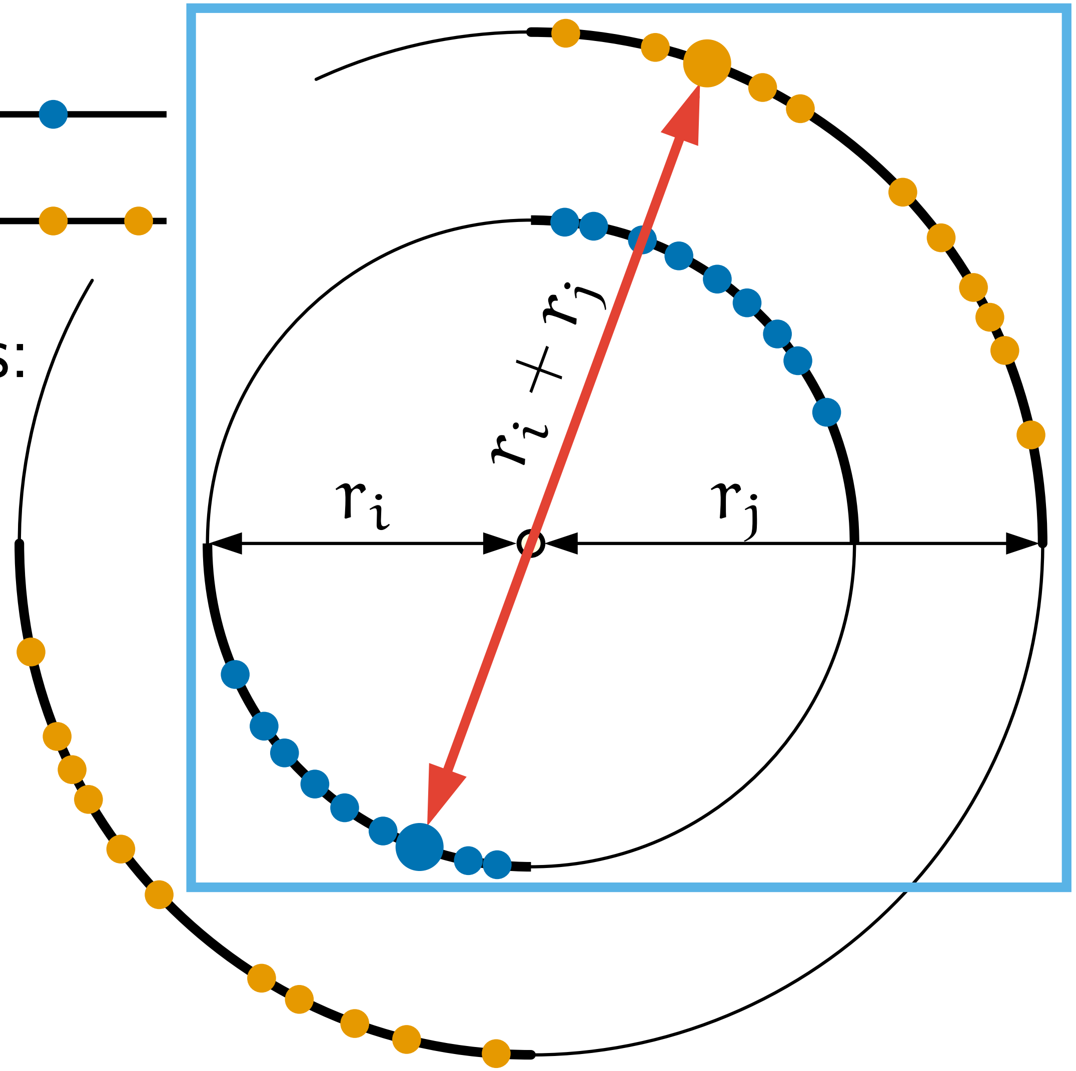
Why range diameter is hard

A reduction from set intersection

Set intersection problem

- Input: m sets of positive real numbers $S_1, \ldots, S_m$
- Query: $S_i \cap S_j = \emptyset$?





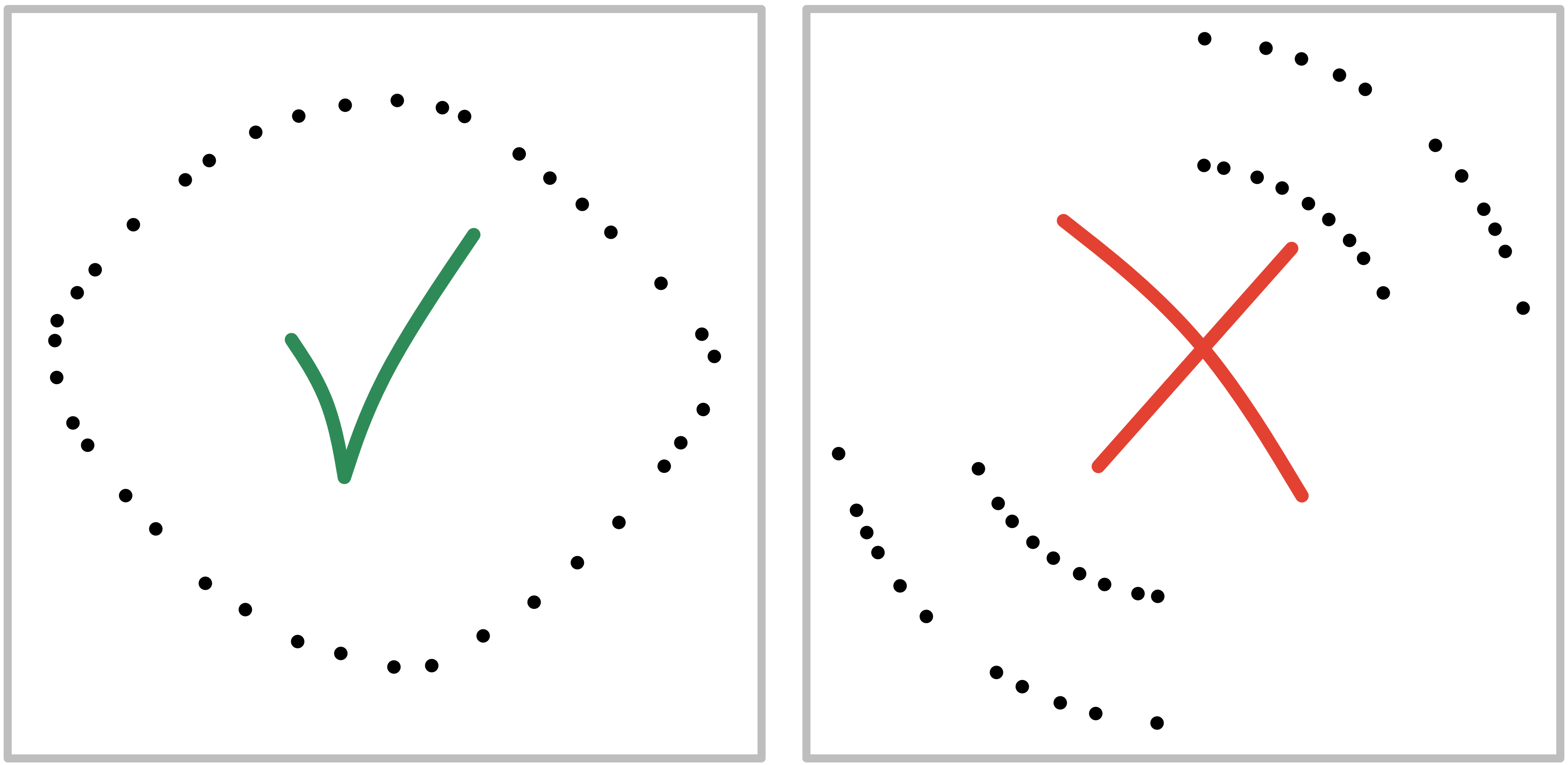
Set intersection conjectures:

- $O(1)$ query $\Rightarrow$ $\tilde{\Omega}(m^2)$ space
- $O(\log^{O(1)} n)$ query $\Rightarrow$ $\Omega(m^{2-\varepsilon})$ space

[Pătrașcu and Roditty, FOCS 2010]

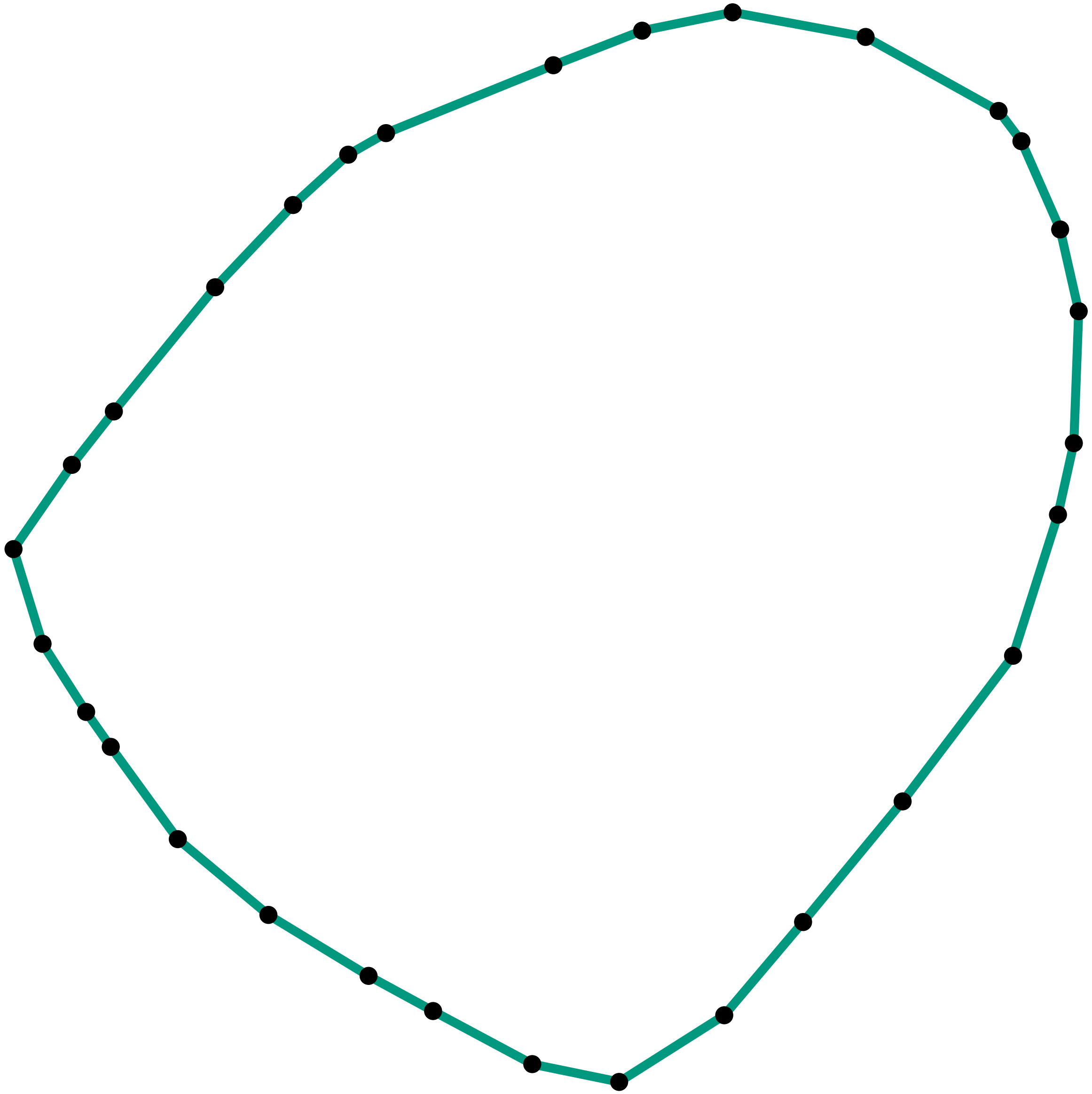
Points in convex position

If general problem is hard, what about special cases, such as points in convex position?



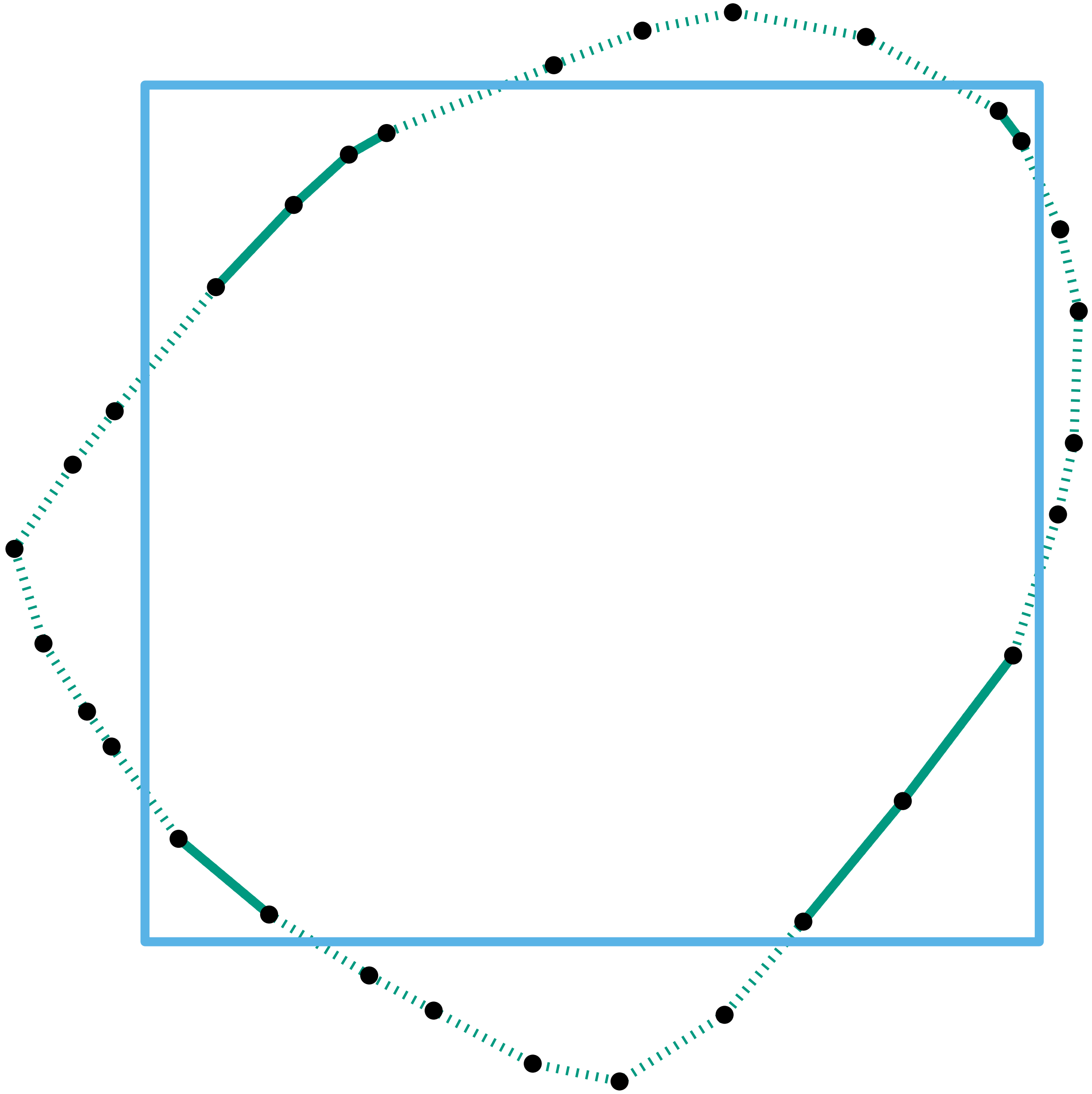
Note: Point set from reduction not in convex position

Points in convex position



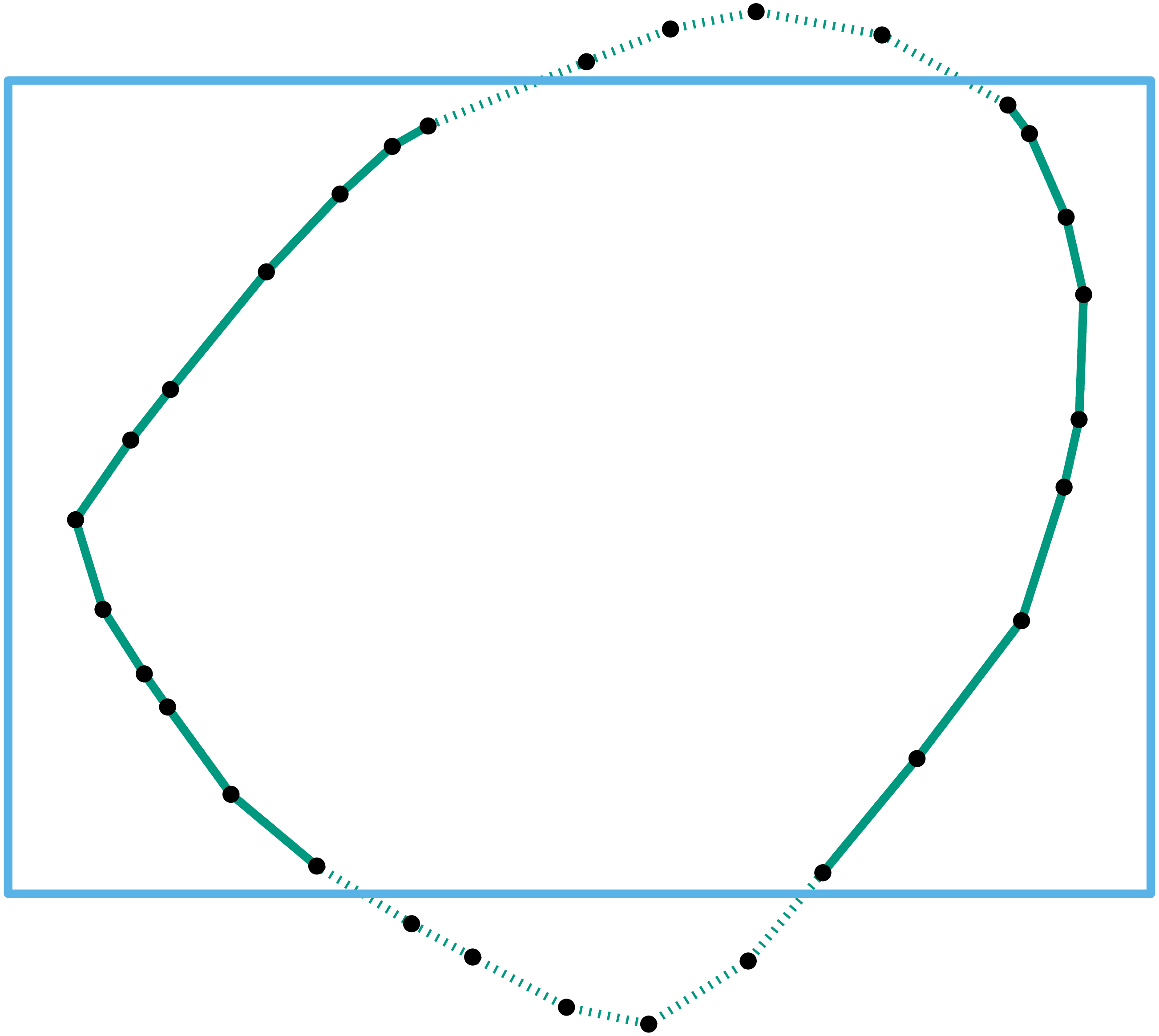
Points in convex position

At most four *sections* of polygon intersect query



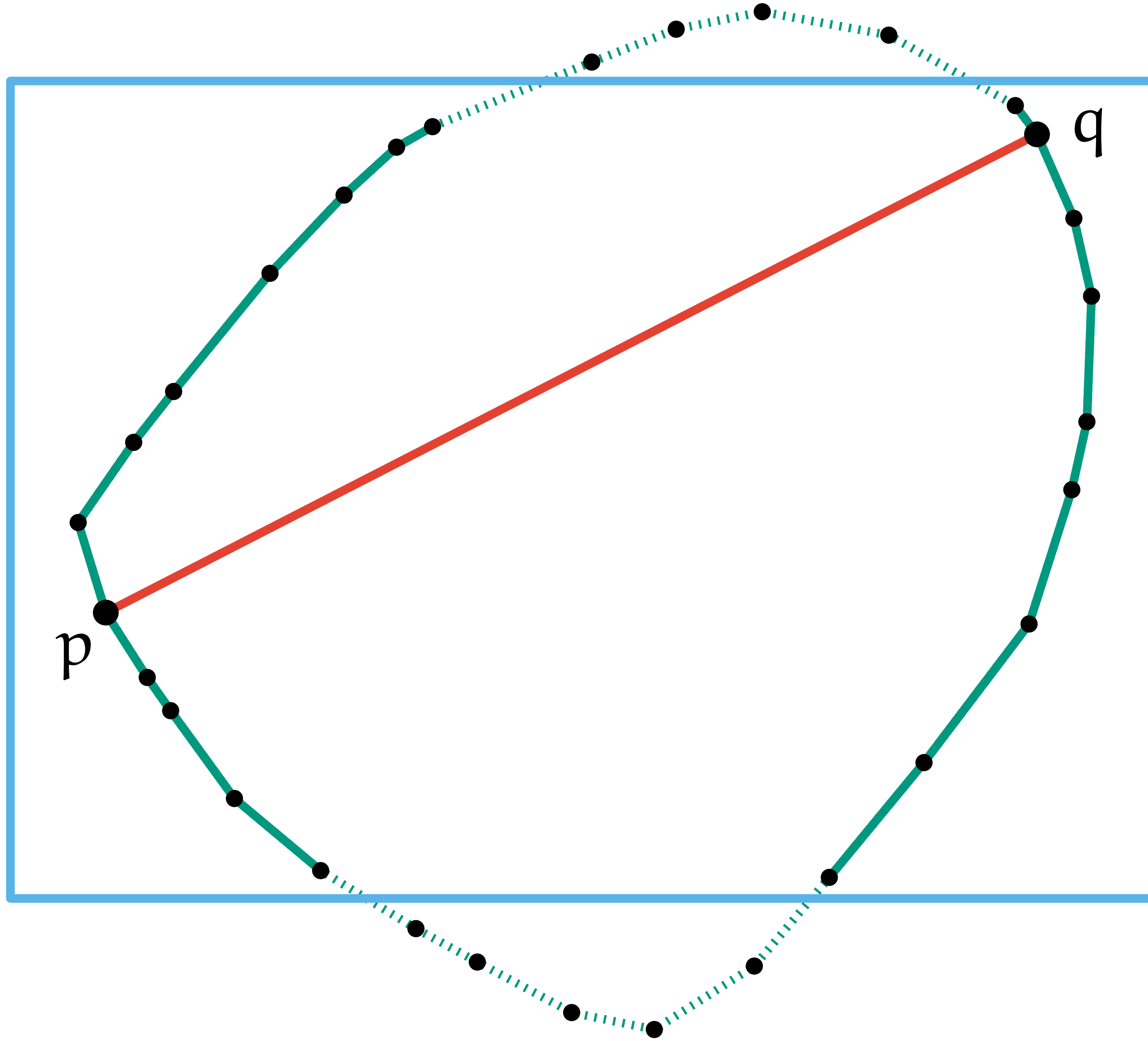
Points in convex position

Consider two sections



Points in convex position

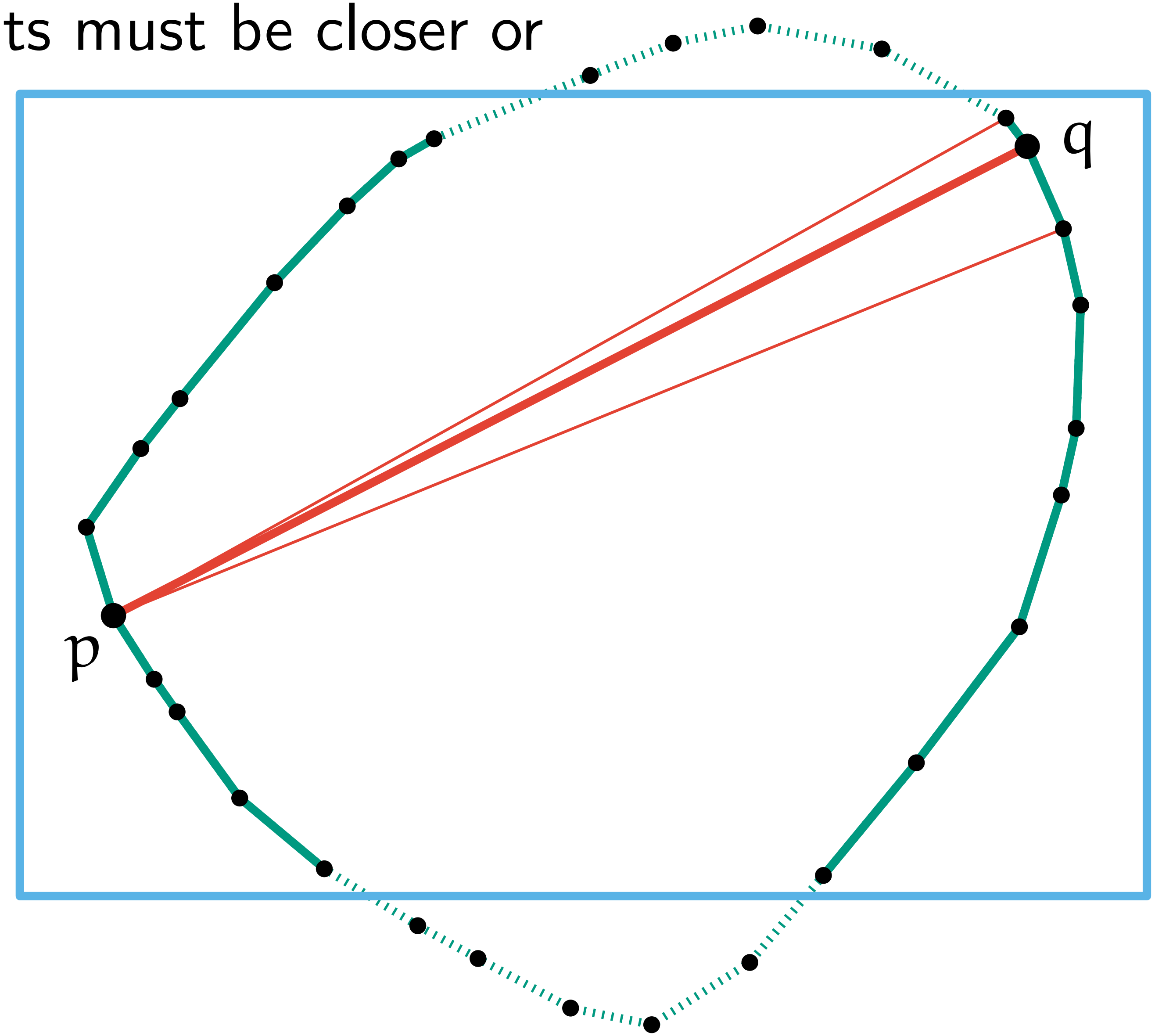
Consider two sections and their pair of furthest points



Points in convex position

Consider two sections and their pair of furthest points

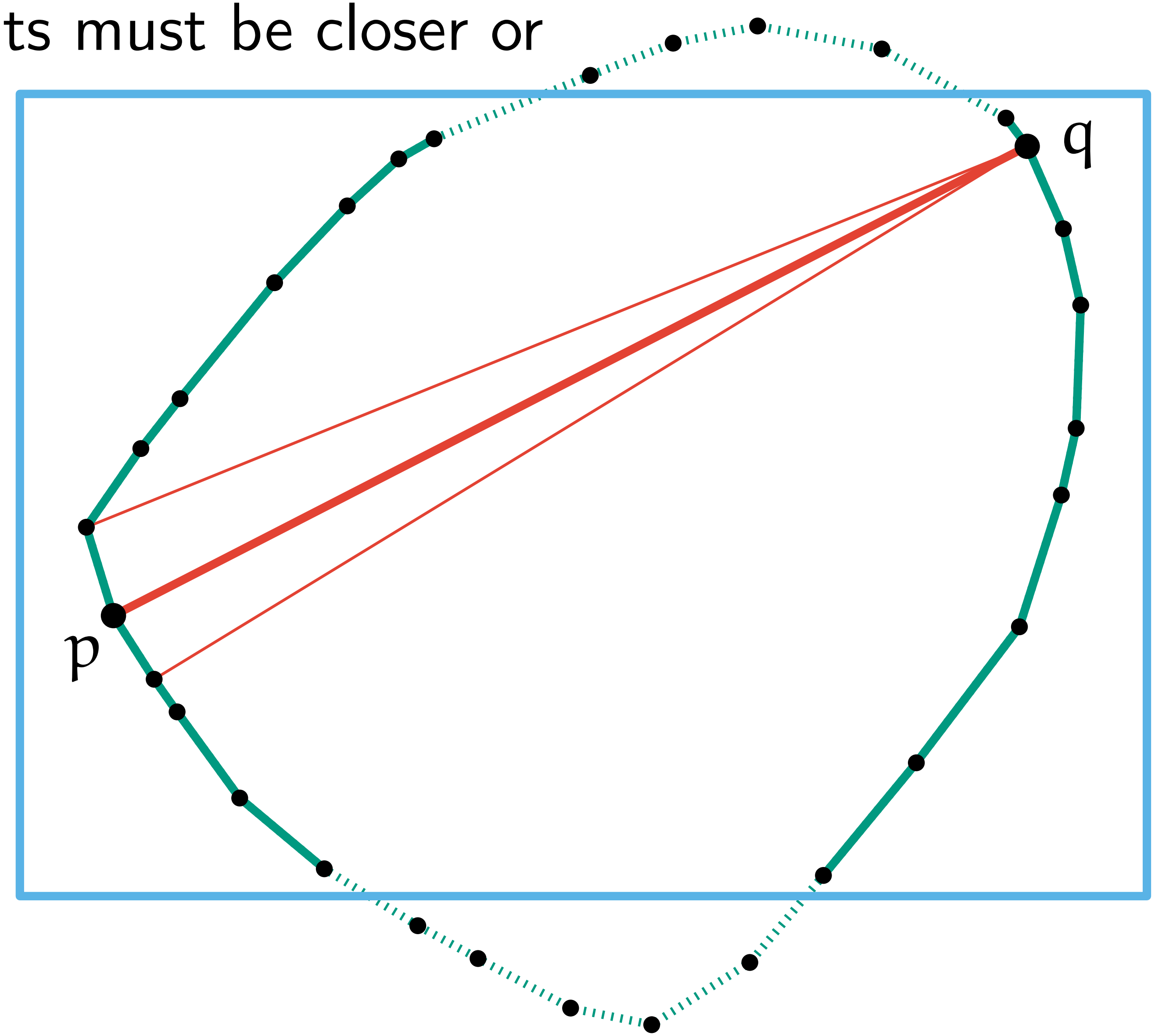
Neighbouring points must be closer or out of range



Points in convex position

Consider two sections and their pair of furthest points

Neighbouring points must be closer or out of range

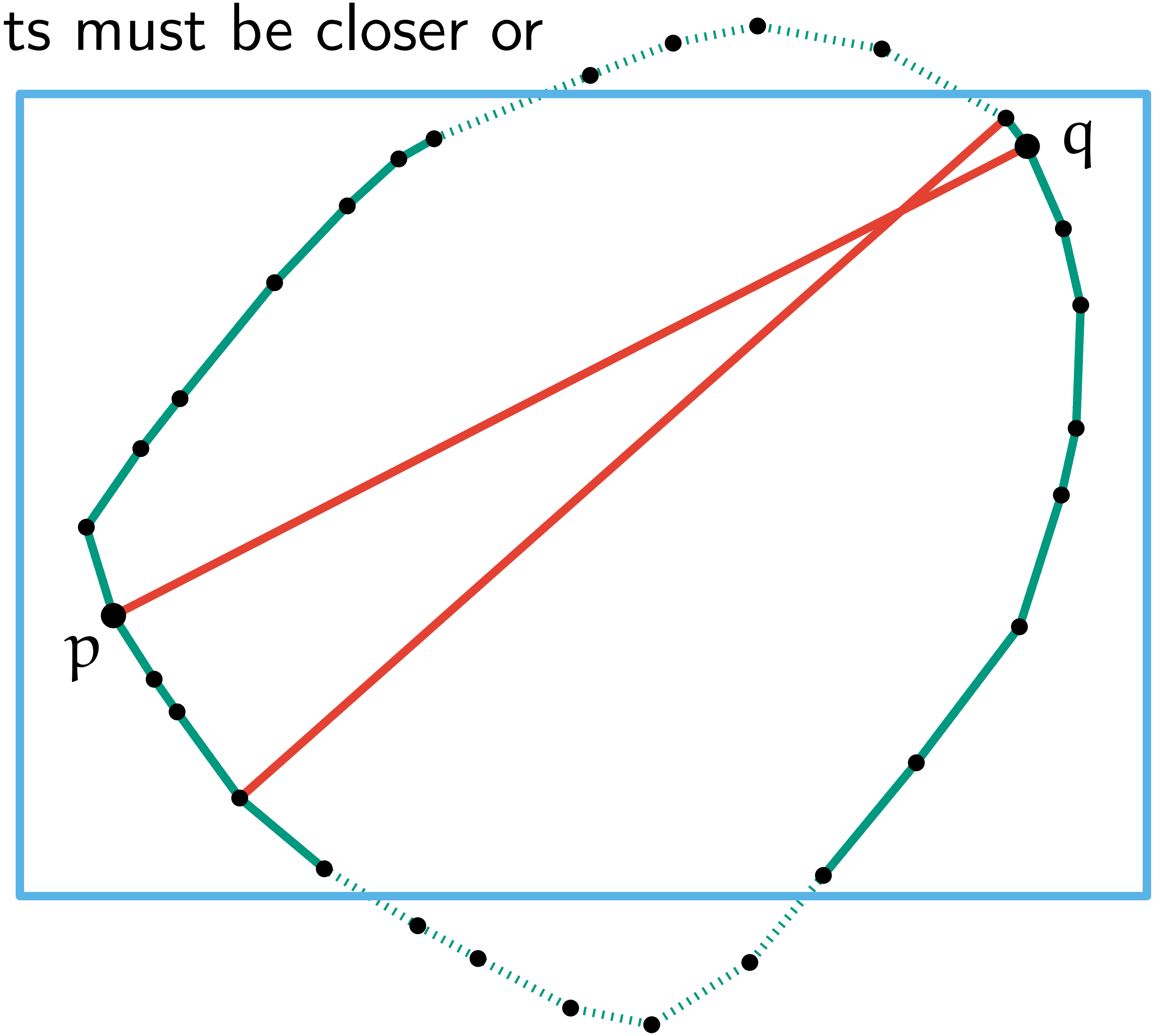


Points in convex position

Consider two sections and their pair of furthest points

Neighbouring points must be closer or out of range

Pairs of *reciprocal local maxima*



Points in convex position

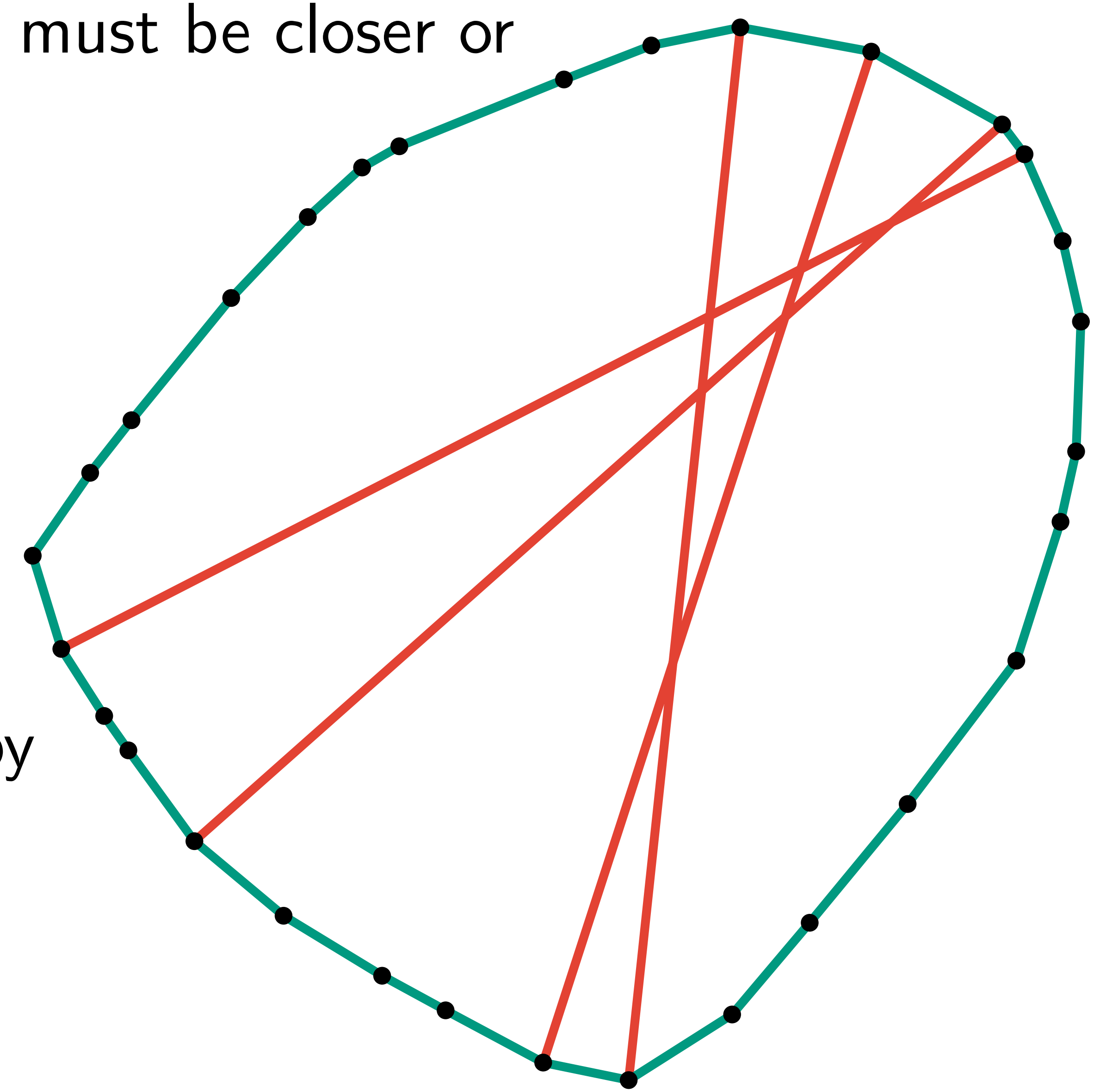
Consider two sections and their pair of furthest points

Neighbouring points must be closer or out of range

Pairs of *reciprocal local maxima*

Lemma: $O(n)$ such pairs in total

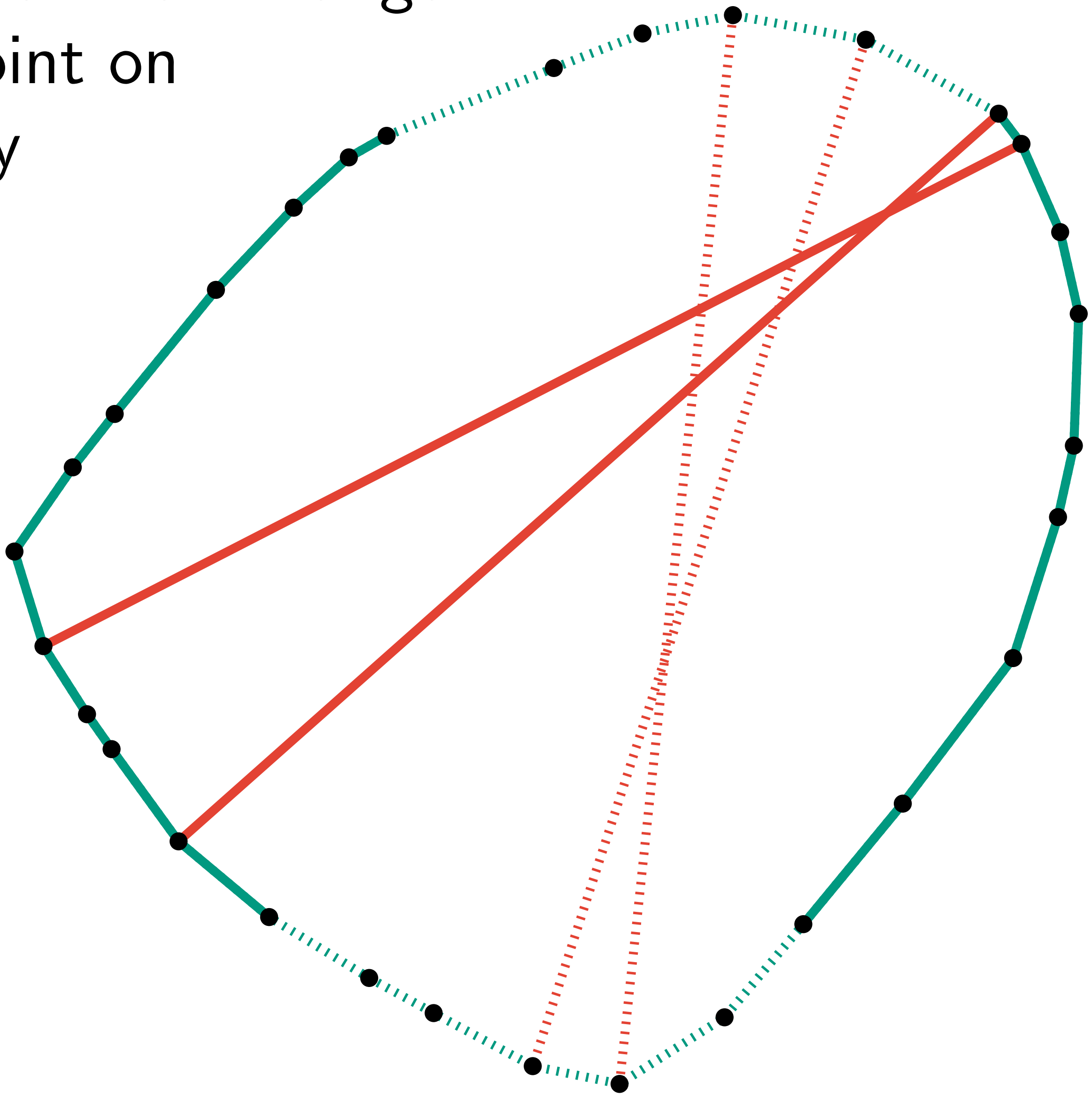
Proof: Intuitively, each pair is visited by the rotating calipers algorithm.



Points in convex position

Answering queries

Diameter = maximum of furthest pairs of
(1) reciprocal local maxima in range
(2) pairs with one point on
section boundary



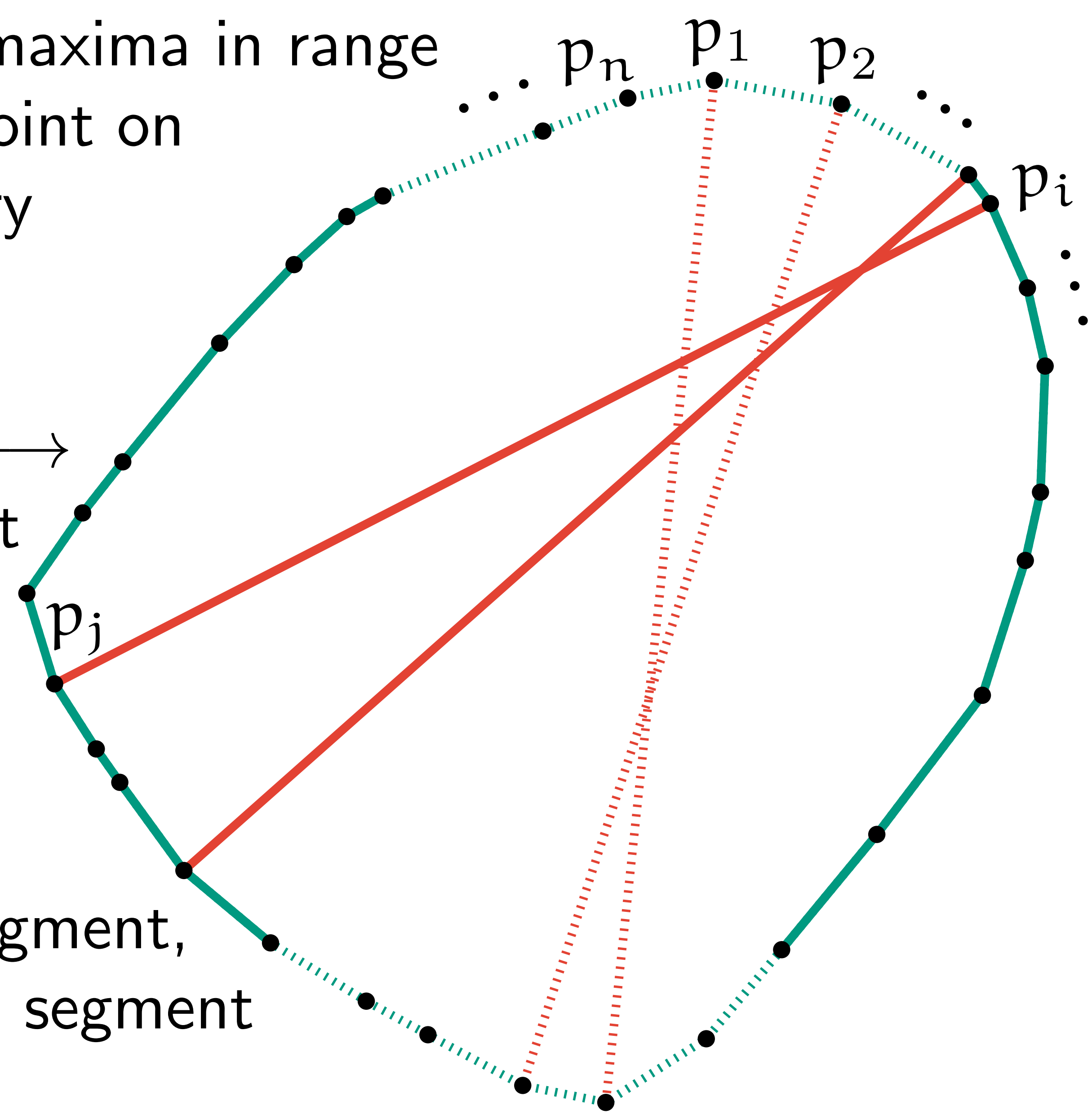
Points in convex position

Answering queries

Diameter = maximum of furthest pairs of
(1) reciprocal local maxima in range
(2) pairs with one point on
section boundary

For (1):

- map pair p_i, p_j $\longrightarrow$ point (i, j), weight $= d(p_i, p_j)$
- 2D range-max data structure
- query:
  x-range = first segment,
  y-range = second segment

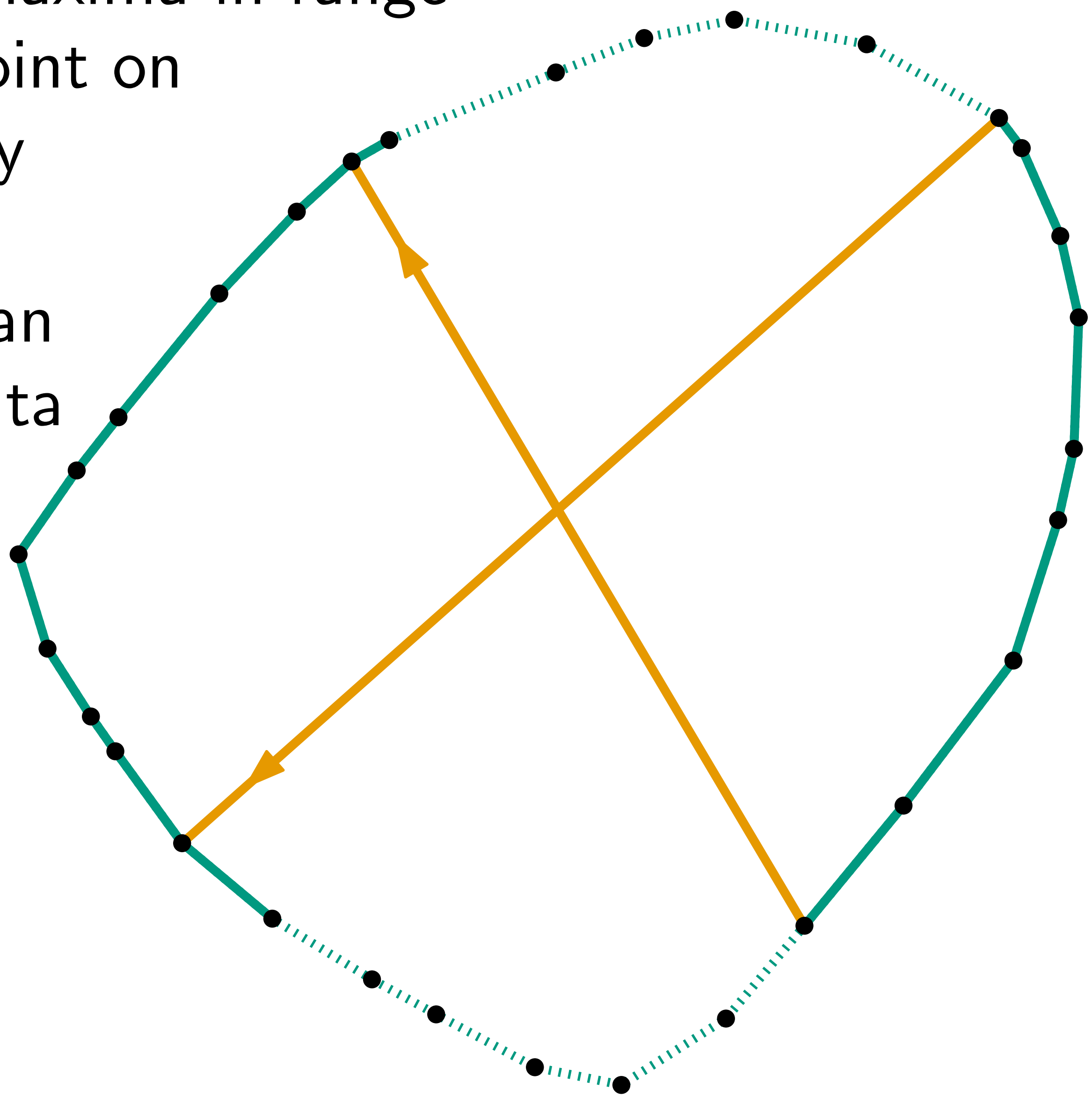


Points in convex position

Answering queries

Diameter = maximum of furthest pairs of
(1) reciprocal local maxima in range
(2) pairs with one point on
section boundary

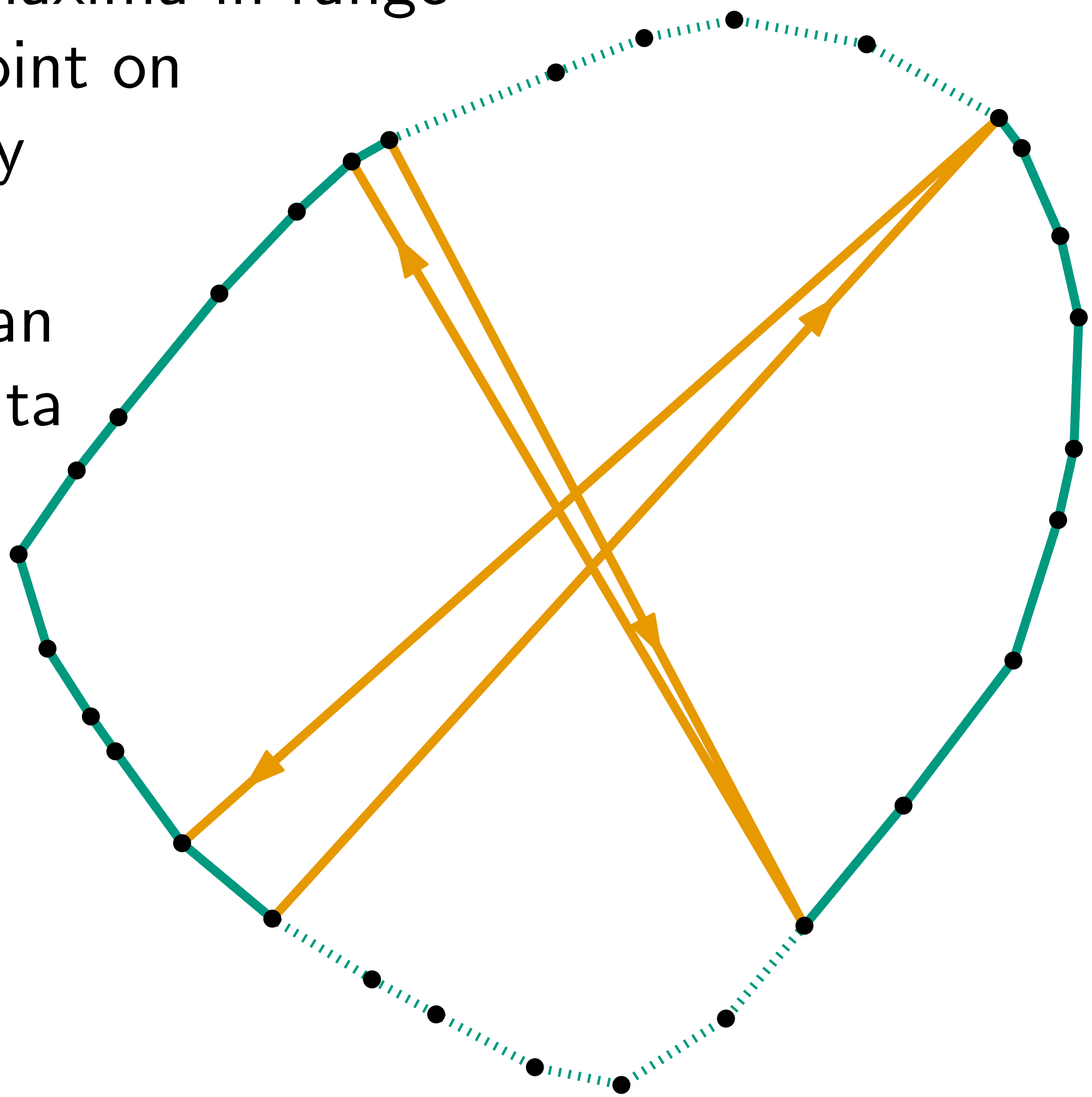
For (2), we develop an $O(n \log n)$-space data structure with query time $O(\log n)$



Points in convex position

Answering queries

Diameter = maximum of furthest pairs of
(1) reciprocal local maxima in range
(2) pairs with one point on
section boundary



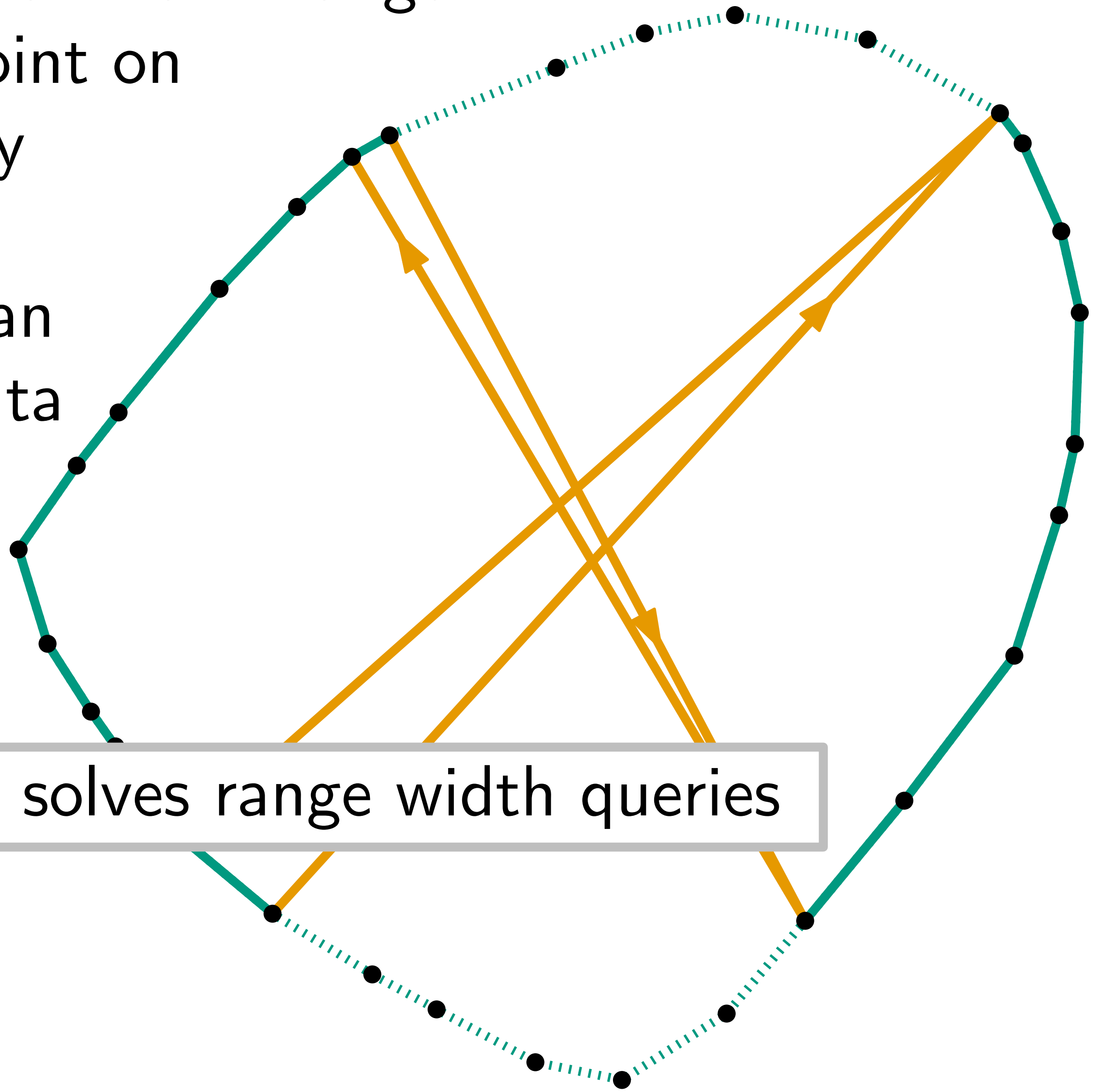
For (2), we develop an $O(n \log n)$-space data structure with query time $O(\log n)$

Points in convex position

Answering queries

Diameter = maximum of furthest pairs of
(1) reciprocal local maxima in range
(2) pairs with one point on section boundary

For (2), we develop an $O(n \log n)$-space data structure with query time $O(\log n)$



Small extension also solves range width queries

Conclusions

- Range diameter is as hard as set intersection
- For two independently preprocessed convex polygons P and Q, computing furthest pair in $P \cup Q$ takes $\tilde{\Omega}(\mathtt{min}(|P|, |Q|))$ time

Conclusions

- Range diameter is as hard as set intersection
- For two independently preprocessed convex polygons P and Q, computing furthest pair in $P \cup Q$ takes $\tilde{\Omega}(\mathtt{min}(|P|, |Q|))$ time
- Number of **reciprocal local maxima** of any convex polygon is linear $\rightarrow$ possibly useful in other contexts
- For points in **convex** position: range **diameter** and **width** queries in $O(n \log n)$ space and $O(\log n)$ query time (or $O(n \log^{\varepsilon} n)$ space, $O(\log^2 n)$ query time)