Subquadratic diameter computation in graphs of bounded VC-dimension

Timothy Chan, Hsien-Chih Chang, Jie Gao, Hung Le,
Sándor Kisfaludi-Bak, Da Wei Zheng

Helsinki Algorithms and Theory Days 2025

Aalto University
School of Science

Super-quadratic distance problems

Given a simple undirected graph G on n vertices and m edges, and the corrsponding shortest-path metric,

- **Diameter**
  Find the maximum distance among vertices
- **Eccentricites**
  For each vertex v find the minimum radius of ball at v that covers G.
- **Distance Oracle**
  Create a data structure, that, when given a pair of points, returns their distance in $\widetilde{O}(1)$ time.

Super-quadratic distance problems

Given a simple undirected graph G on n vertices and m edges, and the corrsponding shortest-path metric,

- **Diameter**
  Find the maximum distance among vertices
- **Eccentricites**
  For each vertex v find the minimum radius of ball at v that covers G.
- **Distance Oracle**
  Create a data structure, that, when given a pair of points, returns their distance in $\widetilde{O}(1)$ time.

Super-quadratic distance problems

Given a simple undirected graph G on n vertices and m edges, and the corrsponding shortest-path metric,

- **Diameter**
  Find the maximum distance among vertices
- **Eccentricites**
  For each vertex v find the minimum radius of ball at v that covers G.
- **Distance Oracle**
  Create a data structure, that, when given a pair of points, returns their distance in $\widetilde{O}(1)$ time.

General: weighted $O(mn)$, unweighted $O(n^\omega \log n)$
Sparse graphs: $O(n^2)$

Super-quadratic distance problems

Given a simple undirected graph G on n vertices and m edges, and the corrsponding shortest-path metric,

- **Diameter**
  Find the maximum distance among vertices
- **Eccentricites**
  For each vertex v find the minimum radius of ball at v that covers G.
- **Distance Oracle**
  Create a data structure, that, when given a pair of points, returns their distance in $\widetilde{O}(1)$ time.

General: weighted $O(mn)$, unweighted $O(n^{\omega} \log n)$
Sparse graphs: $O(n^2)$

Lower bound by Roditty and V. Williams (2013):
No $O(n^{2-\varepsilon})$ algo to decide diameter 2 vs 3 in sparse graphs under SETH.

SotA sub-quadratic diameter computation

First breakthrough:
Cabello '18, Gawrychowski et al. '21: diameter of **planar** graphs in $\widetilde{O}(n^{5/3})$

SotA sub-quadratic diameter computation

First breakthrough:
Cabello '18, Gawrychowski et al. '21: diameter of **planar** graphs in $\widetilde{O}(n^{5/3})$

K_h-minor-free graphs: $\widetilde{O}(n^{2-1/(3h-1)})$ [DHV22, LW24]
Graphs of distance VC-dimension d: $\widetilde{O}(Dmn^{1-1/d})$ [DHV22, DKP24]

graph param., def later

diameter

SotA sub-quadratic diameter computation

First breakthrough:
Cabello '18, Gawrychowski et al. '21: diameter of **planar** graphs in $\widetilde{O}(n^{5/3})$

K_h-minor-free graphs: $\widetilde{O}(n^{2-1/(3h-1)})$ [DHV22, LW24]
Graphs of distance VC-dimension d: $\widetilde{O}(Dmn^{1-1/d})$ [DHV22, DKP24]
graph param., def later
diameter

Axis-parallel unit squares: $O(Dn^{7/4})$ [DKP24]
Unit disk graphs with $+\mathbf{2}$ **additive error**: $\widetilde{O}(n^{2-1/18})$ [CGL24]

SotA sub-quadratic diameter computation

First breakthrough:
Cabello '18, Gawrychowski et al. '21: diameter of **planar** graphs in $\widetilde{O}(n^{5/3})$

K_h-minor-free graphs: $\widetilde{O}(n^{2-1/(3h-1)})$ [DHV22, LW24]
Graphs of distance VC-dimension d: $\widetilde{O}(Dmn^{1-1/d})$ [DHV22, DKP24]
graph param., def later
diameter

Axis-parallel unit squares: $O(Dn^{7/4})$ [DKP24]
Unit disk graphs with $+2$ **additive error**: $\widetilde{O}(n^{2-1/18})$ [CGL24]

$\Omega(n^{2-\varepsilon})$ lower bound for many intersection graphs [B**K**KNP22]:
Unit balls, axis-aligned unit cubes, unit segments, etc.

SotA sub-quadratic diameter computation

First breakthrough:
Cabello '18, Gawrychowski et al. '21: diameter of **planar** graphs in $\widetilde{O}(n^{5/3})$

K_h-minor-free graphs: $\widetilde{O}(n^{2-1/(3h-1)})$ [DHV22, LW24]
Graphs of distance VC-dimension d: $\widetilde{O}(Dmn^{1-1/d})$ [DHV22, DKP24]
graph param., def later
diameter

Axis-parallel unit squares: $O(Dn^{7/4})$ [DKP24]
Unit disk graphs with $+2$ **additive error**: $\widetilde{O}(n^{2-1/18})$ [CGL24]

$\Omega(n^{2-\varepsilon})$ lower bound for many intersection graphs [B**K**KNP22]:
Unit balls, axis-aligned unit cubes, unit segments, etc.

Is there a subquadratic algorithm
for diameter in unit disk graphs?

Results: Subquadratic algorithms for diameter

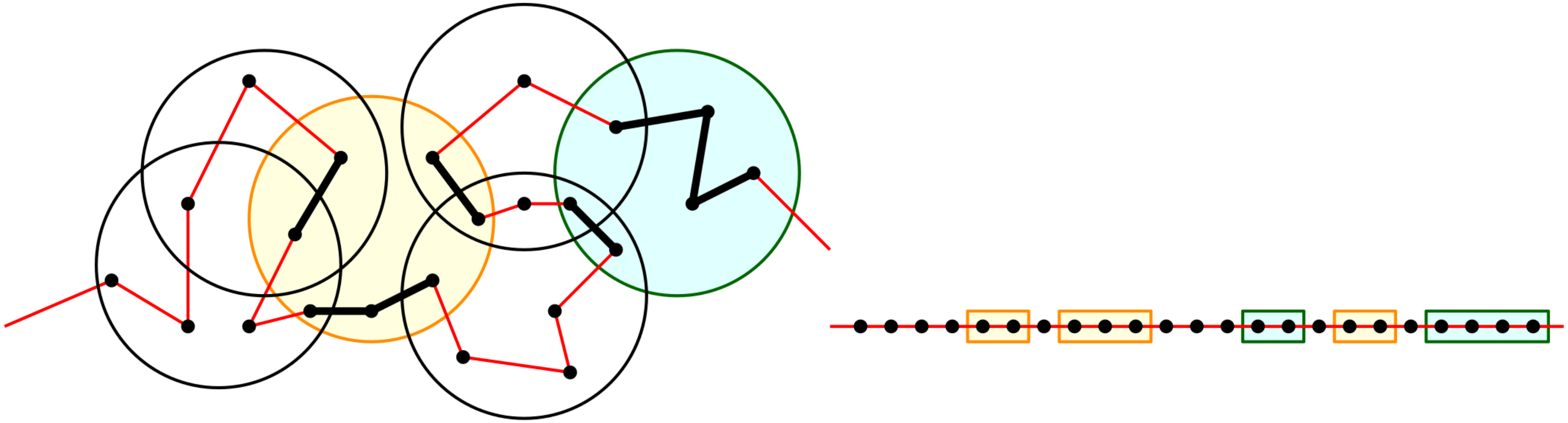
graph class	best previous		new
planar	$\widetilde{O}(n^{5/3})$	[Cab18, GKM+21]	
K_h-minor-free	$\widetilde{O}(n^{2-1/(3h-1)})$	[DHV22, LW24]	$\widetilde{O}(n^{2-1/(2h-2)})$
VC-dim.-bounded	$\widetilde{O}(\min\{Dmn^{1-1/d},\ mn\})$	[DHV22, DKP24]	$\widetilde{O}(mn^{1-1/(2d)})$
unit square	$\widetilde{O}(\min\{Dn^{7/4},\ n^2\})$	[DKP24]	$O^*(n^{2-1/8})$
arbitrary square	$\widetilde{O}(n^2)$	[CS19]	$\widetilde{O}(n^{2-1/12})$
unit disk	$O(n^2\sqrt{\frac{\log\log n}{\log n}})$	[CS16]	$O^*(n^{2-1/18})$

Results: Subquadratic algorithms for diameter

graph class	best previous		new
planar	$\widetilde{O}(n^{5/3})$	[Cab18, GKM+21]	
K_h-minor-free	$\widetilde{O}(n^{2-1/(3h-1)})$	[DHV22, LW24]	$\widetilde{O}(n^{2-1/(2h-2)})$
VC-dim.-bounded	$\widetilde{O}(\min\{Dmn^{1-1/d},\ mn\})$	[DHV22, DKP24]	$\widetilde{O}(mn^{1-1/(2d)})$
unit square	$\widetilde{O}(\min\{Dn^{7/4},\ n^2\})$	[DKP24]	$O^*(n^{2-1/8})$
arbitrary square	$\widetilde{O}(n^2)$	[CS19]	$\widetilde{O}(n^{2-1/12})$
unit disk	$O(n^2\sqrt{\frac{\log\log n}{\log n}})$	[CS16]	$O^*(n^{2-1/18})$

- We don't need separators!
- More efficient ball growing process based on very simple LDD
- New data structures and ideas for geom. intersection graphs

Implicit representations



Good implicit representation via stabbing paths?

Diameter $\leq D$ iff all balls in $\mathcal{N}^D := \{N^D(v) \mid v \in G\}$ cover $V(G)$.
Strategy: grow balls to compute all balls $\mathcal{N} = \bigcup_r \mathcal{N}^r$.

Good implicit representation via stabbing paths?

Diameter $\leq D$ iff all balls in $\mathcal{N}^D := \{N^D(v) \mid v \in G\}$ cover $V(G)$.
Strategy: grow balls to compute all balls $\mathcal{N} = \bigcup_r \mathcal{N}^r$.

Issue: Enumerating all balls takes $O(n^3)$ time. With $O(n)$-time BFS: $O(n^2)$.

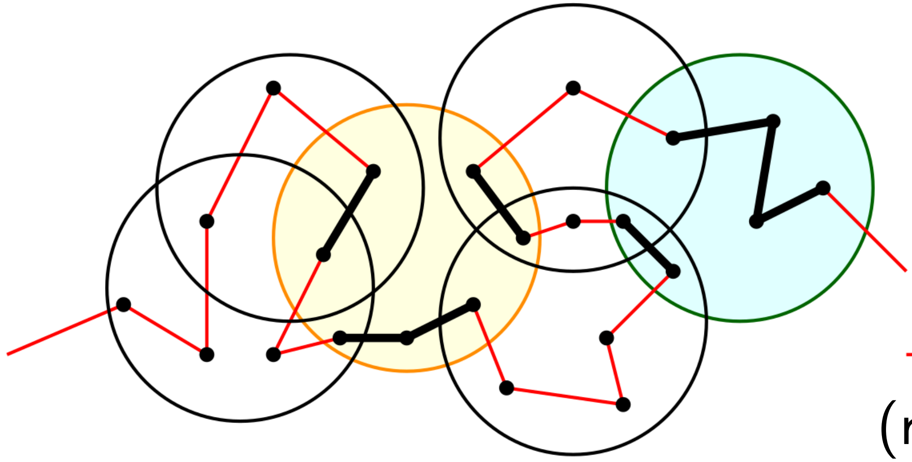
Good implicit representation via stabbing paths?

Diameter $\leq D$ iff all balls in $\mathcal{N}^D := \{N^D(v) \mid v \in G\}$ cover $V(G)$.
Strategy: grow balls to compute all balls $\mathcal{N} = \bigcup_r \mathcal{N}^r$.

Issue: Enumerating all balls takes $O(n^3)$ time. With $O(n)$-time BFS: $O(n^2)$.

We need an *implicit* representation for the balls.

Implicit representation of disks via geometric stabbing path:



(recall segment tree)

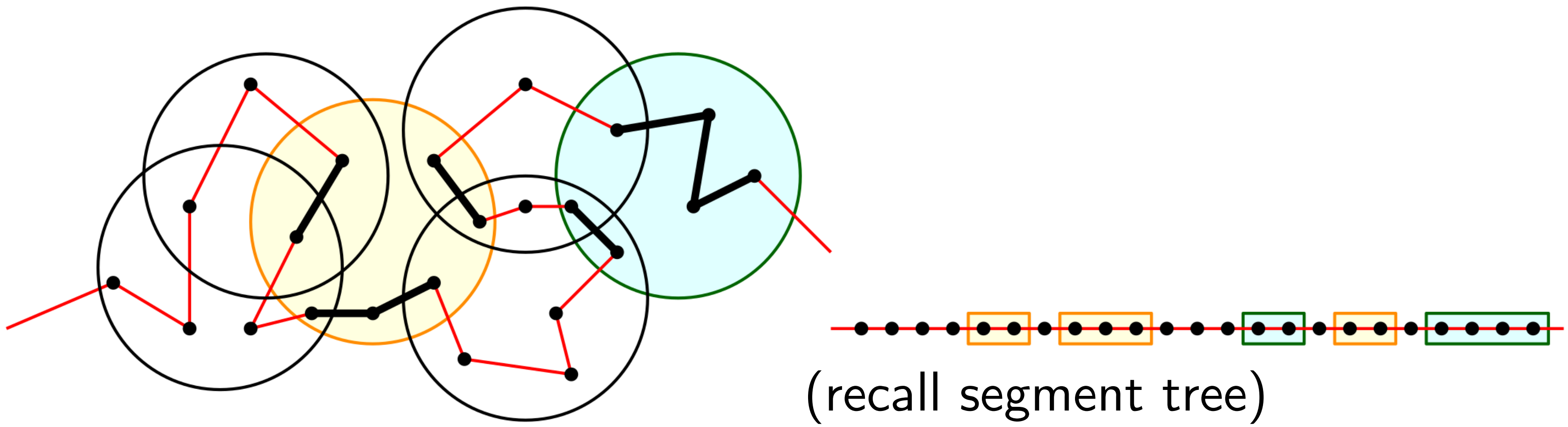
Good implicit representation via stabbing paths?

Diameter $\leq D$ iff all balls in $\mathcal{N}^D := \{N^D(v) \mid v \in G\}$ cover $V(G)$.
Strategy: grow balls to compute all balls $\mathcal{N} = \bigcup_r \mathcal{N}^r$.

Issue: Enumerating all balls takes $O(n^3)$ time. With $O(n)$-time BFS: $O(n^2)$.

We need an *implicit* representation for the balls.

Implicit representation of disks via geometric stabbing path:



A stabbing path of G is any complete ordering λ of $V(G)$.
To represent $X \subset V(G)$, list its elements as intervals of λ (denoted: $\mathrm{Rep}_\lambda(X)$)

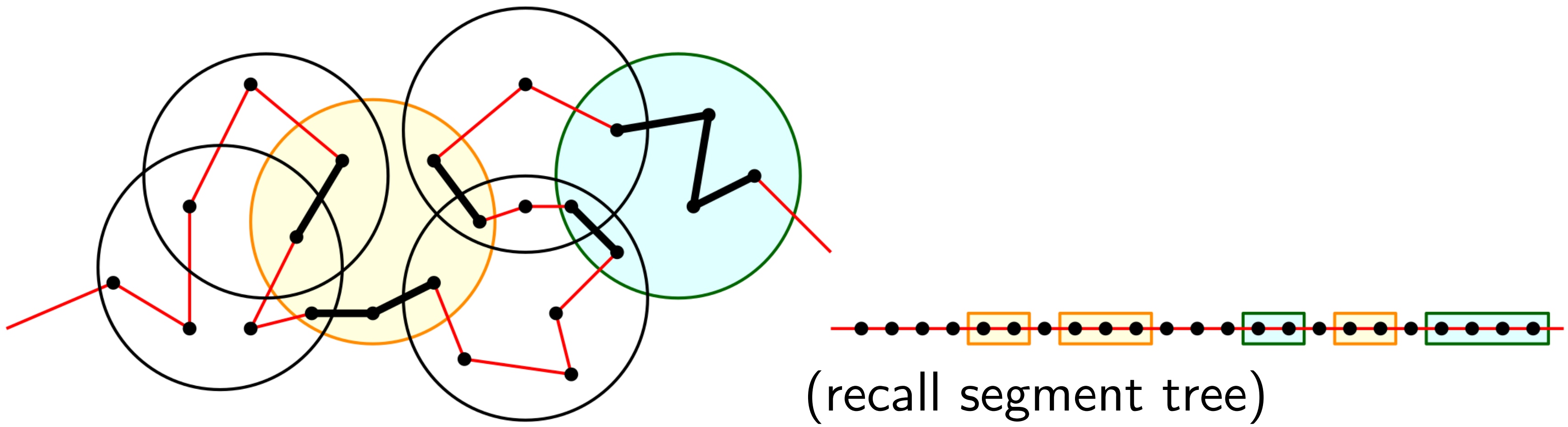
Good implicit representation via stabbing paths?

Diameter $\leq D$ iff all balls in $\mathcal{N}^D := \{N^D(v) \mid v \in G\}$ cover $V(G)$.
Strategy: grow balls to compute all balls $\mathcal{N} = \bigcup_r \mathcal{N}^r$.

Issue: Enumerating all balls takes $O(n^3)$ time. With $O(n)$-time BFS: $O(n^2)$.

We need an *implicit* representation for the balls.

Implicit representation of disks via geometric stabbing path:



(recall segment tree)

A stabbing path of G is any complete ordering λ of $V(G)$.
To represent $X \subset V(G)$, list its elements as intervals of λ (denoted: $\text{Rep}_\lambda(X)$)

We want subquadratic representation in amortized sense:

$$\sum_{X \in \mathcal{N}} |\text{Rep}_\lambda(X)| = O(n^{1.999}).$$

Distance VC dimension in graphs

Given a set family $\mathcal{X}$ over universe V, the set $A \subset V$ is *shattered* by $\mathcal{X}$ if:
for all $S \subset A$ there is $X \in \mathcal{X}$ s.t. $X \cap A = S$.

i.e., $\mathcal{X}$ can carve out every subset of A.

Distance VC dimension in graphs

Given a set family $\mathcal{X}$ over universe V, the set $A \subset V$ is *shattered* by $\mathcal{X}$ if:
for all $S \subset A$ there is $X \in \mathcal{X}$ s.t. $X \cap A = S$.

i.e., $\mathcal{X}$ can carve out every subset of A.

The *VC dimension* of $\mathcal{X}$ is the maximum $|A|$ such that A is shattered by $\mathcal{X}$.

small VC-dimension $\simeq$ samples give a faithful represnetation

Distance VC dimension in graphs

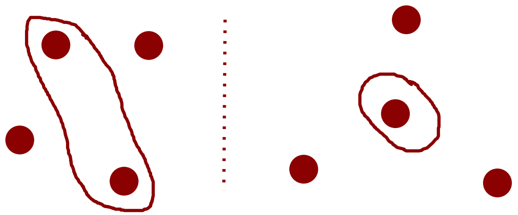
Given a set family $\mathcal{X}$ over universe V, the set $A \subset V$ is *shattered* by $\mathcal{X}$ if:
for all $S \subset A$ there is $X \in \mathcal{X}$ s.t. $X \cap A = S$.

i.e., $\mathcal{X}$ can carve out every subset of A.

The *VC dimension* of $\mathcal{X}$ is the maximum $|A|$ such that A is shattered by $\mathcal{X}$.

small VC-dimension $\simeq$ samples give a faithful represnetation

E.g. $V = \mathbb{R}^2$, $\mathcal{X}$ =halfplanes $\rightarrow$ VC-dim: $d = 3$.



Distance VC dimension in graphs

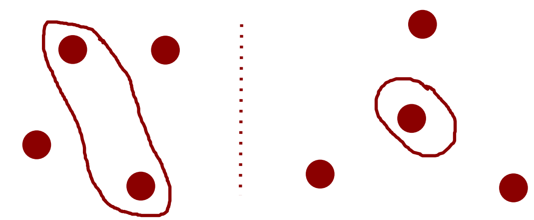
Given a set family $\mathcal{X}$ over universe V, the set $A \subset V$ is *shattered* by $\mathcal{X}$ if: for all $S \subset A$ there is $X \in \mathcal{X}$ s.t. $X \cap A = S$.

i.e., $\mathcal{X}$ can carve out every subset of A.

The *VC dimension* of $\mathcal{X}$ is the maximum $|A|$ such that A is shattered by $\mathcal{X}$.

small VC-dimension $\simeq$ samples give a faithful represnetation

E.g. $V = \mathbb{R}^2$, $\mathcal{X}$ =halfplanes $\rightarrow$ VC-dim: $d = 3$.



$G :=$ undirected unweighted graph, $N^r(v) =$ ball of radius r at $v \in V(G)$.

The *distance VC-dimension* d *of* G is the VC dimension of $\mathcal{N} := \bigcup_r \mathcal{N}^r(v)$.

Stabbing paths and VC-dimension

Lemma (Informal)
Given* a set system $(V, \mathcal{N})$ of VC-dimension** d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high*** probability:

$$\sum_{X \in \mathcal{N}} |\mathsf{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

Stabbing paths and VC-dimension

Lemma (Informal)
Given* a set system $(V, \mathcal{N})$ of VC-dimension** d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high*** probability:

$$\sum_{X \in \mathcal{N}} |\mathsf{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

For implicit representations, we need:

1. VC-dimension bound
2. Small ball system $\widehat{\mathcal{N}}$ of size $O(n^{1+1/d-0.01})$

Stabbing paths and VC-dimension

Lemma (Informal)

Given* a set system $(V, \mathcal{N})$ of VC-dimension** d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high*** probability:

$$\sum_{X \in \mathcal{N}} |\mathsf{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

For implicit representations, we need:

1. VC-dimension bound
2. Small ball system $\widehat{\mathcal{N}}$ of size $O(n^{1+1/d-0.01})$

Theorem [KZ25]

K_h-minor free graphs have distance VC dimension at most $h-1$.

Stabbing paths and VC-dimension

Lemma (Informal)

Given⋆ a set system $(V, \mathcal{N})$ of VC-dimension⋆⋆ d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high⋆⋆⋆ probability:

$$\sum_{X \in \mathcal{N}} |\mathsf{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

For implicit representations, we need:

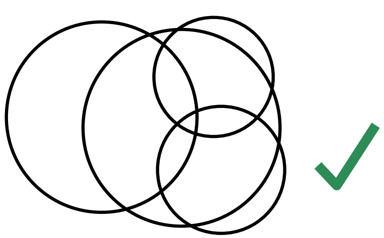
1. VC-dimension bound
2. Small ball system $\widehat{\mathcal{N}}$ of size $O(n^{1+1/d-0.01})$

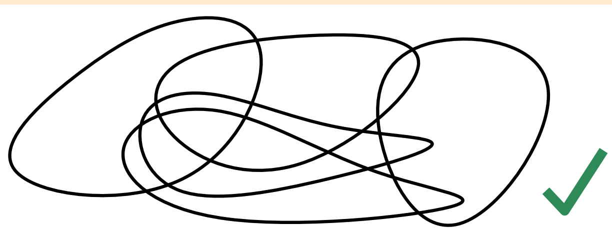
Theorem [KZ25]

K_h-minor free graphs have distance VC dimension at most $h-1$.

Theorem (adapted from [CGL24])

Intersection graphs of pseudo-disks have distance VC dimension at most 4.





Stabbing paths and VC-dimension

Lemma (Informal)

Given* a set system $(V, \mathcal{N})$ of VC-dimension** d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high*** probability:

$$\sum_{X \in \mathcal{N}} |\mathrm{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

For implicit representations, we need:

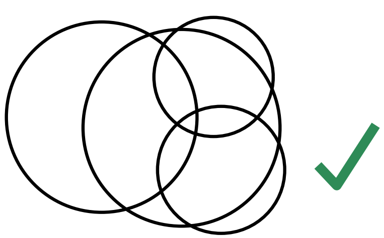
1. VC-dimension bound
2. Small ball system $\widehat{\mathcal{N}}$ of size $O(n^{1+1/d-0.01})$

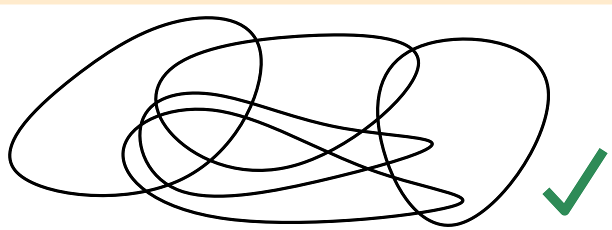
Theorem [KZ25]

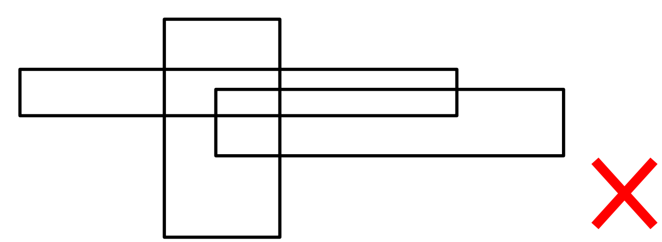
K_h-minor free graphs have distance VC dimension at most $h-1$.

Theorem (adapted from [CGL24])

Intersection graphs of pseudo-disks have distance VC dimension at most 4.







Stabbing paths and VC-dimension

Lemma (Informal)

Given* a set system $(V, \mathcal{N})$ of VC-dimension** d we can construct a stabbing path λ in $\widetilde{O}(n^{1+1/d})$ time such that with high*** probability:

$$\sum_{X \in \mathcal{N}} |\mathsf{Rep}_\lambda(S)| = \widetilde{O}(|\mathcal{N}| n^{1-1/d})$$

Constructed via recursive sampling.

For implicit representations, we need:

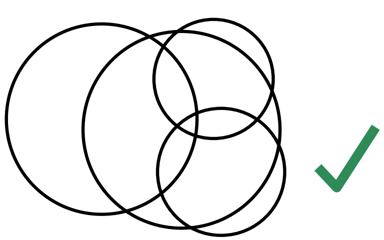
1. VC-dimension bound ✓
2. Small ball system $\widehat{\mathcal{N}}$ of size $O(n^{1+1/d-0.01})$ Coming up next

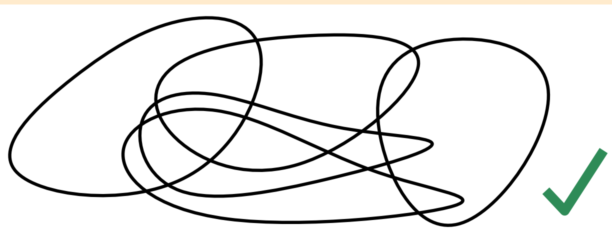
Theorem [KZ25]

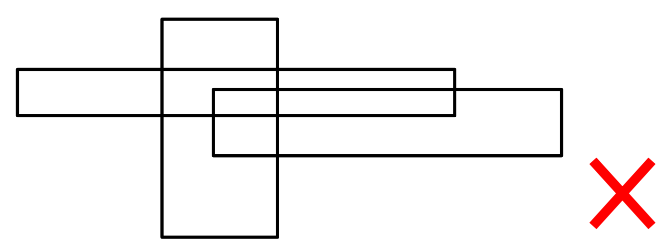
K_h-minor free graphs have distance VC dimension at most $h-1$.

Theorem (adapted from [CGL24])

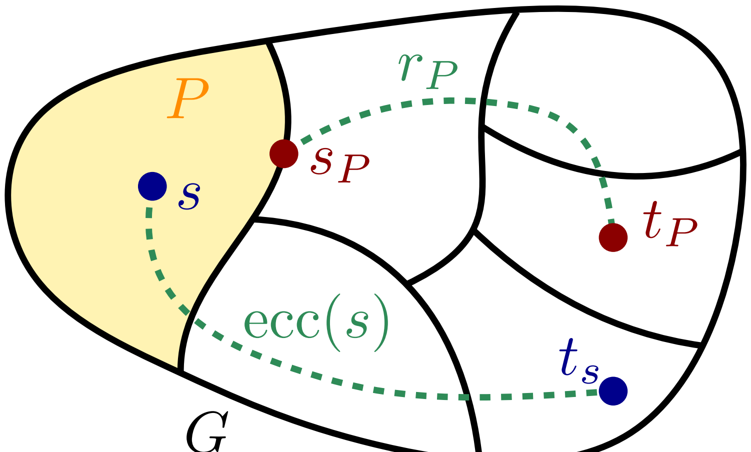
Intersection graphs of pseudo-disks have distance VC dimension at most 4.







Generic framework and data structures



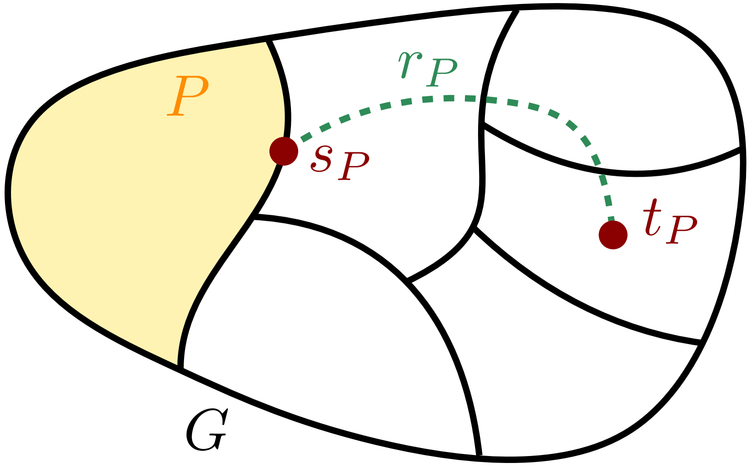
Generic framework: LDD + ball growing

Step 1. Fix Δ, decompose G into $\widetilde{O}(n/\Delta)$ pieces P of (strong) diameter $\leq \Delta$ s.t. total boundary is small: $\sum_P |\partial P| = \widetilde{O}(n/\Delta)$.

Generic framework: LDD + ball growing

Step 1. Fix Δ, decompose G into $\widetilde{O}(n/\Delta)$ pieces P of (strong) diameter $\leq \Delta$ s.t. total boundary is small: $\sum_P |\partial P| = \widetilde{O}(n/\Delta)$.

Step 2. Run BFS from each boundary vertex $v \in \bigcup_P \partial P$, obtaining also $\mathrm{ecc}(v)$. Set $r_P := \max_{v \in \partial P} \mathrm{ecc}(v)$.

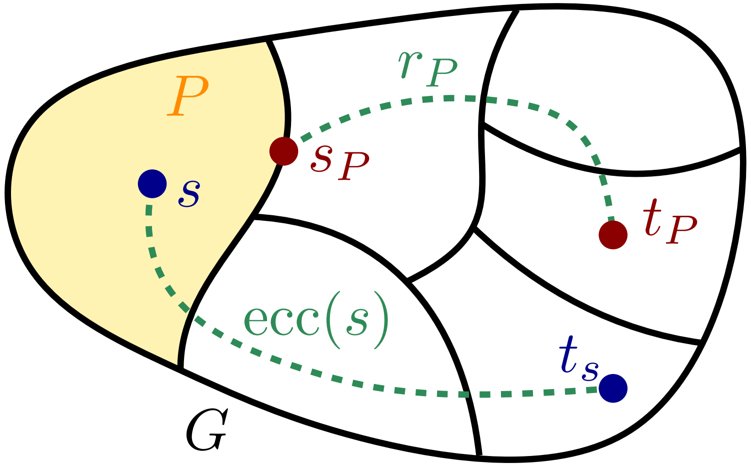


Generic framework: LDD + ball growing

<u>Step 1.</u> Fix Δ, decompose G into $\widetilde{O}(n/\Delta)$ pieces P of (strong) diameter $\leq \Delta$ s.t. total boundary is small: $\sum_P |\partial P| = \widetilde{O}(n/\Delta)$.

<u>Step 2.</u> Run BFS from each boundary vertex $v \in \bigcup_P \partial P$, obtaining also $\mathrm{ecc}(v)$. Set $r_P := \max_{v \in \partial P} \mathrm{ecc}(v)$.

By triangle-ineq:
$\mathrm{dist}_G(s_P, t) - \Delta \leq \mathrm{dist}_G(s, t) \leq \mathrm{dist}_G(s_P, t) + \Delta$



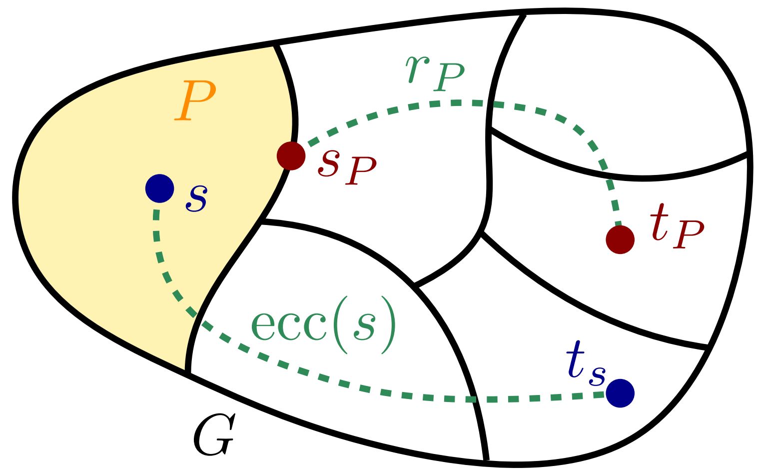
Generic framework: LDD + ball growing

Step 1. Fix Δ, decompose G into $\widetilde{O}(n/\Delta)$ pieces P of (strong) diameter $\leq \Delta$ s.t. total boundary is small: $\sum_P |\partial P| = \widetilde{O}(n/\Delta)$.

Step 2. Run BFS from each boundary vertex $v \in \bigcup_P \partial P$, obtaining also $\mathrm{ecc}(v)$. Set $r_P := \max_{v \in \partial P} \mathrm{ecc}(v)$.

By triangle-ineq:
$\mathrm{dist}_G(s_P, t) - \Delta \leq \mathrm{dist}_G(s, t) \leq \mathrm{dist}_G(s_P, t) + \Delta$

$$\Rightarrow \mathrm{dist}(s_P, t_s) \geq r_P - 2\Delta$$



The *relevant region of P* is

$$R_P := \{v \in G \mid \mathrm{dist}_G(s_P, v) \geq r_P - 2\Delta\}.$$

The *modified r-ball* of $s \in P$ is $\widehat{N}^r(s) := N^r(s) \cap R_P$. Let $\widehat{\mathcal{N}} := \{\widehat{N}^r(s)\}_{r,s}$.

Generic framework: LDD + ball growing

Step 1. Fix Δ, decompose G into $\widetilde{O}(n/\Delta)$ pieces P of (strong) diameter $\leq \Delta$ s.t. total boundary is small: $\sum_P |\partial P| = \widetilde{O}(n/\Delta)$.

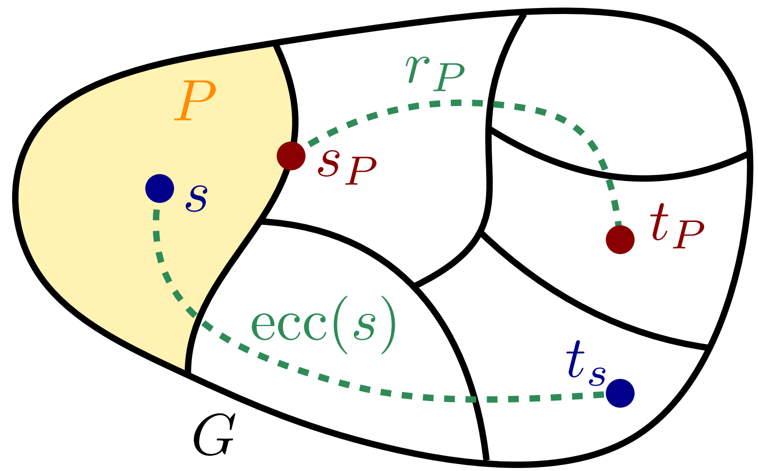
Step 2. Run BFS from each boundary vertex $v \in \bigcup_P \partial P$, obtaining also $\mathrm{ecc}(v)$. Set $r_P := \max_{v \in \partial P} \mathrm{ecc}(v)$.

By triangle-ineq:
$\mathrm{dist}_G(s_P, t) - \Delta \leq \mathrm{dist}_G(s, t) \leq \mathrm{dist}_G(s_P, t) + \Delta$

$$\Rightarrow \mathrm{dist}(s_P, t_s) \geq r_P - 2\Delta$$

The *relevant region of P* is

$$R_P := \{v \in G \mid \mathrm{dist}_G(s_P, v) \geq r_P - 2\Delta\}.$$



The *modified r-ball* of $s \in P$ is $\widehat{N}^r(s) := N^r(s) \cap R_P$. Let $\widehat{\mathcal{N}} := \{\widehat{N}^r(s)\}_{r,s}$.

Lemma

The modified ball system $\widehat{\mathcal{N}}$ has the same VC-dim as $\mathcal{N}$ and $|\widehat{\mathcal{N}}| = O(\Delta n)$

Step 3. Compute stabbing path λ for $\widehat{\mathcal{N}}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$, compute $\mathrm{Rep}_\lambda(\widehat{N}^r(s))$.

Efficient ball growing?

Step 3. Compute stabbing path λ for $\hat{N}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$, compute $\text{Rep}_\lambda(\widehat{N}^r(s))$.

Goal: Given $\text{Rep}_\lambda(\widehat{N}^r(v))$ for all $v \in V(G)$, compute $\text{Rep}_\lambda(\widehat{N}^{r+1}(s))$

Efficient ball growing?

Step 3. Compute stabbing path λ for $\hat{N}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$,
compute $\mathsf{Rep}_\lambda(\widehat{N}^r(s))$.

Goal: Given $\mathsf{Rep}_\lambda(\widehat{N}^r(v))$ for all $v \in V(G)$, compute $\mathsf{Rep}_\lambda(\widehat{N}^{r+1}(s))$

$$\mathsf{Rep}_\lambda(\widehat{N}^{r+1}(s)) = \bigcup_{v \in N(s)} \mathsf{Rep}_\lambda(\widehat{N}^r(v))$$

Efficient ball growing?

Step 3. Compute stabbing path λ for $\hat{N}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$, compute $\mathsf{Rep}_\lambda(\widehat{N}^r(s))$.

Goal: Given $\mathsf{Rep}_\lambda(\widehat{N}^r(v))$ for all $v \in V(G)$, compute $\mathsf{Rep}_\lambda(\widehat{N}^{r+1}(s))$

$$\mathsf{Rep}_\lambda(\widehat{N}^{r+1}(s)) = \bigcup_{v \in N(s)} \mathsf{Rep}_\lambda(\widehat{N}^r(v))$$

Can be done directly in sparse graphs as $|N(s)|$ is (amortized) small

Efficient ball growing?

Step 3. Compute stabbing path λ for $\hat{N}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$, compute $\text{Rep}_\lambda(\widehat{N}^r(s))$.

Goal: Given $\text{Rep}_\lambda(\widehat{N}^r(v))$ for all $v \in V(G)$, compute $\text{Rep}_\lambda(\widehat{N}^{r+1}(s))$

$$\text{Rep}_\lambda(\widehat{N}^{r+1}(s)) = \bigcup_{v \in N(s)} \text{Rep}_\lambda(\widehat{N}^r(v))$$

Can be done directly in sparse graphs as $|N(s)|$ is (amortized) small

In dense graphs, we need to solve:

Interval Searching
Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1 : n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

Efficient ball growing?

Step 3. Compute stabbing path λ for $\hat{N}$.
For each P and each $s \in P$, from $r = r_P - 3\Delta - 1$ to $r = r_P + \Delta$,
compute $\text{Rep}_\lambda(\widehat{N}^r(s))$.

Goal: Given $\text{Rep}_\lambda(\widehat{N}^r(v))$ for all $v \in V(G)$, compute $\text{Rep}_\lambda(\widehat{N}^{r+1}(s))$

$$\text{Rep}_\lambda(\widehat{N}^{r+1}(s)) = \bigcup_{v \in N(s)} \text{Rep}_\lambda(\widehat{N}^r(v))$$

Can be done directly in sparse graphs
as $|N(s)|$ is (amortized) small

In dense graphs, we need to solve:

Interval Searching
Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1 : n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

We need*: $\widetilde{O}$(input size) preprocessing and $\widetilde{O}$(output size) query time

Data strucutre problem reductions

DSP1: Interval Searching

Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1 : n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

Data strucutre problem reductions

DSP1: Interval Searching
Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1 : n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

↓ Reduction with polylog preprocessing and query overhead

DSP2: Interval Cover
Given objects $\mathcal{O}$ and $I : \mathcal{O} \to [1 : n]$, design a data structure that:
for query $q \in \mathcal{O}$ decides if the union of $\{I(o) \mid o \in \mathcal{O} \text{ intersects } q\}$ *covers* $I(q)$.

Hard and non-decomposable range searching...

Data strucutre problem reductions

DSP1: Interval Searching

Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1:n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

↓ Reduction with polylog preprocessing and query overhead

DSP2: Interval Cover

Given objects $\mathcal{O}$ and $I : \mathcal{O} \to [1:n]$, design a data structure that:
for query $q \in \mathcal{O}$ decides if the union of $\{I(o) \mid o \in \mathcal{O} \text{ intersects } q\}$ *covers* $I(q)$.

Hard and non-decomposable range searching...

↓ $\sim$ Reduction with $n^{o(1)}$ preprocessing and query overhead
Slice into blocks of size b.

DSP3: Rainbow colored intersection searching

Given objects $\mathcal{O}$ and $C : \mathcal{O} \to [n]$ (color), design a data structure that:
for query $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.

Data strucutre problem reductions

DSP1: Interval Searching

Given set of objects $\mathcal{O}$, each associated with some integer intervals of $[1:n]$, design a data structure that:
for query $q \in \mathcal{O}$ returns the union of the representations of the objects $o \in \mathcal{O}$ that intersect q.

↓ Reduction with polylog preprocessing and query overhead

DSP2: Interval Cover

Given objects $\mathcal{O}$ and $I : \mathcal{O} \to [1:n]$, design a data structure that:
for query $q \in \mathcal{O}$ decides if the union of $\{I(o) \mid o \in \mathcal{O} \text{ intersects } q\}$ *covers* $I(q)$.

Hard and non-decomposable range searching...

↓ $\sim$ Reduction with $n^{o(1)}$ preprocessing and query overhead
Slice into blocks of size b.

DSP3: Rainbow colored intersection searching

Given objects $\mathcal{O}$ and $C : \mathcal{O} \to [n]$ (color), design a data structure that:
for query $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.

Theorem

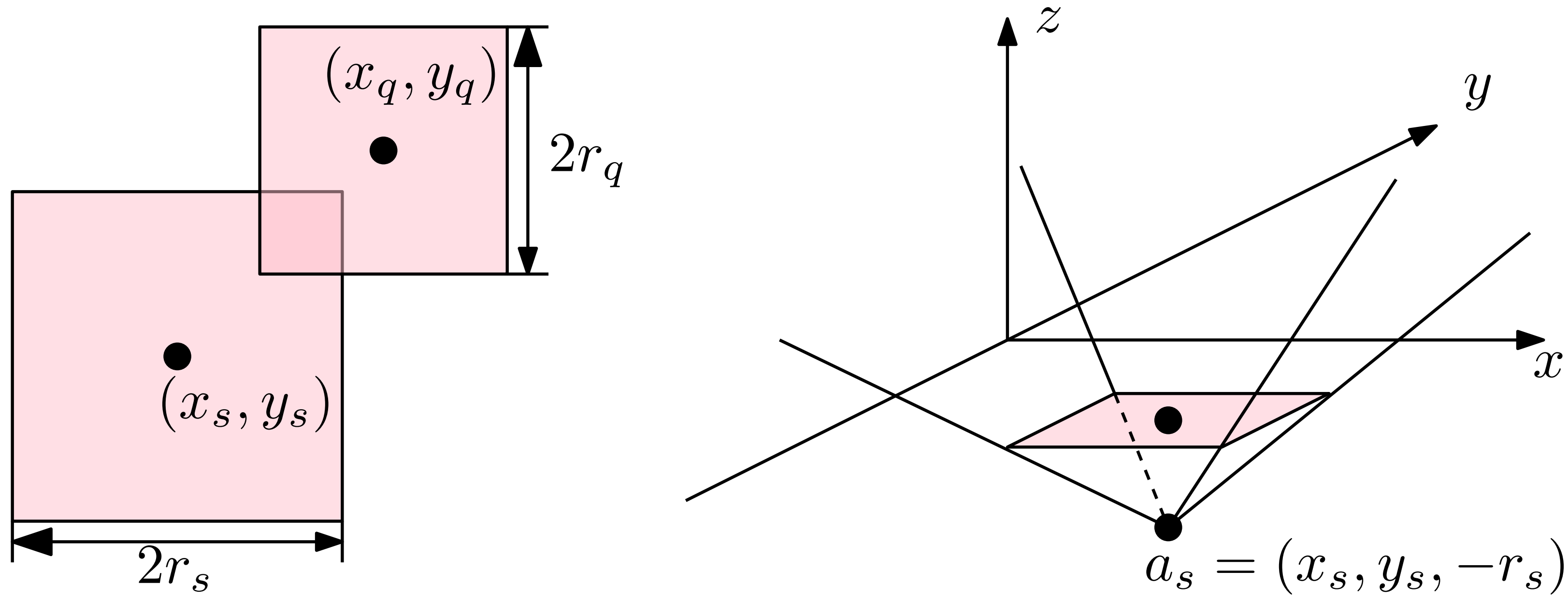
If we can construct in $\widetilde{O}(|\mathcal{O}_{RC}|)$ time a data structure $\mathcal{D}_{RC}$ with $\widetilde{O}(1)$ query for DSP3, then for any $b \in [1, n]$, we can construct a data structure for DSP2 with total run time $\widetilde{O}(N_{IC} \cdot b + L_{IC}/b)$.

Data structure for squares

Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.

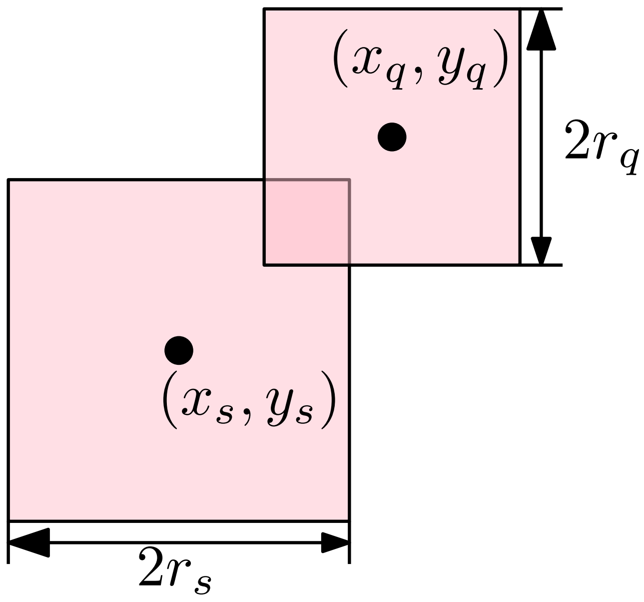
Data structure for squares

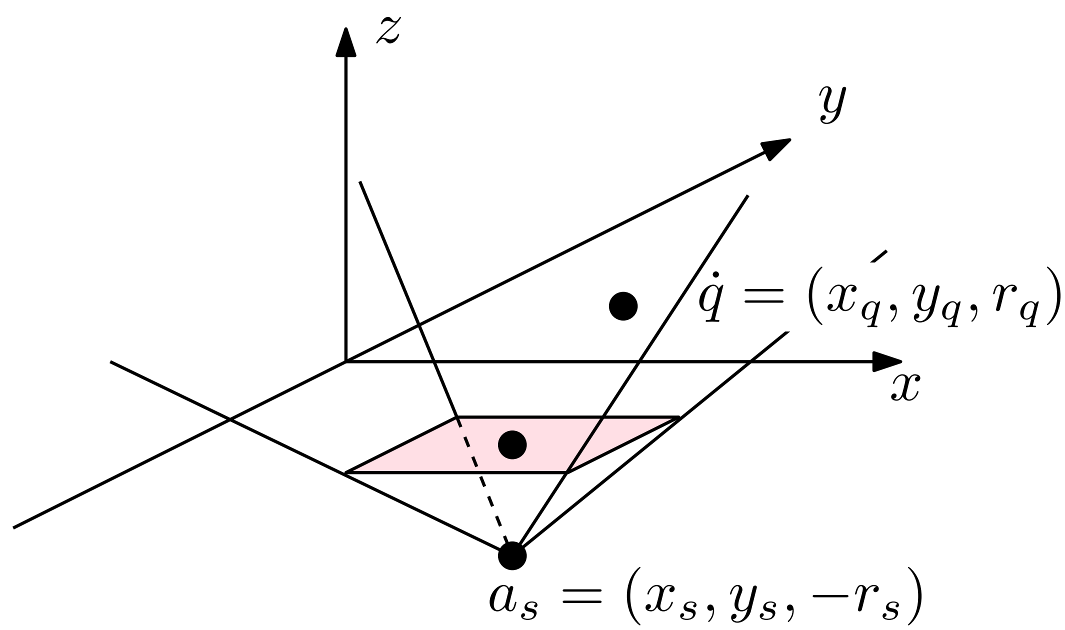
Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.



Data structure for squares

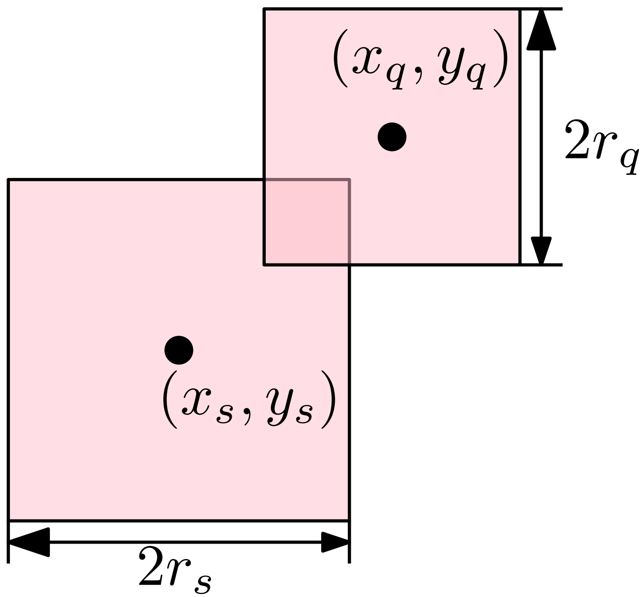
Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.

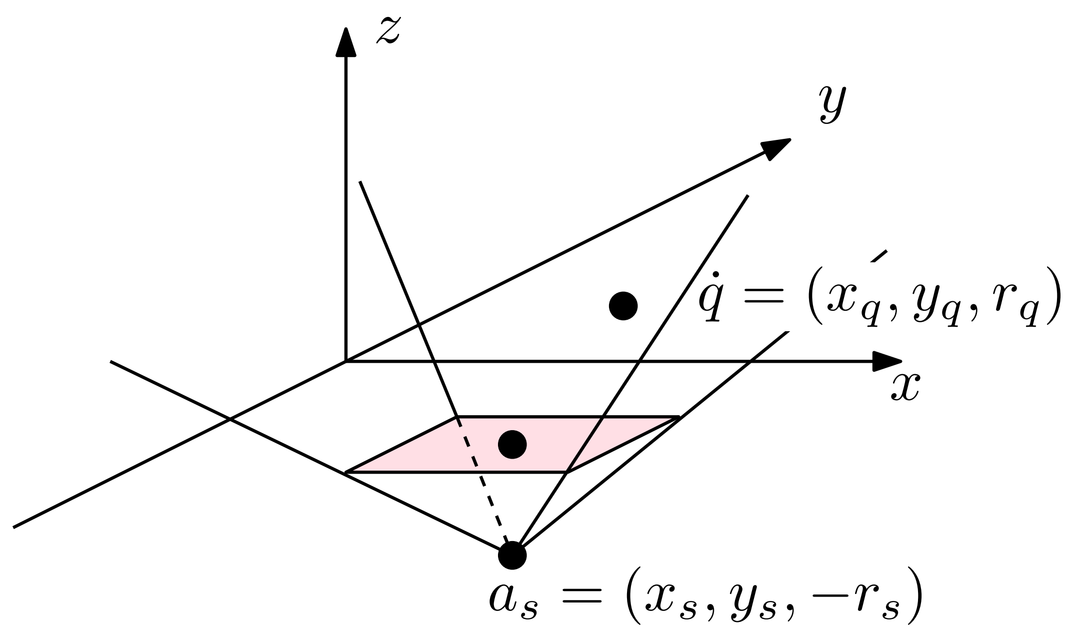




Data structure for squares

Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O} \text{ intersects } q\}$.

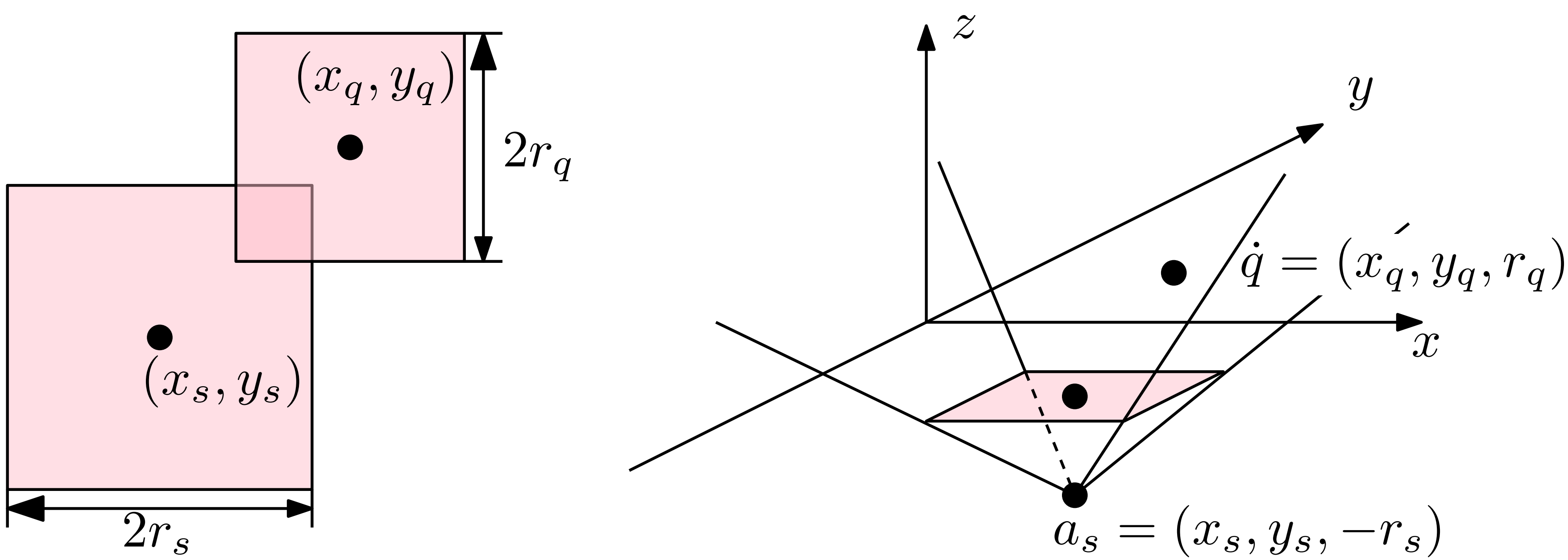




Rough idea:
- q has intersection with color class i iff $\dot{q}$ is above lower envelope of color-i cones.

Data structure for squares

Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O}$ intersects $q\}$.

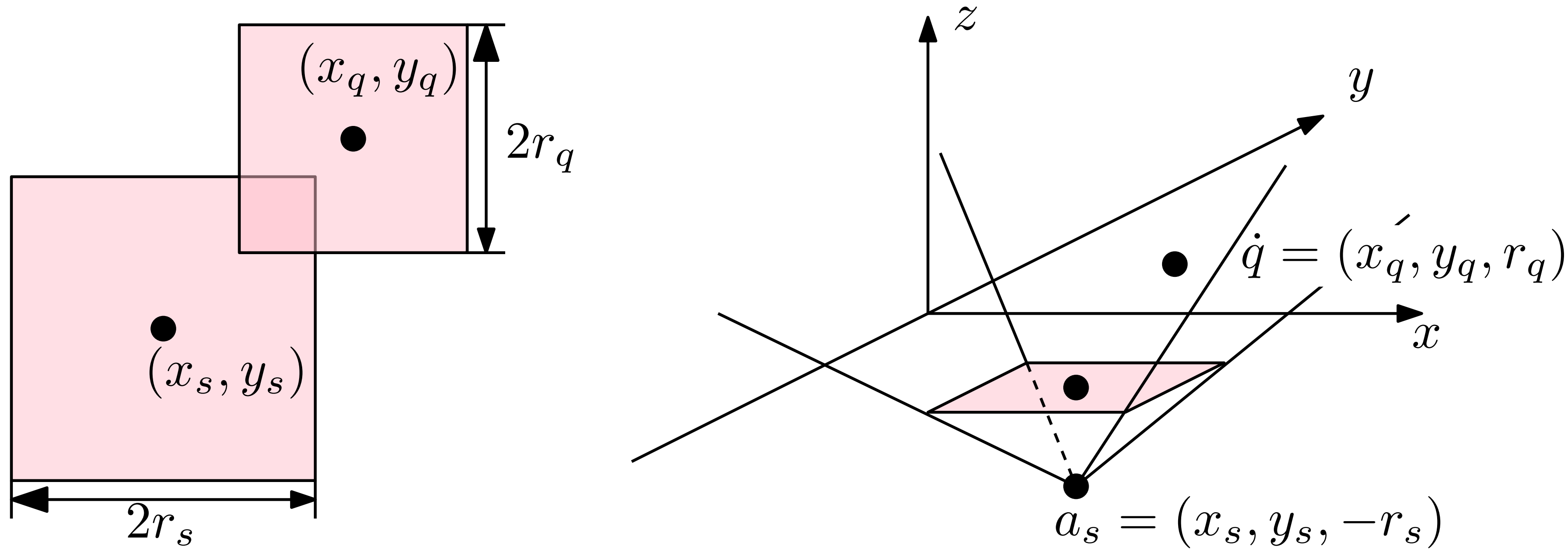


Rough idea:

- q has intersection with color class i iff $\dot{q}$ is above lower envelope of color-i cones.
- for each color class, compute lower envelope of cones

Data structure for squares

Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O}$ intersects $q\}$.

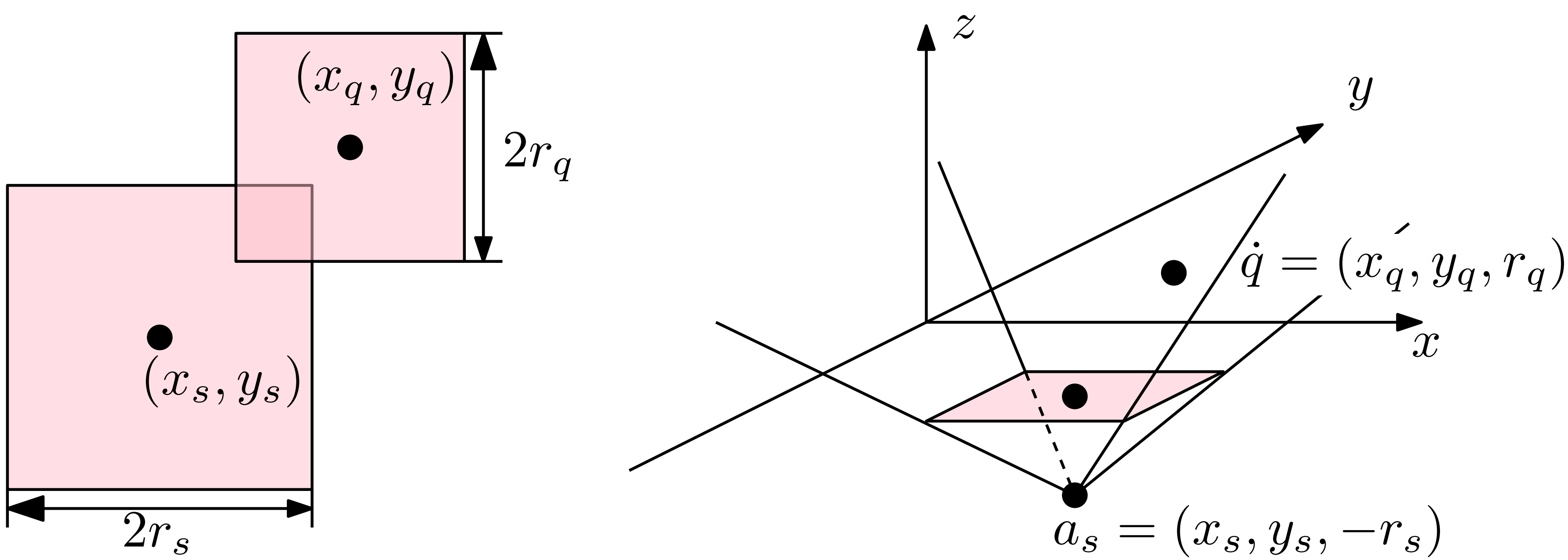


Rough idea:

- q has intersection with color class i iff $\dot{q}$ is above lower envelope of color-i cones.
- for each color class, compute lower envelope of cones
- slice the space above lower envelope into pw disjoint slabs of fixed directions

Data structure for squares

Rainbow colored intersection searching
Given squares $\mathcal{O}$ and coloring $C : \mathcal{O} \to [n]$, design a data structure that:
given square $q \in \mathcal{O}$ decides if all colors appear in $\{C(o) \mid o \in \mathcal{O}$ intersects $q\}$.



Rough idea:

- q has intersection with color class i iff $\dot{q}$ is above lower envelope of color-i cones.
- for each color class, compute lower envelope of cones
- slice the space above lower envelope into pw disjoint slabs of fixed directions
- build (reverse) range counting data structure on all slabs

Running time analysis for squares

Stabbing path λ gives:

$$\sum_{P} \sum_{s \in P} \sum_{r=r_P-2\Delta}^{r_P+\Delta} |\mathsf{Rep}_\lambda(\hat{N}^r[s])| \;=\; \widetilde{O}(\Delta \cdot n^{2-1/d}) = \widetilde{O}(\Delta \cdot n^{7/4}) \quad (d=4)$$

Running time analysis for squares

Stabbing path λ gives:

$$\sum_{P} \sum_{s \in P} \sum_{r=r_P-2\Delta}^{r_P+\Delta} |\mathsf{Rep}_{\lambda}(\hat{N}^r[s])| \ = \ \widetilde{O}(\Delta \cdot n^{2-1/d}) = \widetilde{O}(\Delta \cdot n^{7/4}) \quad (d=4)$$

The total length of all the intervals for $s \in P$ is at most $2|P| \cdot |R_P| = O(|P| \cdot n)$.
Ball growing running time for fixed P, r:

$$\widetilde{O}\Big(b \cdot \sum_{s \in P} \Big(|\mathsf{Rep}_{\lambda_P}(\hat{N}^{r-1}[s])| + |\mathsf{Rep}_{\lambda_P}(\hat{N}^{r}[s])|\Big) + |P|n/b\Big)$$

Running time analysis for squares

Stabbing path λ gives:

$$\sum_{P}\sum_{s\in P}\sum_{r=r_P-2\Delta}^{r_P+\Delta} |\mathsf{Rep}_\lambda(\hat{N}^r[s])| \;=\; \widetilde{O}(\Delta\cdot n^{2-1/d}) = \widetilde{O}(\Delta\cdot n^{7/4}) \quad (d=4)$$

The total length of all the intervals for $s\in P$ is at most $2|P|\cdot|R_P| = O(|P|\cdot n)$.
Ball growing running time for fixed P, r:

$$\widetilde{O}\Big(b\cdot\sum_{s\in P}\Big(|\mathsf{Rep}_{\lambda_P}(\hat{N}^{r-1}[s])| + |\mathsf{Rep}_{\lambda_P}(\hat{N}^{r}[s])|\Big) + |P|n/b\Big)$$

BFSes on ∂P → n^2/Δ

Constructing λ → $n^{5/4}$

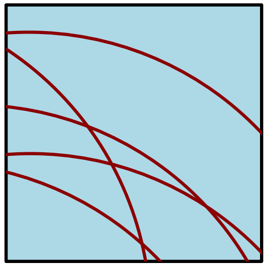
$$\widetilde{O}(n^2/\Delta + n^{5/4}) + \sum_{P}\sum_{r=r_P-2\Delta}^{r_P+\Delta}\widetilde{O}\left(b\cdot\sum_{s\in P}\Big((|\mathsf{Rep}_{\lambda_P}(\hat{N}^{r-1}[s])| + |\mathsf{Rep}_{\lambda_P}(\hat{N}^{r}[s])|\Big) + |P|n\right.$$

$$= \widetilde{O}(n^2/\Delta + n^{5/4}) + \widetilde{O}(b\Delta\cdot n^{7/4}) + \widetilde{O}(n^2\Delta/b) \quad (d=4)$$

$$= \widetilde{O}(n^{2-1/16}). \quad \text{(for optimal choices of } b=\Delta^2 \text{ and } \Delta = n^{1/16})$$

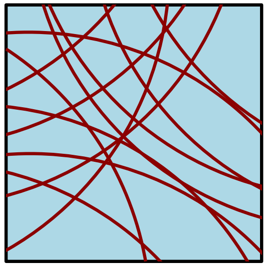
Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.



Unit disks: types, resamples

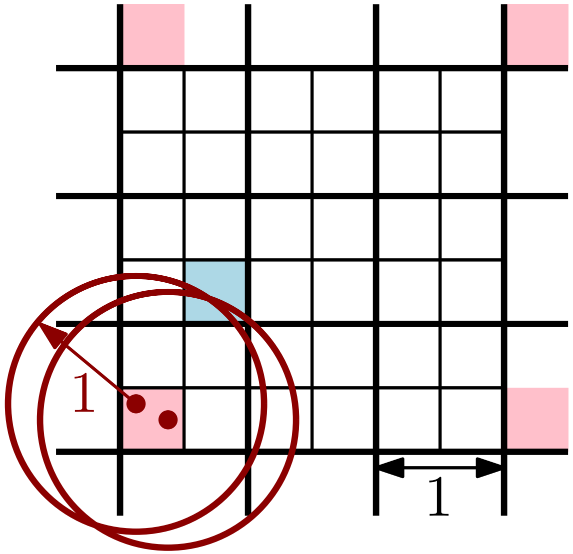
Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

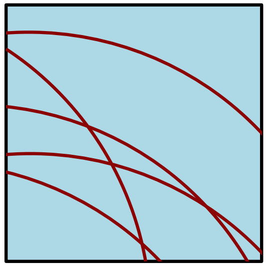


Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

Idea: split into $O(1)$ *modulo classes*.



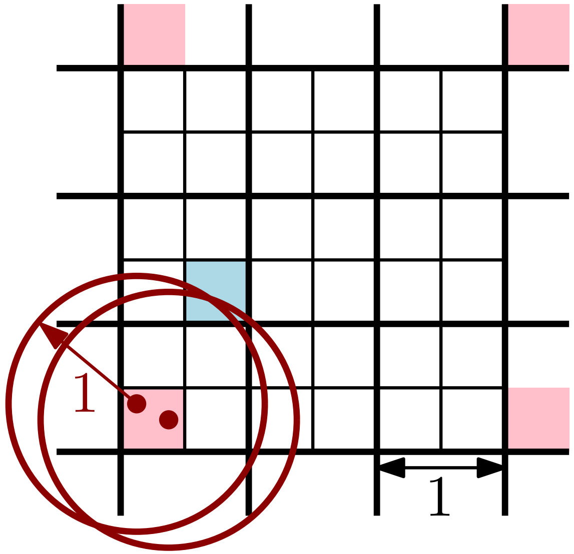


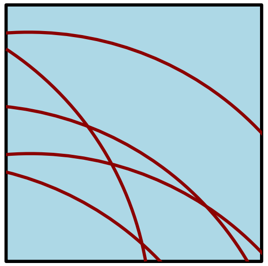
Pseudolines

Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

Idea: split into $O(1)$ *modulo classes*.





Pseudolines

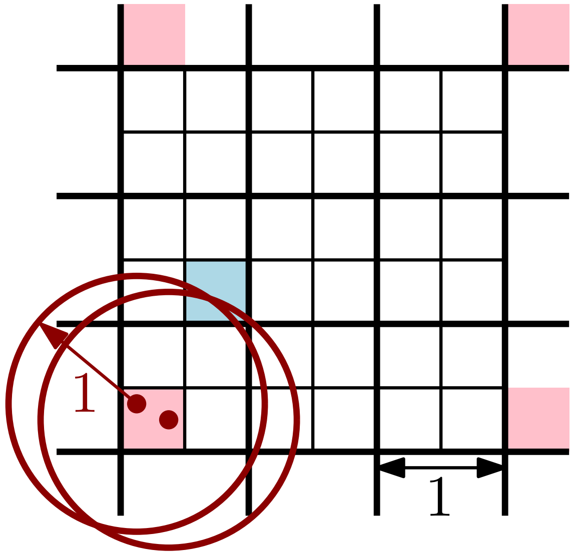
Technical difficulties:

- The intermediate sets are no longer neighborhood balls (new set system).
  $\rightarrow$ Fortunately VC-dimension is still ≤ 4, but *only for fixed radius* r.

Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

Idea: split into $O(1)$ *modulo classes*.



Pseudolines

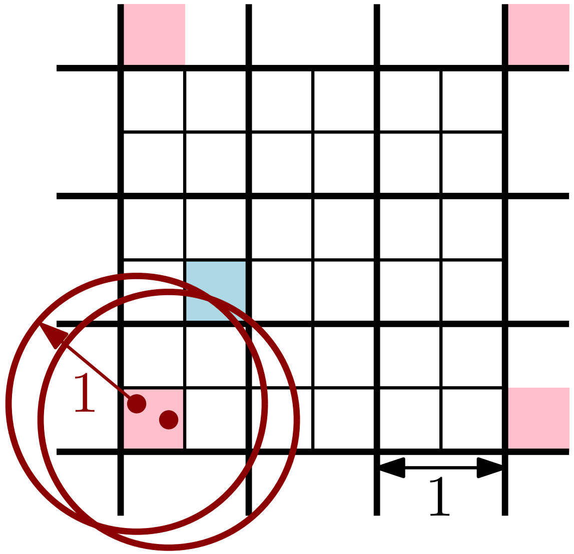
Technical difficulties:

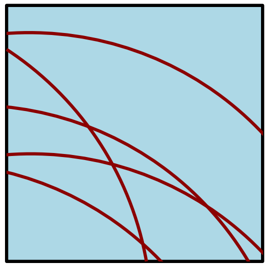
- The intermediate sets are no longer neighborhood balls (new set system).
  $\rightarrow$ Fortunately VC-dimension is still ≤ 4, but *only for fixed radius* r.
- Need new stabbing path & representations for each r (and each P)

Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

Idea: split into $O(1)$ *modulo classes*.





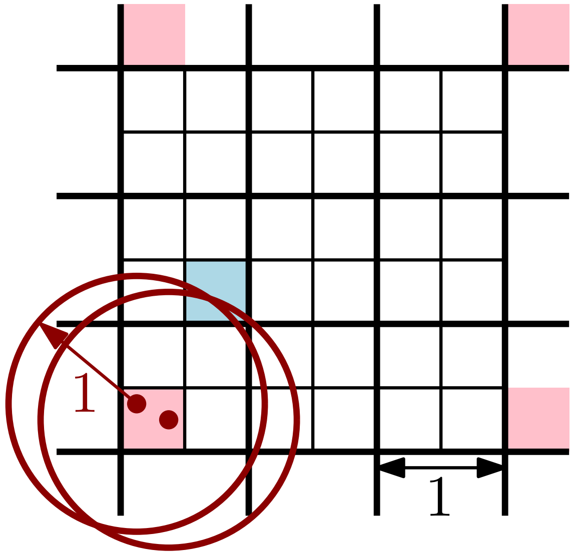
Technical difficulties:

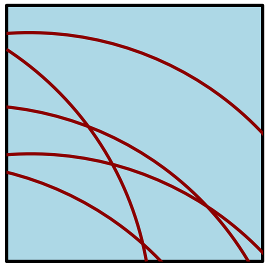
- The intermediate sets are no longer neighborhood balls (new set system).
  $\rightarrow$ Fortunately VC-dimension is still ≤ 4, but *only for fixed radius* r.
- Need new stabbing path & representations for each r (and each P)
- Need to work with balls from two or three different types.
  $\rightarrow$ Fortunately, the combined set systems still have VC-dimension ≤ 8.

Unit disks: types, resamples

Base data strucutre problem is related to Hopcroft's problem, $\Omega(n^{1/3})$ query time likely.

Idea: split into $O(1)$ *modulo classes*.





Pseudolines

Technical difficulties:

- The intermediate sets are no longer neighborhood balls (new set system).
  $\rightarrow$ Fortunately VC-dimension is still ≤ 4, but *only for fixed radius* r.
- Need new stabbing path & representations for each r (and each P)
- Need to work with balls from two or three different types.
  $\rightarrow$ Fortunately, the combined set systems still have VC-dimension ≤ 8.
- Switching stabbing paths is too costly if pieces are small.
  $\rightarrow$ Work only with pieces larger than a threshold; for small pieces, we switch to a different algo (based on distance compression)

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for Diameter-2 in UDG

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for $\textsc{Diameter-}2$ in UDG
- Subquadratic algo for $\textsc{Diameter-}2$ and $\textsc{Diameter-}3$ of unit cubes in $\mathbb{R}^3$.

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for Diameter-2 in UDG
- Subquadratic algo for Diameter-2 and Diameter-3 of unit cubes in $\mathbb{R}^3$.
- Algorithms/hardness for Diameter-k in intersection graphs.
  Sometimes $d = g(k)$, gives $O(n^{2-f(k)})$ diameter algo

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for $\textsc{Diameter}$-2 in UDG
- Subquadratic algo for $\textsc{Diameter}$-2 and $\textsc{Diameter}$-3 of unit cubes in $\mathbb{R}^3$.
- Algorithms/hardness for $\textsc{Diameter}$-k in intersection graphs.
  Sometimes $d = g(k)$, gives $O(n^{2-f(k)})$ diameter algo

(Very bold) Conjecture
There is some $f(d) > 0$ such that if $\mathcal{G}$ is a (nice?) graph class with distance VC-dim$\leq d$, then there is a $O(n^{2-f(d)})$ time algorithm for diameter.

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for Diameter-2 in UDG
- Subquadratic algo for Diameter-2 and Diameter-3 of unit cubes in $\mathbb{R}^3$.
- Algorithms/hardness for Diameter-k in intersection graphs.
  Sometimes $d = g(k)$, gives $O(n^{2-f(k)})$ diameter algo

(Very bold) Conjecture
There is some $f(d) > 0$ such that if $\mathcal{G}$ is a (nice?) graph class with distance VC-dim$\leq d$, then there is a $O(n^{2-f(d)})$ time algorithm for diameter.

Open:

?1 disk graphs
(needs data strucutre breakthrough)?

?2 Super-linear fine-grained lower bounds?

Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for Diameter-2 in UDG
- Subquadratic algo for Diameter-2 and Diameter-3 of unit cubes in $\mathbb{R}^3$.
- Algorithms/hardness for Diameter-k in intersection graphs.
  Sometimes $d = g(k)$, gives $O(n^{2-f(k)})$ diameter algo

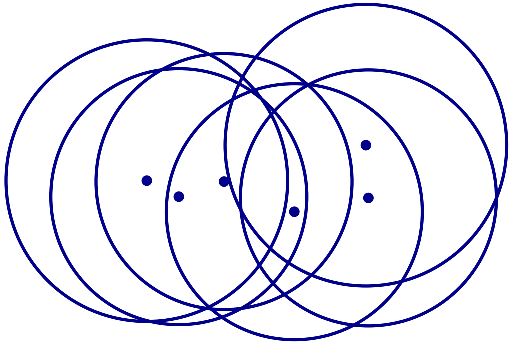
(Very bold) Conjecture
There is some $f(d) > 0$ such that if $\mathcal{G}$ is a (nice?) graph class with distance VC-dim$\leq d$, then there is a $O(n^{2-f(d)})$ time algorithm for diameter.

Open:

?1 disk graphs
(needs data strucutre breakthrough)?

?2 Super-linear fine-grained lower bounds?

Diameter-2 for disks or radii $[1, 1+\varepsilon]$?



Ongoing work, a conjecture, and open problems

VC-dim d + efficient interval cover DS
$\rightsquigarrow O(n^{2-f(d)})$ diameter computation/distance oracle

Work in progress:

- $\widetilde{O}(n^{4/3})$ for Diameter-2 in UDG
- Subquadratic algo for Diameter-2 and Diameter-3 of unit cubes in $\mathbb{R}^3$.
- Algorithms/hardness for Diameter-k in intersection graphs.
  Sometimes $d = g(k)$, gives $O(n^{2-f(k)})$ diameter algo

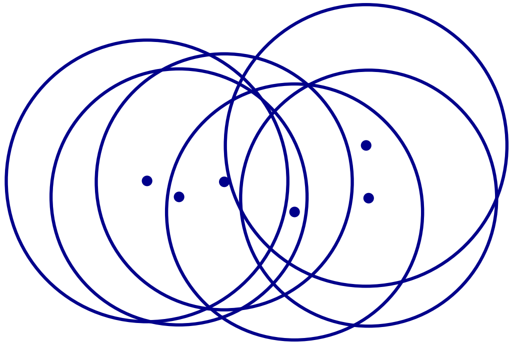
(Very bold) Conjecture
There is some $f(d) > 0$ such that if $\mathcal{G}$ is a (nice?) graph class with distance VC-dim$\leq d$, then there is a $O(n^{2-f(d)})$ time algorithm for diameter.

Open:

?1 disk graphs
(needs data strucutre breakthrough)?

?2 Super-linear fine-grained lower bounds?

Diameter-2 for disks or radii $[1, 1+\varepsilon]$?



Thanks for listening!

Diameter:

graph class	best previous		new
planar	$\widetilde{O}(n^{5/3})$	[Cab18, GKM+21]	
K_h-minor-free	$\widetilde{O}(n^{2-1/(3h-1)})$	[DHV22, LW24]	$\widetilde{O}(n^{2-1/(2h-2)})$
VC-dim.-bounded	$\widetilde{O}(\min\{Dmn^{1-1/d},\ mn\})$	[DHV22, DKP24]	$\widetilde{O}(mn^{1-1/(2d)})$
unit square	$\widetilde{O}(\min\{Dn^{7/4},\ n^2\})$	[DKP24]	$O^*(n^{2-1/8})$
arbitrary square	$\widetilde{O}(n^2)$	[CS19]	$\widetilde{O}(n^{2-1/12})$
unit disk	$O(n^2\sqrt{\frac{\log\log n}{\log n}})$	[CS16]	$O^*(n^{2-1/18})$

Distance oracle
(construction time/space)
Query: $\widetilde{O}(1)$.

graph class	best previous		new
planar	$n^{3/2+o(1)}$, $n^{1+o(1)}$	[CGL+23]	
K_h-minor-free	$\widetilde{O}(n^{2-1/(3h-1)})$	[LW24]	
VC-dim.-bounded	$O(mn)$, $O(n^2)$	folklore	$\widetilde{O}(mn^{1-1/(4d+1)})$
unit square	$\widetilde{O}(n^2)$	[CS19]	$O^*(n^{2-1/16})$
arbitrary square	$\widetilde{O}(n^2)$	[CS19]	$\widetilde{O}(n^{2-1/20})$
unit disk	$O(n^2\sqrt{\frac{\log\log n}{\log n}})$	[CS16]	$O^*(n^{2-1/20})$