A Lower Bound for Union Volume Estimation and Approximating Klee's Measure Problem

Karl Bringmann Kasper Green Larsen **André Nusser**
Eva Rotenberg Yanheng Wang

Introduction

The Problems

Union Volume Estimation

Klee's Measure Problem

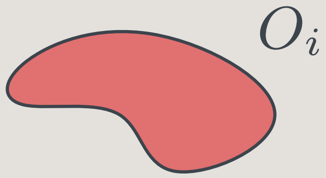
The Problems

Union Volume Estimation

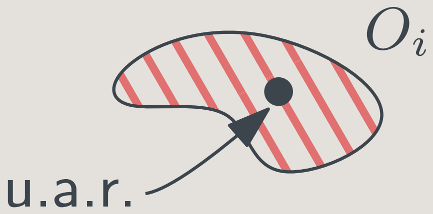
Given:

Objects $O_1, \ldots, O_n$ with access:

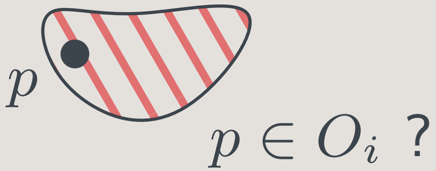
1) $\mathrm{Volume}(i)$:



2) $\mathrm{Sample}(i)$



3) $\mathrm{Contains}(p, i)$



Klee's Measure Problem

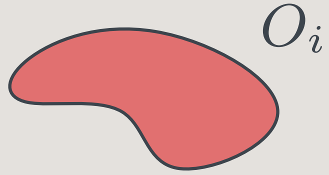
The Problems

Union Volume Estimation

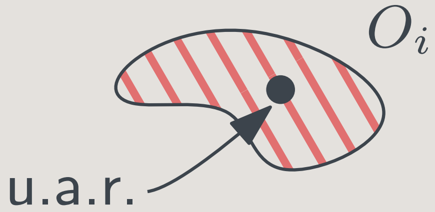
Given:

Objects $O_1, \ldots, O_n$ with access:

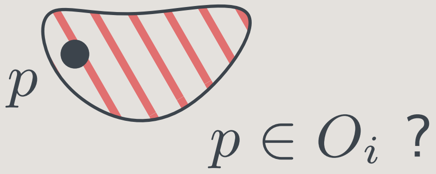
1) $\text{Volume}(i)$:



2) $\text{Sample}(i)$



3) $\text{Contains}(p, i)$



Want:

$(1+\varepsilon)$-approximation of $\text{Vol}(\bigcup_i O_i)$ with constant probability

Klee's Measure Problem

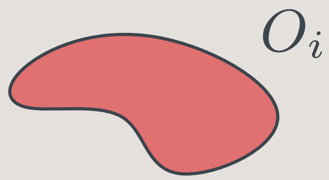
The Problems

Union Volume Estimation

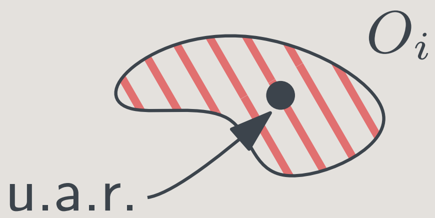
Given:

Objects $O_1, \ldots, O_n$ with access:

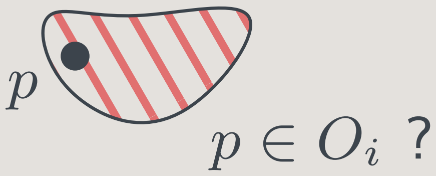
1) $\mathrm{Volume}(i)$:



2) $\mathrm{Sample}(i)$



3) $\mathrm{Contains}(p, i)$



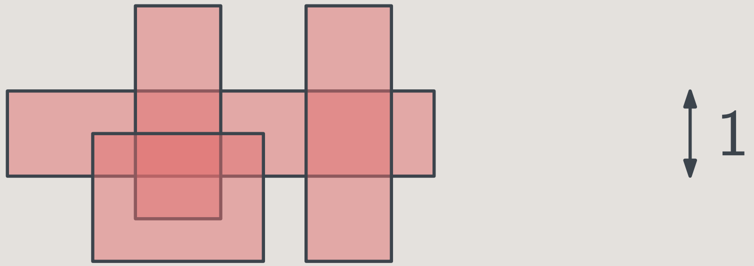
Want:

$(1+\varepsilon)$-approximation of $\mathrm{Vol}(\bigcup_i O_i)$ with constant probability

Klee's Measure Problem

Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



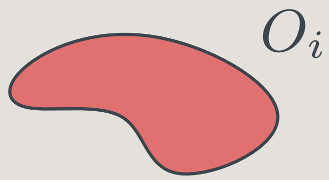
The Problems

Union Volume Estimation

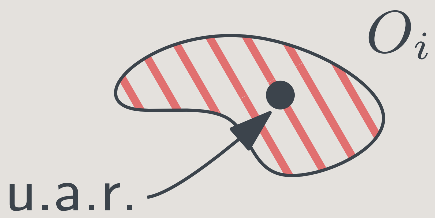
Given:

Objects $O_1, \ldots, O_n$ with access:

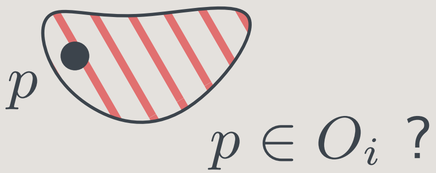
1) $\mathrm{Volume}(i)$:



2) $\mathrm{Sample}(i)$



3) $\mathrm{Contains}(p, i)$



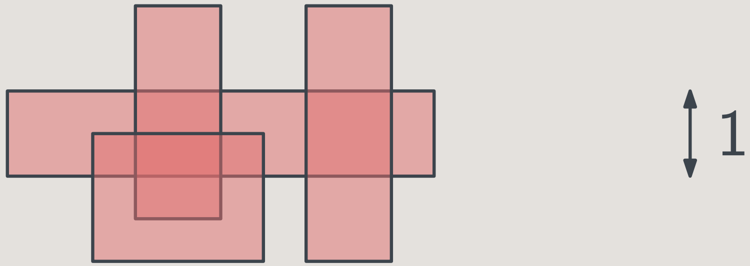
Want:

$(1+\varepsilon)$-approximation of $\mathrm{Vol}(\bigcup_i O_i)$ with constant probability

Klee's Measure Problem

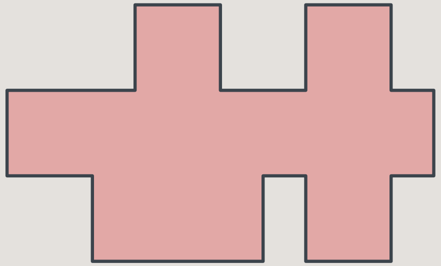
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Want:

$\mathrm{Vol}(\bigcup_i O_i)$, or some approximation

Vol() = 10

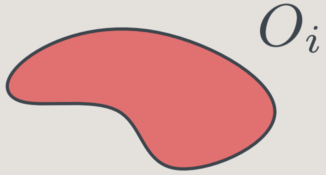
The Problems

Union Volume Estimation

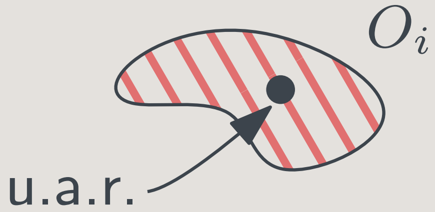
Given:

Objects $O_1, \ldots, O_n$ with access:

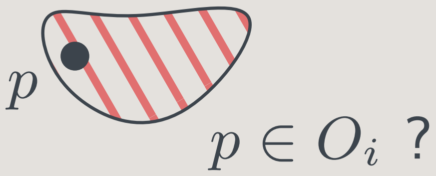
1) $\mathrm{Volume}(i)$:



2) $\mathrm{Sample}(i)$



3) $\mathrm{Contains}(p, i)$



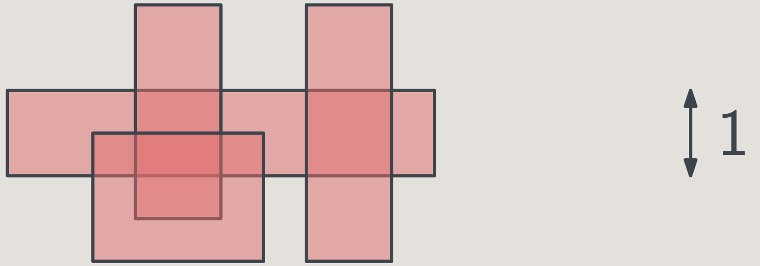
Want:

$(1+\varepsilon)$-approximation of $\mathrm{Vol}(\bigcup_i O_i)$ with constant probability

Klee's Measure Problem

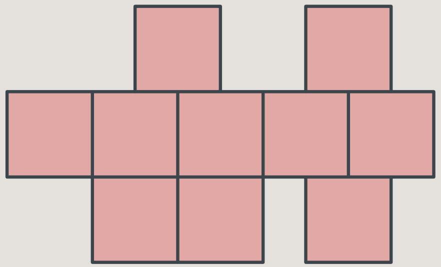
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Want:

$\mathrm{Vol}(\bigcup_i O_i)$, or some approximation

$\mathrm{Vol}($  $) = 10$

Motivation

Union Volume Estimation

DNF Counting

Given DNF, approximate number of satisfying assignments.

$$(A \wedge B) \vee (A \wedge C)$$

answer: 3 satisfying assignments

Motivation

Union Volume Estimation

DNF Counting

Given DNF, approximate number of satisfying assignments.

$$(A \wedge B) \vee (A \wedge C)$$

answer: 3 satisfying assignments

Network Reliability

Given a graph G and an edge failure probability p, what is the probability that G gets disconnected?

Motivation

Union Volume Estimation

DNF Counting

Given DNF, approximate number of satisfying assignments.

$$(A \wedge B) \vee (A \wedge C)$$

answer: 3 satisfying assignments

Network Reliability

Given a graph G and an edge failure probability p, what is the probability that G gets disconnected?

Klee's Measure Problem

Motivation

Union Volume Estimation

DNF Counting

Given DNF, approximate number of satisfying assignments.

$$(A \wedge B) \vee (A \wedge C)$$

answer: 3 satisfying assignments

Network Reliability

Given a graph G and an edge failure probability p, what is the probability that G gets disconnected?

Klee's Measure Problem

Klee's Measure Problem

Polygon Containement

What is the largest scaling such that one rectilinear polygon can be translated into another one?

Motivation

Union Volume Estimation

DNF Counting

Given DNF, approximate number of satisfying assignments.

$$(A \wedge B) \vee (A \wedge C)$$

answer: 3 satisfying assignments

Network Reliability

Given a graph G and an edge failure probability p, what is the probability that G gets disconnected?

Klee's Measure Problem

Klee's Measure Problem

Polygon Containement

What is the largest scaling such that one rectilinear polygon can be translated into another one?

Continuous Graph k-Center

Given a graph G, find k centers (potentially also on the edges) such that each vertex is in distance r to some center.

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

approximate:

- $O(n/\varepsilon^2)$ [Karp, Luby, Madras '89], [Bringmann, Friedrich '10]

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

approximate:

- $O(n/\varepsilon^2)$ [Karp, Luby, Madras '89], [Bringmann, Friedrich '10]

Big Questions:

1) Can $O(n/\varepsilon^2)$ be improved for Union Volume Estimation?

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

approximate:

- $O(n/\varepsilon^2)$ [Karp, Luby, Madras '89], [Bringmann, Friedrich '10]

Big Questions:

1) Can $O(n/\varepsilon^2)$ be improved for Union Volume Estimation? **NO!**

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

approximate:

- $O(n/\varepsilon^2)$ [Karp, Luby, Madras '89], [Bringmann, Friedrich '10]

Big Questions:

1) Can $O(n/\varepsilon^2)$ be improved for Union Volume Estimation? **NO!**
2) Algorithm for Klee's Measure Problem faster than $O(n/\varepsilon^2)$?

Related Work

Union Volume Estimation

- Initial work [Luby '84]
- Introduced model [Karp, Luby '85]
- $O(n/\varepsilon^2)$ algorithm [Karp, Luby, Madras '89]

Klee's Measure Problem

exact:

- $O(n^{d/2})$ [Chan '13]
- $\tilde{\Omega}(n^{d/2})$ (combinatorial) [Chan '10]

approximate:

- $O(n/\varepsilon^2)$ [Karp, Luby, Madras '89], [Bringmann, Friedrich '10]

Big Questions:

1) Can $O(n/\varepsilon^2)$ be improved for Union Volume Estimation? **NO!**
2) Algorithm for Klee's Measure Problem faster than $O(n/\varepsilon^2)$? **YES!**

Our Results

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

Our Results

Union Volume Estimation

“requires $\Omega(n/\varepsilon^2)$ time”

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Our Results

Union Volume Estimation

“requires $\Omega(n/\varepsilon^2)$ time”

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Klee’s Measure Problem

“can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time”

Our Results

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Klee's Measure Problem

"can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time"

- must exploit geometry!
- hides $\log^d n$ factors

Our Results

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Klee's Measure Problem

"can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time"

- must exploit geometry!
- hides $\log^d n$ factors

	n/ε^2	$n + \frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{\sqrt{n}}$		
$\varepsilon = 0.001$ $n = 10^6$		

Our Results

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Klee's Measure Problem

"can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time"

- must exploit geometry!
- hides $\log^d n$ factors

	n/ε^2	$n + \frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{\sqrt{n}}$	n^2	n
$\varepsilon = 0.001$ $n = 10^6$		

Our Results

Union Volume Estimation

“requires $\Omega(n/\varepsilon^2)$ time”

- Previously known: $\Omega(n + \frac{1}{\varepsilon^2})$
- Reduction from Gap-Hamming Problem

Klee’s Measure Problem

“can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time”

- must exploit geometry!
- hides $\log^d n$ factors

	n/ε^2	$n + \frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{\sqrt{n}}$	n^2	n
$\varepsilon = 0.001$ $n = 10^6$	$\sim 10^{12}$	$\sim 10^6$

Lower Bound
for
Union Volume Estimation

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Theorem [Indyk, Woodruff '05]
Solving the Gap-Hamming Problem requires $\Omega(T)$ queries into x, y.

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Theorem [Indyk, Woodruff '05]
Solving the Gap-Hamming Problem requires $\Omega(T)$ queries into x, y.

$\Longrightarrow$

Theorem [our work]
Union Volume Estimation requires $\Omega(n/\varepsilon^2)$ queries.

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

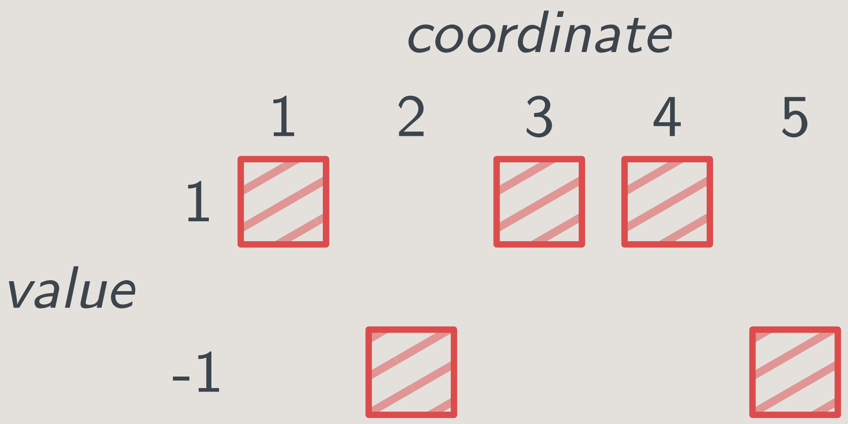
Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

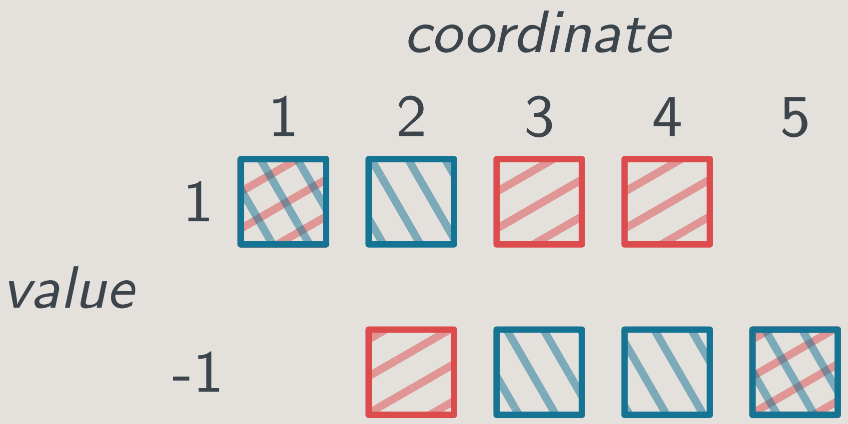
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

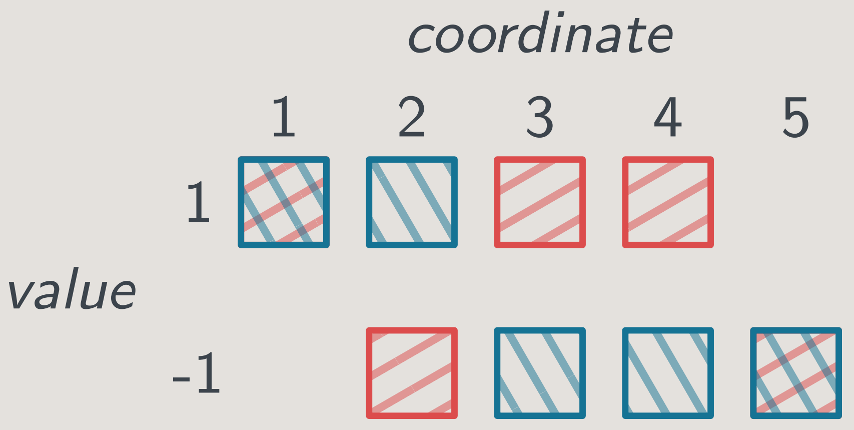
Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

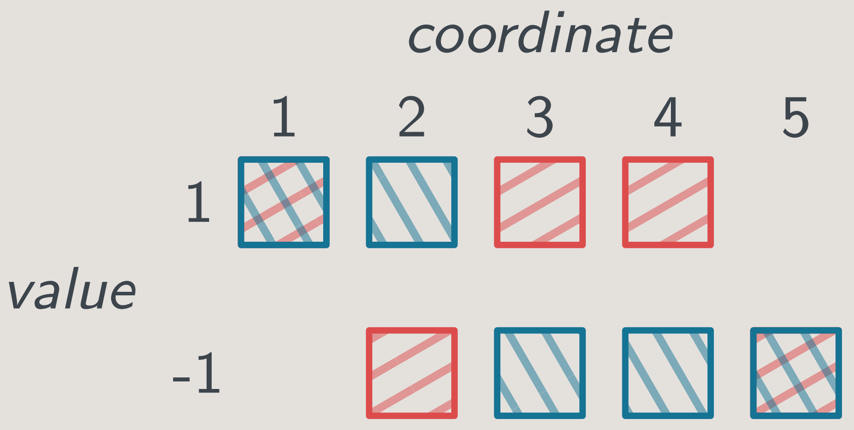
Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

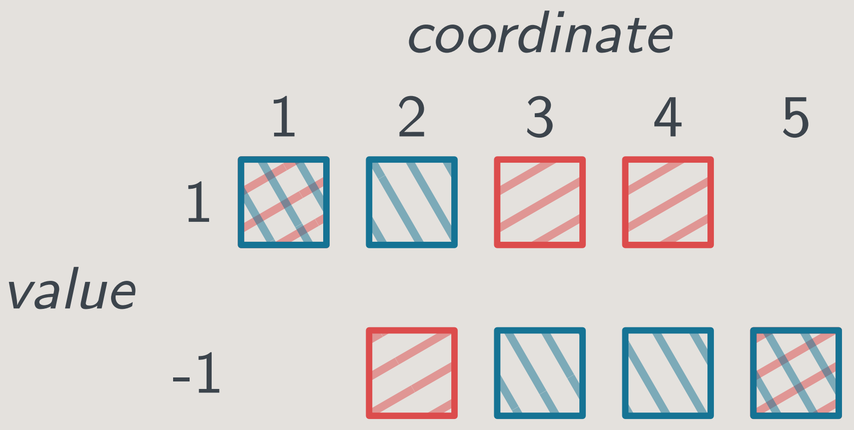
Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

$\in [T, 2T]$

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

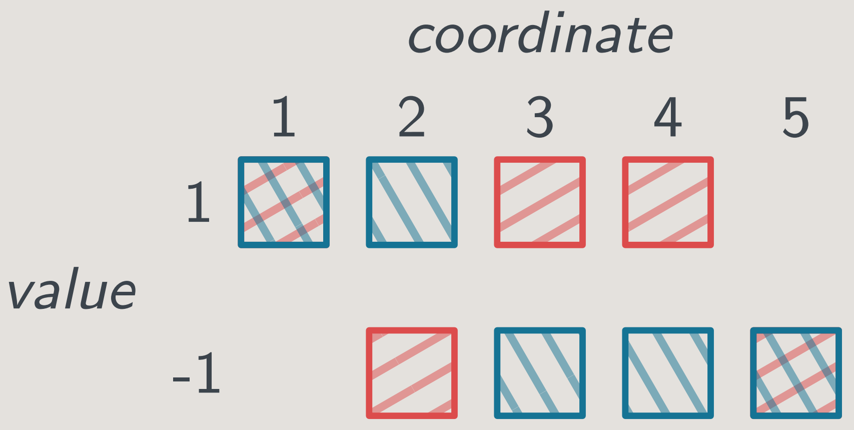
Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$\in [T, 2T]$

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

$\implies$ approximates $\langle x, y \rangle$ within $\pm \varepsilon T$

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

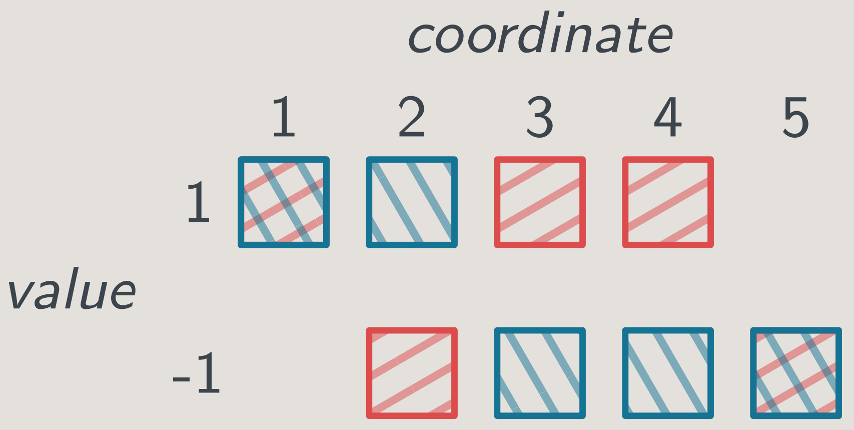
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$\in [T, 2T]$

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

$\implies$ approximates $\langle x, y \rangle$ within $\pm \varepsilon T$

$\frac{1}{\sqrt{T}}$

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

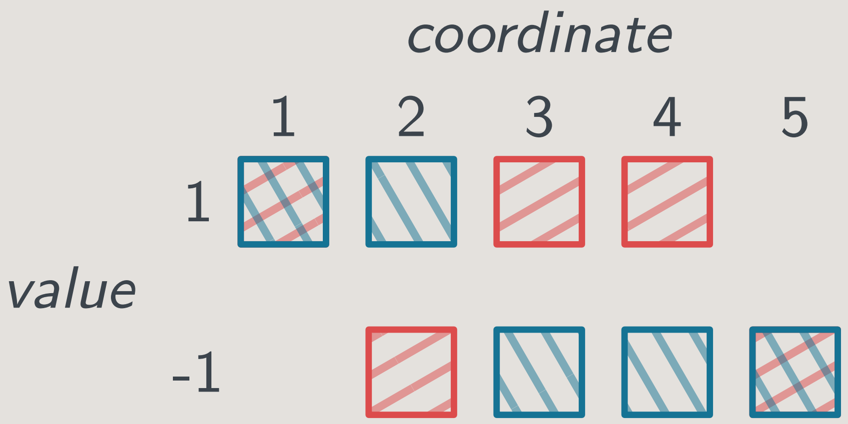
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$\in [T, 2T]$

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

$\implies$ approximates $\langle x, y \rangle$ within $\pm \varepsilon T = \pm\sqrt{T}$

$\frac{1}{\sqrt{T}}$

Lower Bound

Warm up: $\Omega(\frac{1}{\varepsilon^2})$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

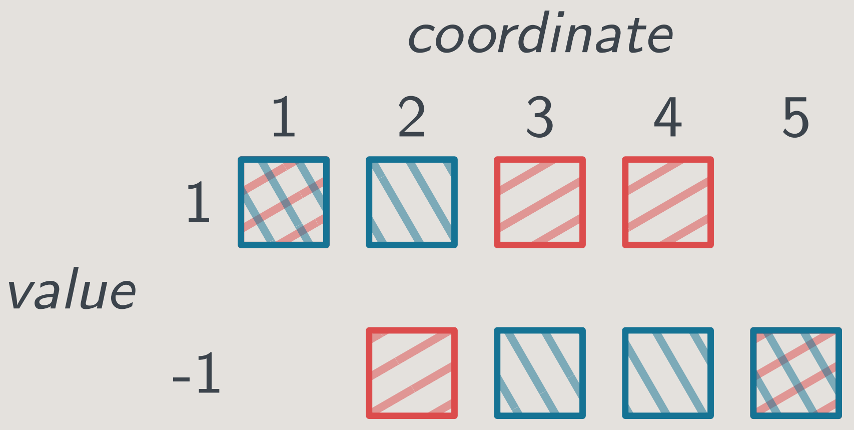
Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



We have: $\langle x, y \rangle = 3T - 2 \cdot \mathrm{Vol}(\square \cup \square)$

Given a $(1+\varepsilon)$-approximation:

$\in [T, 2T]$

$$\langle x, y \rangle = 3T - 2(1 \pm \varepsilon) \cdot \mathrm{Vol}(\square \cup \square)$$

$\Longrightarrow$ approximates $\langle x, y \rangle$ within $\pm \varepsilon T = \pm \sqrt{T}$

$\frac{1}{\sqrt{T}}$

$\Longrightarrow$ we need $\Omega(T) = \Omega(\frac{1}{\varepsilon^2})$ queries!

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Idea: “hide coordindates” using random permutations

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Idea: "hide coordindates" using random permutations

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

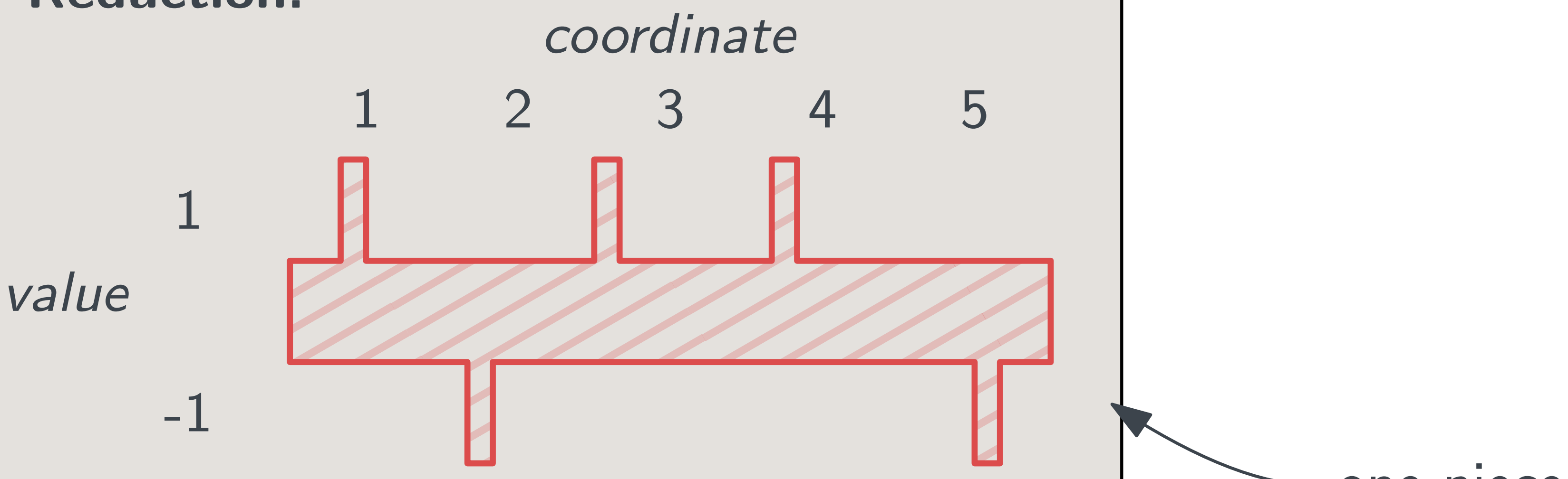
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

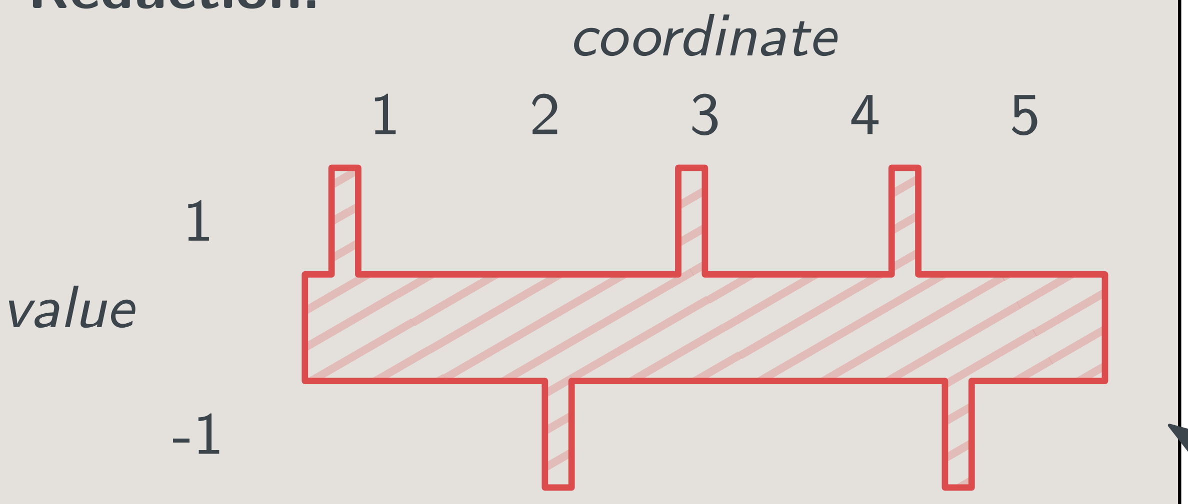
Idea: "hide coordindates" using random permutations

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



one piece

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether
$\langle x, y\rangle \leq -\sqrt{T}$, or
$\langle x, y\rangle \geq \sqrt{T}$

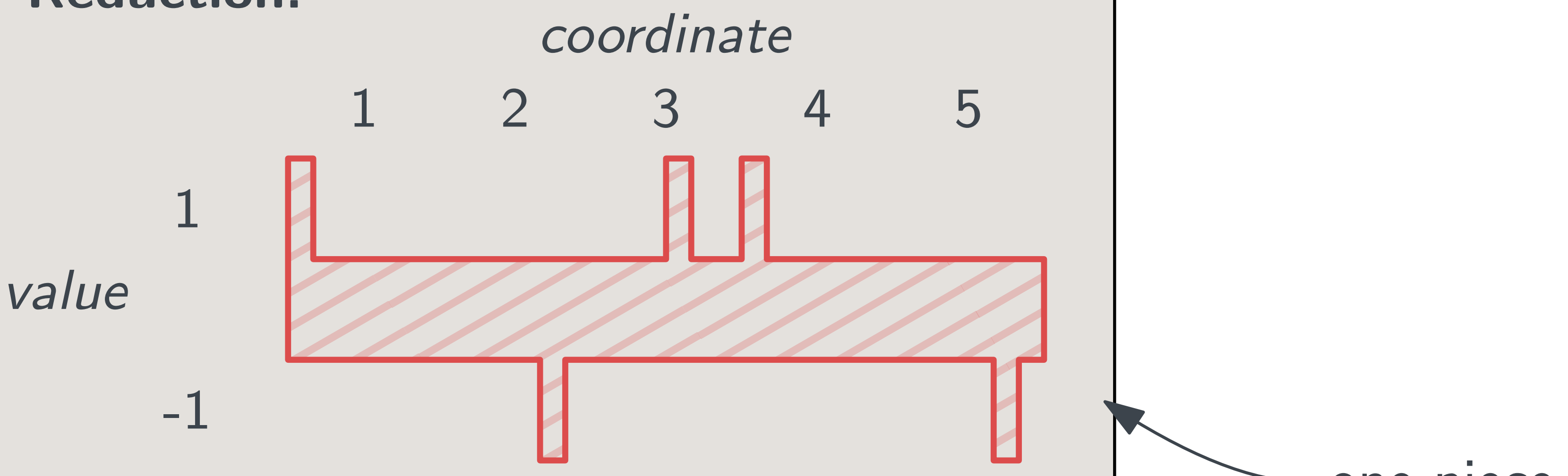
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

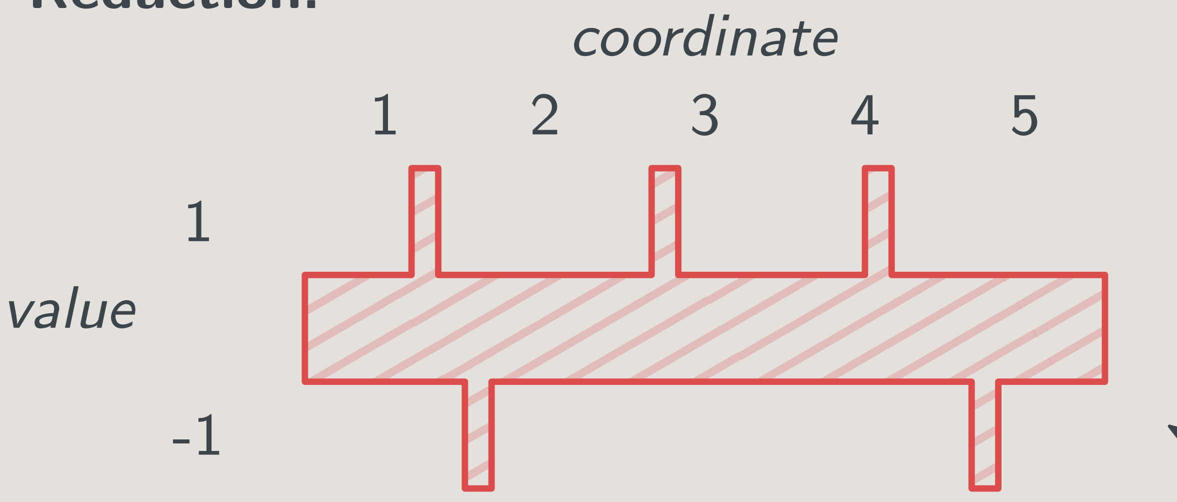
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



one piece

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether
$\langle x, y\rangle \leq -\sqrt{T}$, or
$\langle x, y\rangle \geq \sqrt{T}$

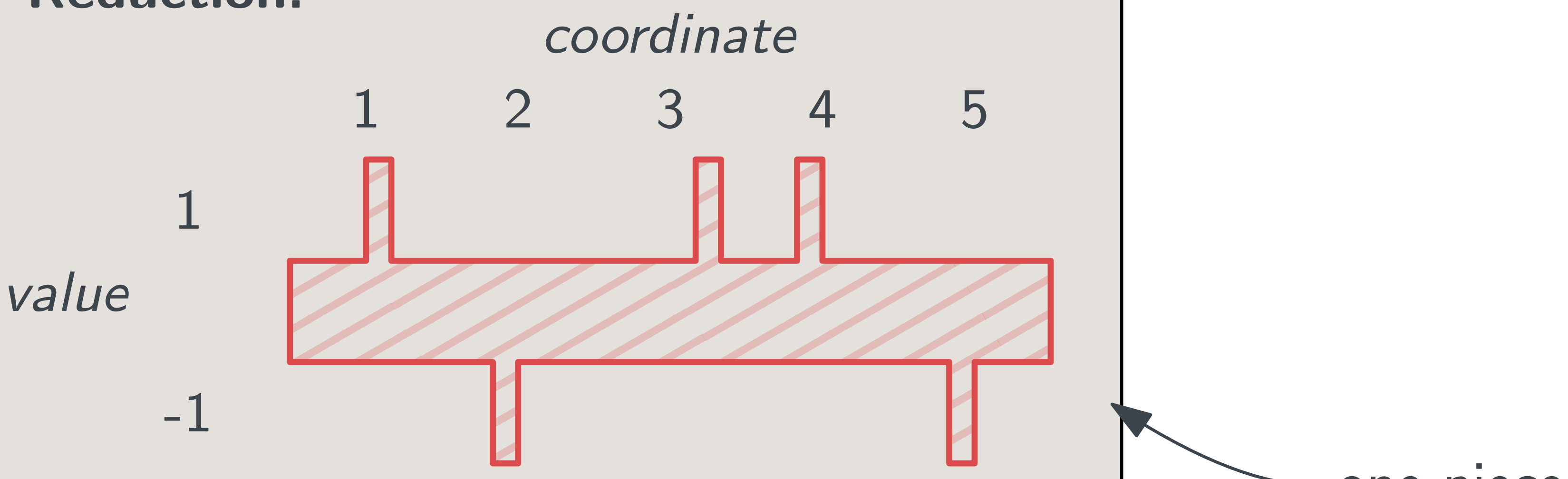
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

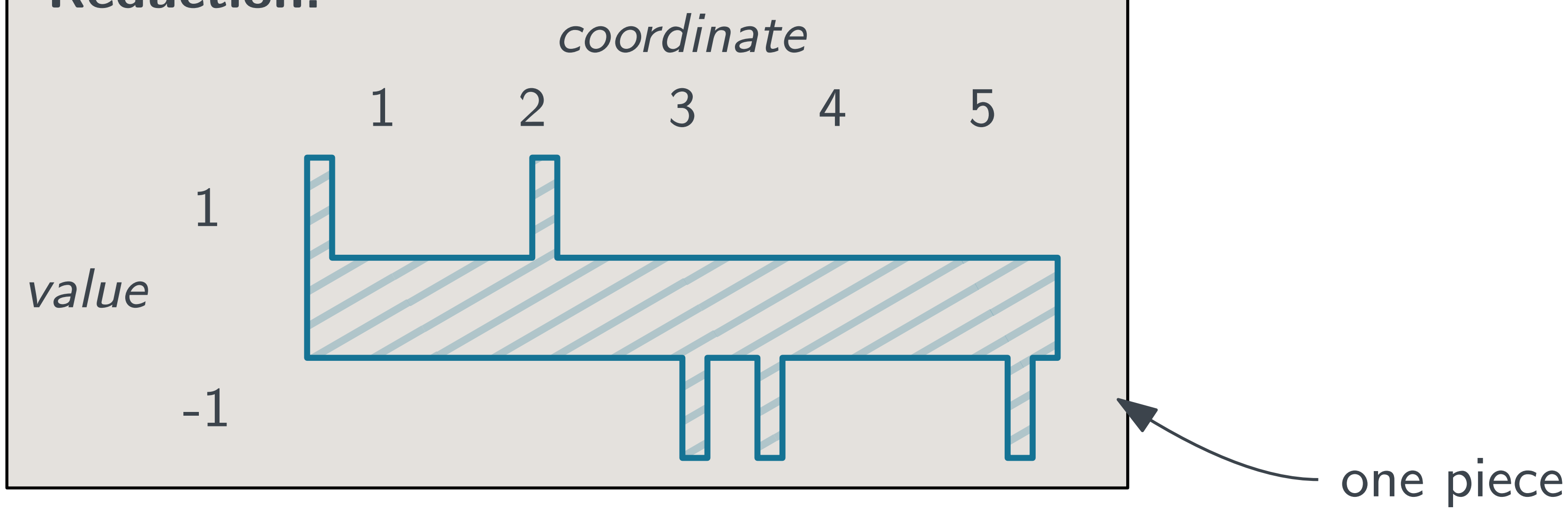
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

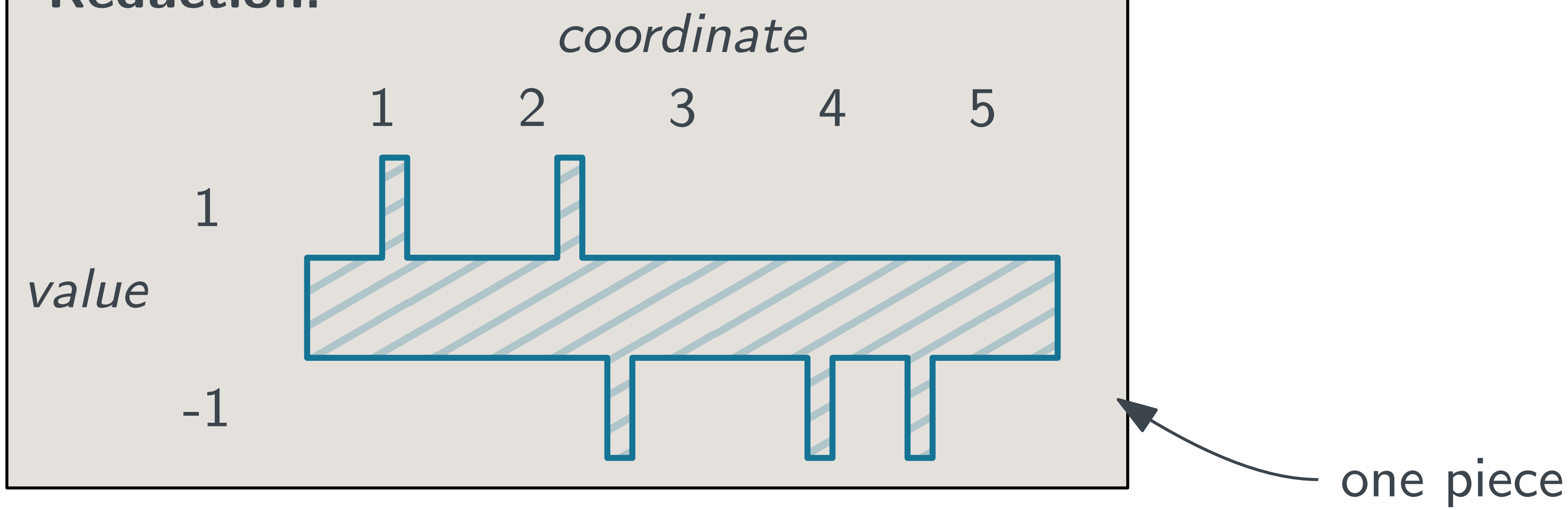
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x,\ y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y\rangle \leq -\sqrt{T}$, or

$\langle x, y\rangle \geq \sqrt{T}$

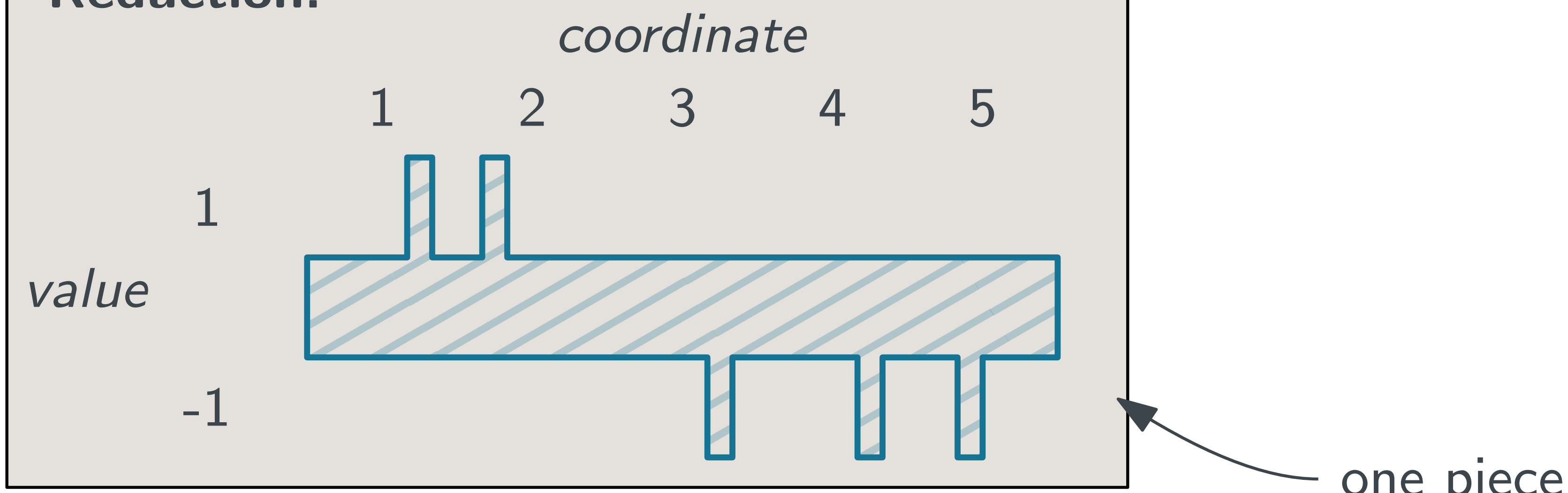
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

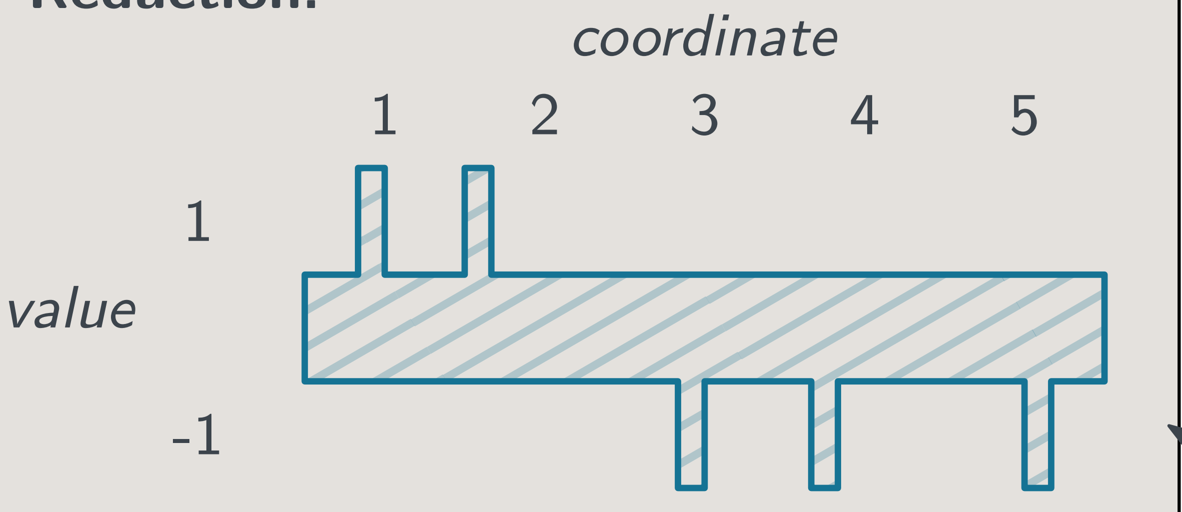
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



one piece

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y\rangle \leq -\sqrt{T}$, or

$\langle x, y\rangle \geq \sqrt{T}$

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

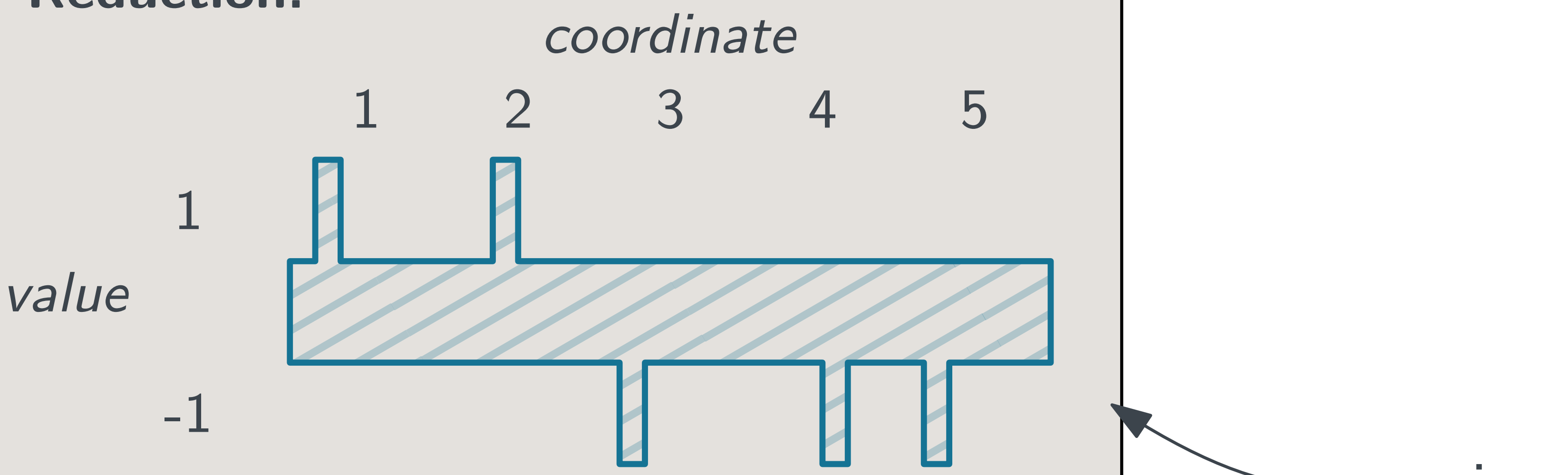
Idea: "hide coordindates" using random permutations

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y\rangle \le -\sqrt{T}$, or

$\langle x, y\rangle \ge \sqrt{T}$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

$\Omega(n/\varepsilon^2)$ lower bound

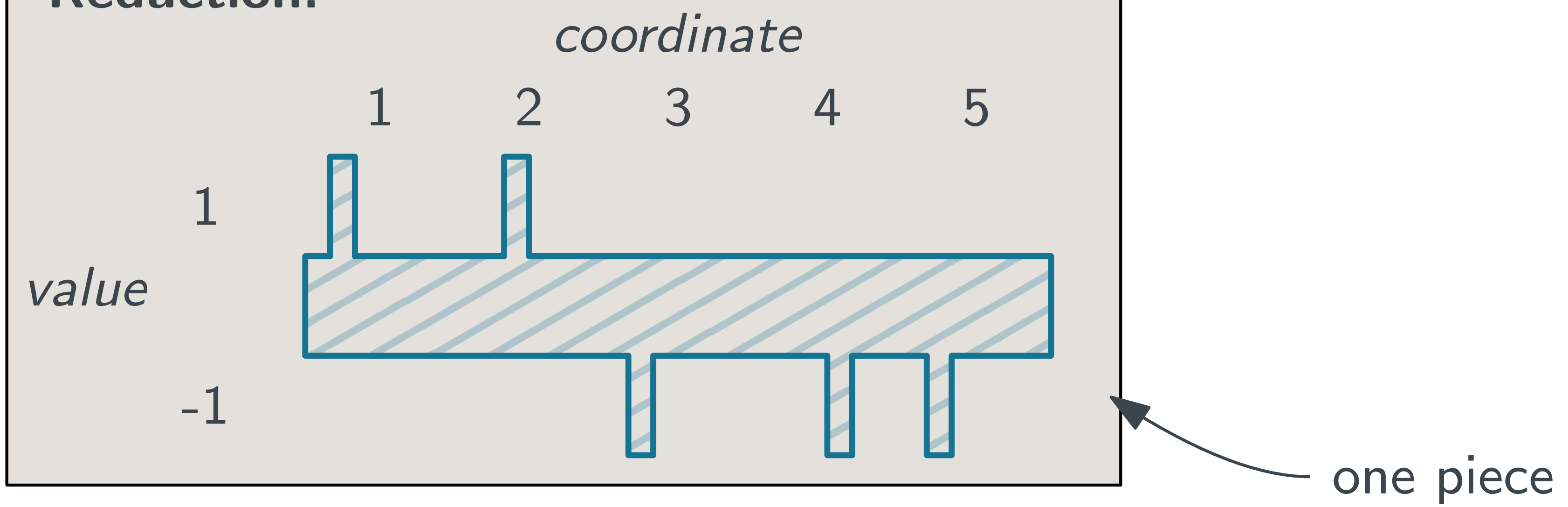
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



$P(\text{sample a coordinate point}) = \frac{1}{n}$

$P(\text{containment check succeeds}) = \frac{1}{n}$

Lower Bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether

$\langle x, y \rangle \leq -\sqrt{T}$, or

$\langle x, y \rangle \geq \sqrt{T}$

$\Omega(n/\varepsilon^2)$ lower bound

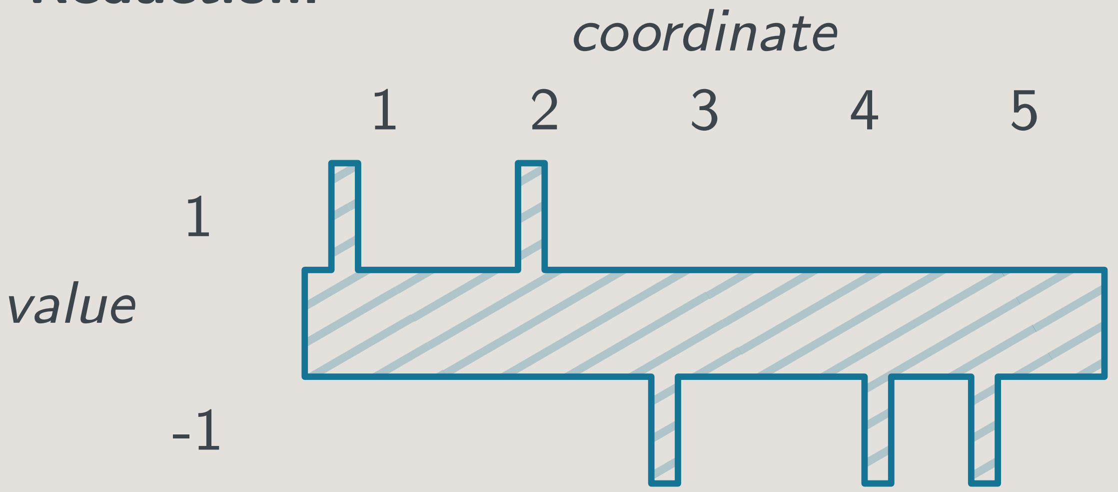
Idea: "hide coordindates" using random permutations

Input:

$$x = (1, -1, 1, 1, -1)$$

$$y = (1, 1, -1, -1, -1)$$

Reduction:



one piece

$P(\text{sample a coordinate point}) = \frac{1}{n}$

$P(\text{containment check succeeds}) = \frac{1}{n}$

↳ $O(n)$ queries for checking single bit

Lower Bound

$\Omega(n/\varepsilon^2)$ lower bound

Gap-Hamming Problem

Given: $x, y \in \{-1, 1\}^T$

Goal: decide whether
$\langle x, y \rangle \leq -\sqrt{T}$, or
$\langle x, y \rangle \geq \sqrt{T}$

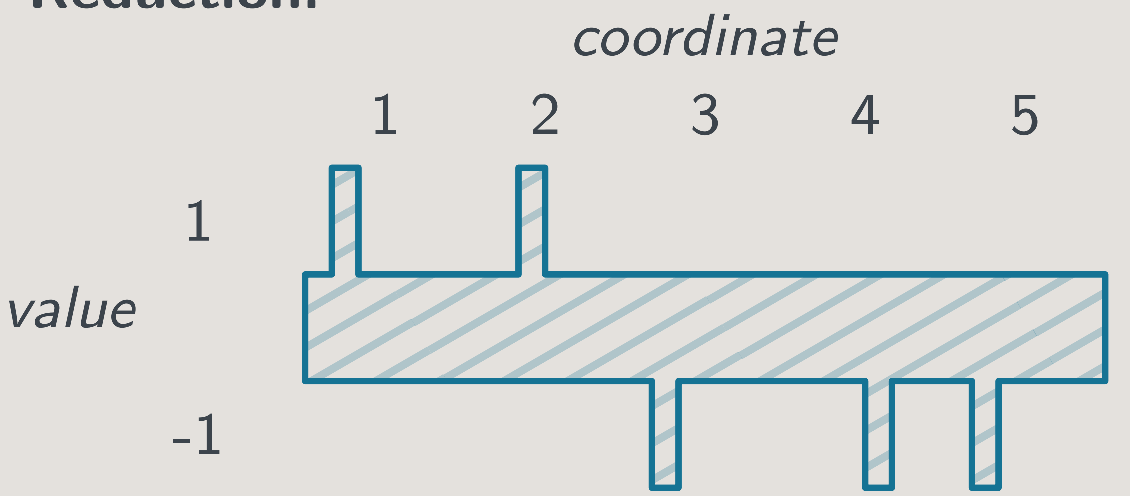
Idea: "hide coordindates" using random permutations

Input:

$x = (1, -1, 1, 1, -1)$

$y = (1, 1, -1, -1, -1)$

Reduction:



one piece

$P(\text{sample a coordinate point}) = \frac{1}{n}$
$P(\text{containment check succeeds}) = \frac{1}{n}$

↳ $O(n)$ queries for checking single bit

↳ $\Omega(n \cdot \frac{1}{\varepsilon^2})$ lower bound

Lower Bound

Restrictions

Lower Bound

Restrictions

can:

- use 3 queries of the model (volume, sample, contains)
- inspect coordinates of sampled points
- query arbitrary (not sampled!) points

Lower Bound

Restrictions

can:

- use 3 queries of the model (volume, sample, contains)
- inspect coordinates of sampled points
- query arbitrary (not sampled!) points

[Karp, Luby, Madras '89] only uses this!

Lower Bound

Restrictions

can:

- use 3 queries of the model (volume, sample, contains)
- inspect coordinates of sampled points
- query arbitrary (not sampled!) points

cannot:

- inspect coordinates of objects

[Karp, Luby, Madras '89] only uses this!

Improved **Upper Bound**

for

Klee's Measure Problem

Upper Bound

Insight from Lower Bound:

Have to use geometry of objects!

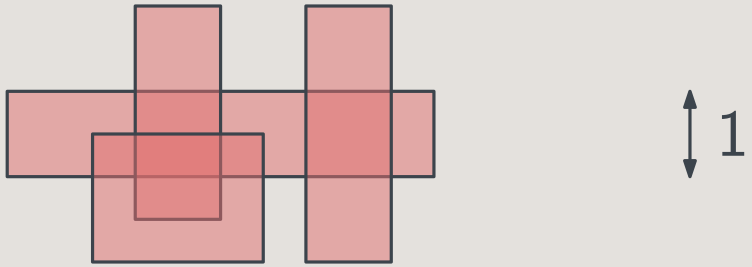
Upper Bound

Reminder:

Klee's Measure Problem

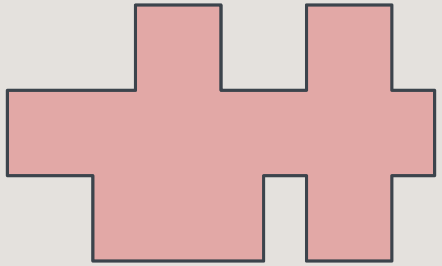
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Goal:

$(1+\varepsilon)$-approximate $\mathrm{Vol}(\bigcup_i O_i)$



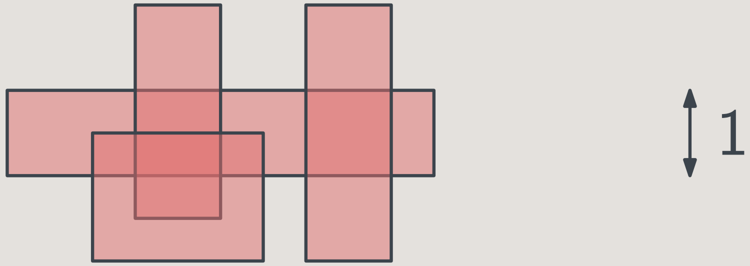
Upper Bound

Reminder:

Klee's Measure Problem

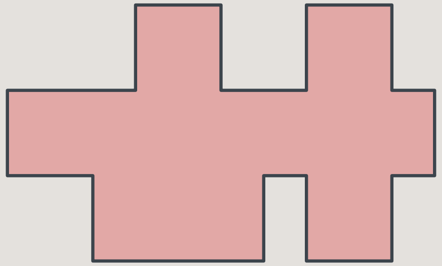
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Goal:

$(1+\varepsilon)$-approximate $\mathrm{Vol}(\bigcup_i O_i)$



Algorithm:

$\tilde{V} \leftarrow O(1)$-approx. of $\mathrm{Vol}(\bigcup_i O_i)$

$S \leftarrow p$-Sample with density $p \approx \frac{1}{\tilde{V}\varepsilon^2}$

return $|S|/p$

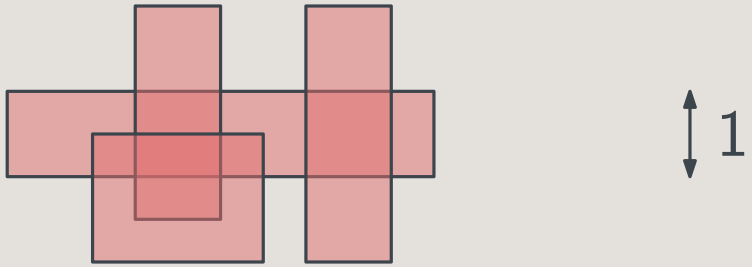
Upper Bound

Reminder:

Klee's Measure Problem

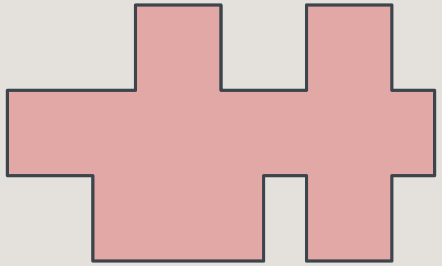
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Goal:

$(1+\varepsilon)$-approximate $\mathrm{Vol}(\bigcup_i O_i)$



Algorithm:

$\tilde{V} \leftarrow O(1)$-approx. of $\mathrm{Vol}(\bigcup_i O_i)$

$S \leftarrow p$-Sample with density $p \approx \frac{1}{\tilde{V}\varepsilon^2}$

return $|S|/p$

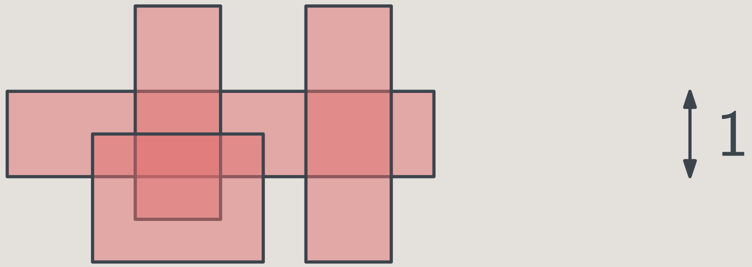
Upper Bound

Reminder:

Klee's Measure Problem

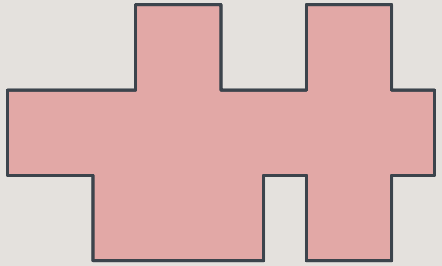
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Goal:

$(1+\varepsilon)$-approximate $\mathrm{Vol}(\bigcup_i O_i)$



Algorithm:

$\tilde{V} \leftarrow O(1)$-approx. of $\mathrm{Vol}(\bigcup_i O_i)$

$S \leftarrow p$-Sample with density $p \approx \frac{1}{\tilde{V}\varepsilon^2}$

return $|S|/p$

$(1+\varepsilon)$-approximation with good probability

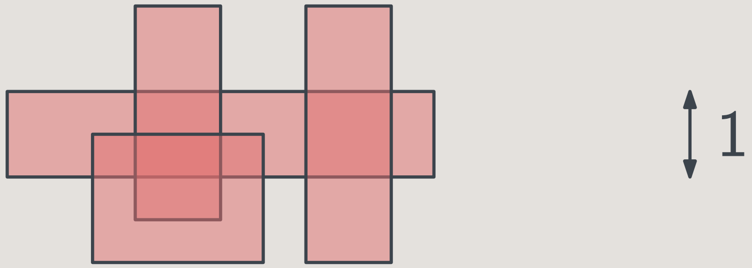
Upper Bound

Reminder:

Klee's Measure Problem

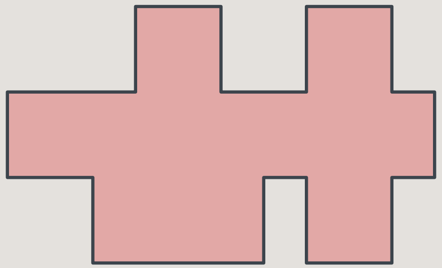
Given:

Boxes $O_1, \ldots, O_n \subset \mathbb{R}^d$



Goal:

$(1+\varepsilon)$-approximate $\mathrm{Vol}(\bigcup_i O_i)$



Algorithm:

$\tilde{V} \leftarrow O(1)$-approx. of $\mathrm{Vol}(\bigcup_i O_i)$

$S \leftarrow p$-Sample with density $p \approx \frac{1}{\tilde{V}\varepsilon^2}$

return $|S|/p$

$(1+\varepsilon)$-approximation with good probability

Goal:
Draw p-Sample from union of boxes.

Upper Bound

p-Sample: Similar Shapes

Upper Bound

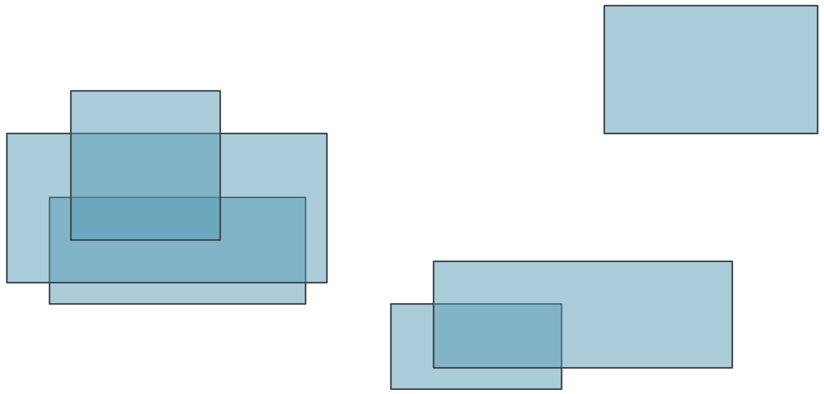
p-Sample: Similar Shapes

Class: All boxes have sizes in
$[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

Upper Bound

p-Sample: Similar Shapes

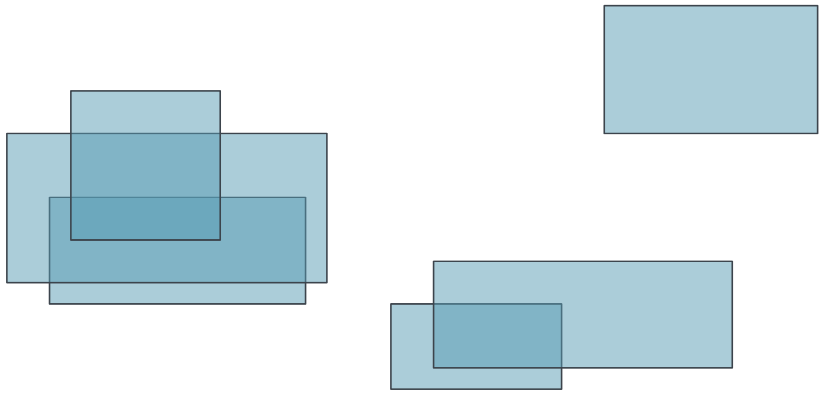
Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



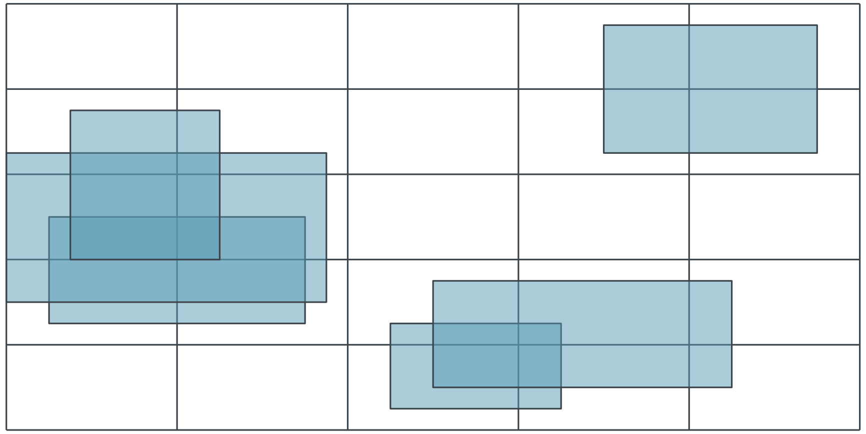
Algorithm:

build orthogonal range data structure

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

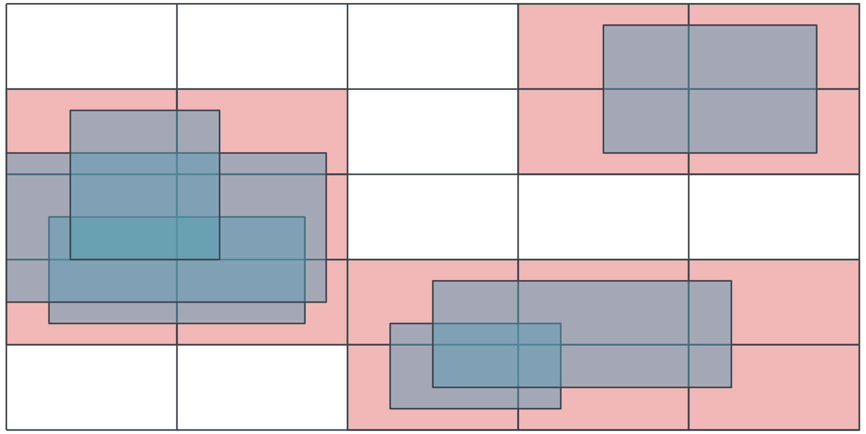
build orthogonal range data structure

create grid, mark active cells $G_1, \ldots G_T$

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

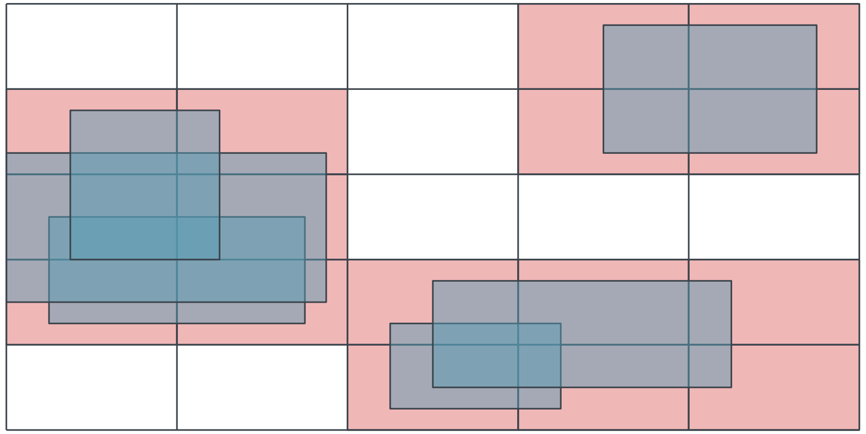
build orthogonal range data structure

create grid, mark active cells $G_1, \ldots G_T$

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

build orthogonal range data structure

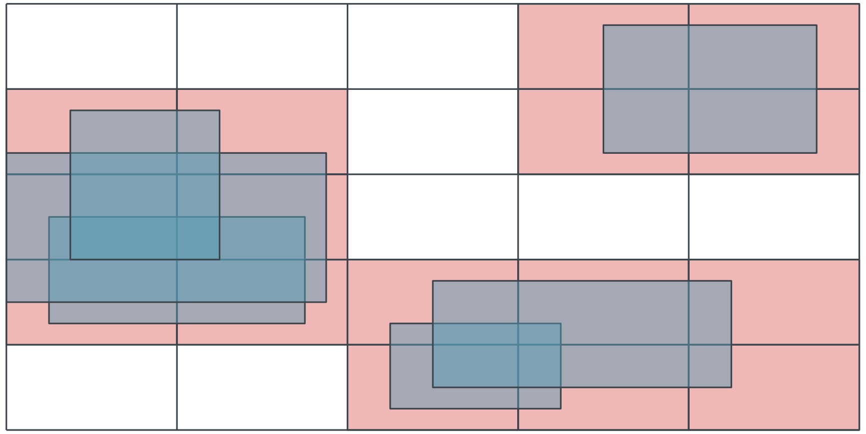
create grid, mark active cells $G_1, \ldots G_T$

$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



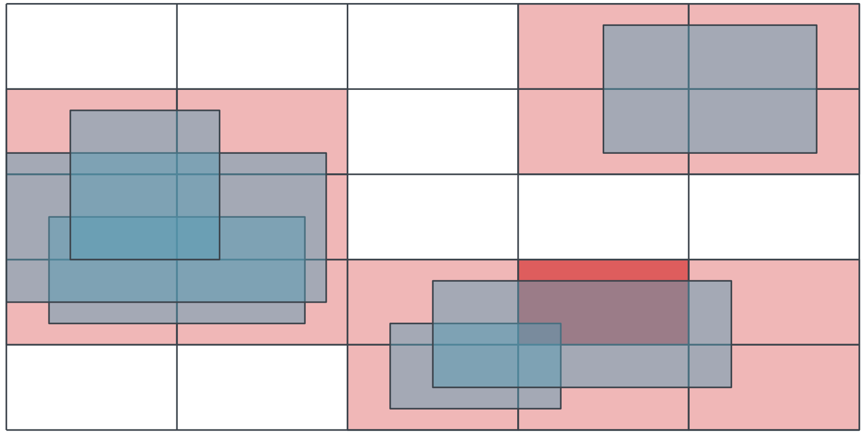
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

build orthogonal range data structure

create grid, mark active cells $G_1, \ldots G_T$

$K \sim$ Pois($p \cdot$ Volume($\bigcup_i G_i$))

for $i \in [K]$

$G \leftarrow$ random cell

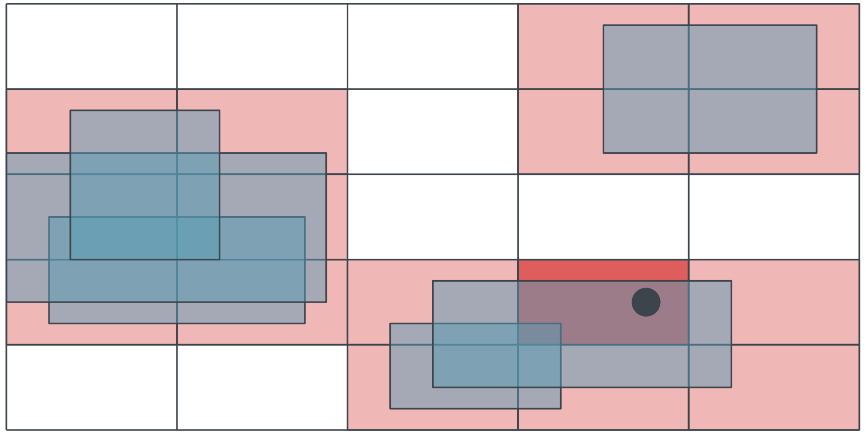
$q \leftarrow$ random point from G

if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



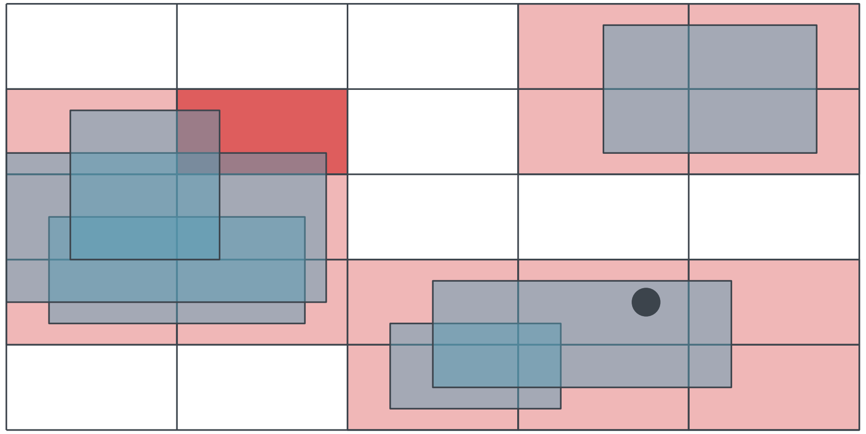
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

build orthogonal range data structure

create grid, mark active cells $G_1, \ldots G_T$

$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$

for $i \in [K]$

$G \leftarrow$ random cell

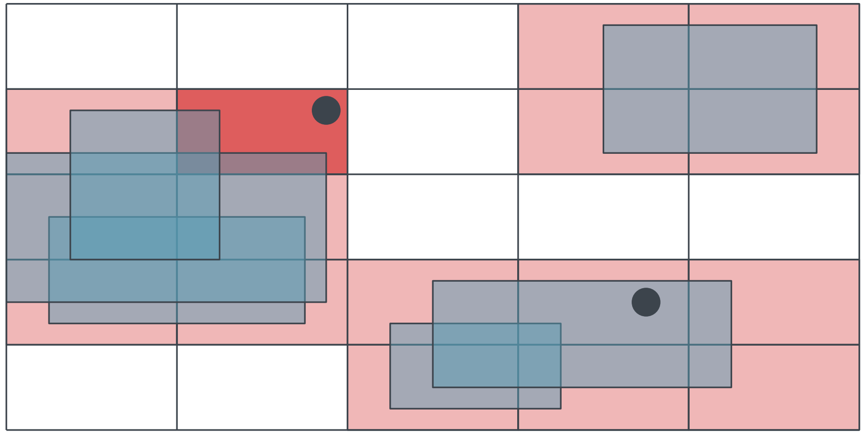
$q \leftarrow$ random point from G

if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

build orthogonal range data structure

create grid, mark active cells $G_1, \ldots G_T$

$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$

for $i \in [K]$

  $G \leftarrow$ random cell

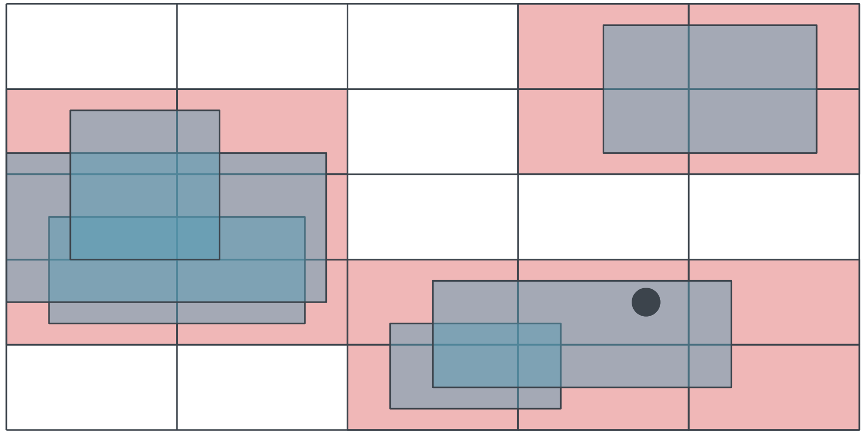
  $q \leftarrow$ random point from G

  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



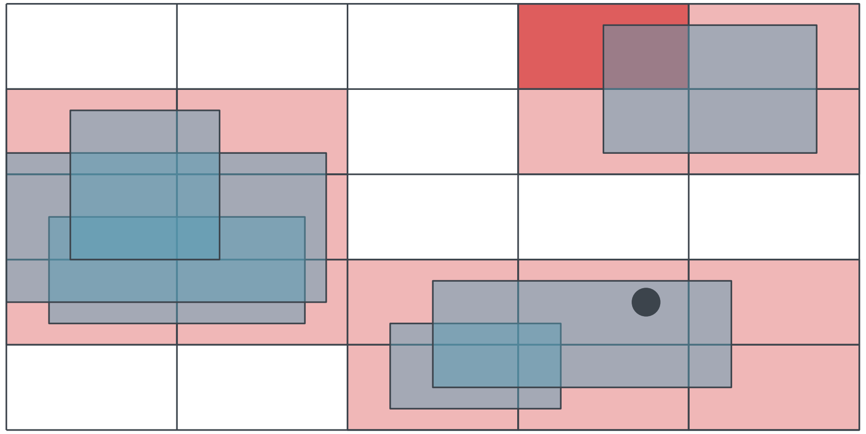
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim$ Pois($p \cdot$ Volume($\bigcup_i G_i$))
for $i \in [K]$
    $G \leftarrow$ random cell
    $q \leftarrow$ random point from G
    if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



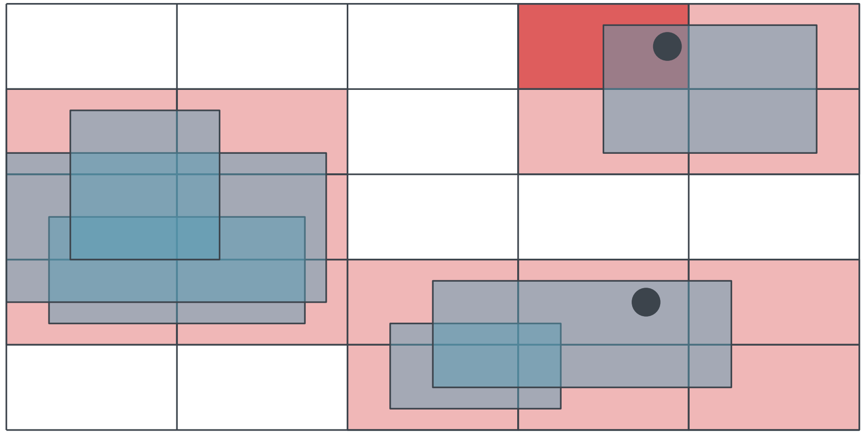
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



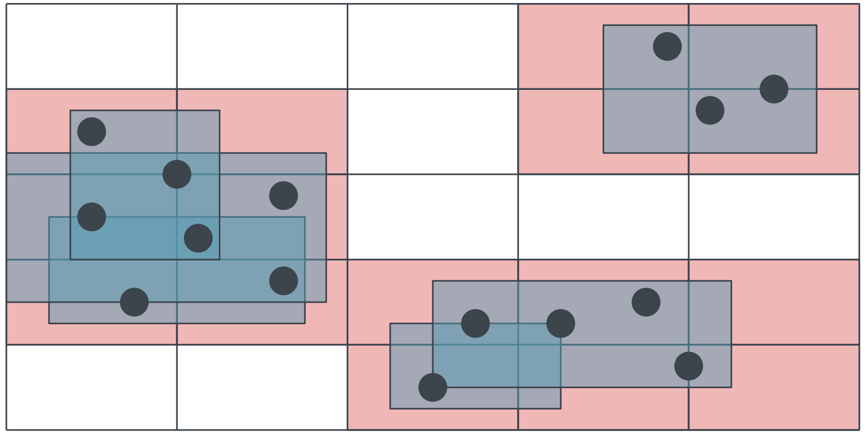
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



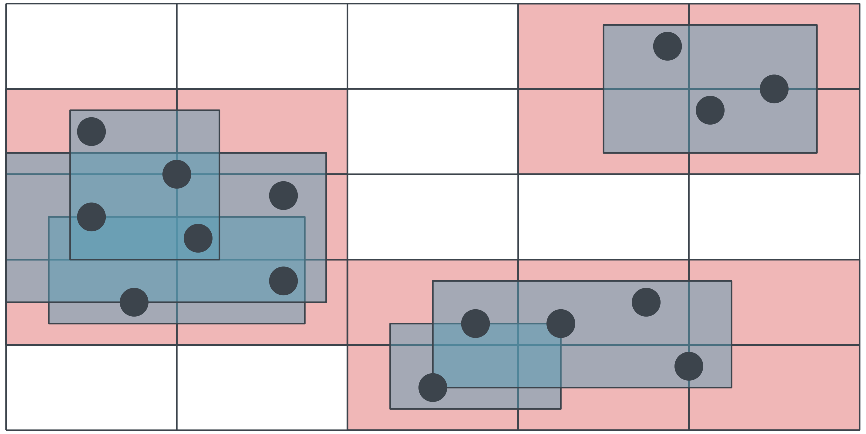
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



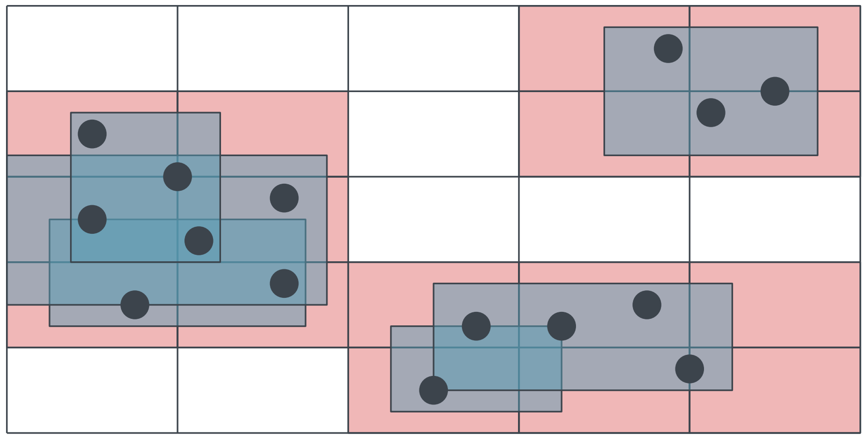
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples
return samples

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

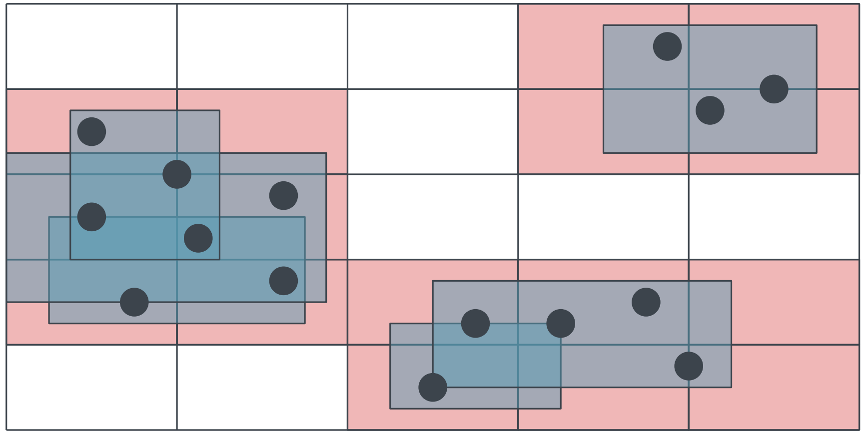
- build orthogonal range data structure
- create grid, mark active cells $G_1, \ldots G_T$
- $K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
- for $i \in [K]$
  - $G \leftarrow$ random cell
  - $q \leftarrow$ random point from G
  - if q contained in box: add q to samples
- return samples

Correctness?

Upper Bound

p-Sample: Similar Shapes

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Running Time?

$$O((n+K) \cdot \log^{O(d)} n)$$

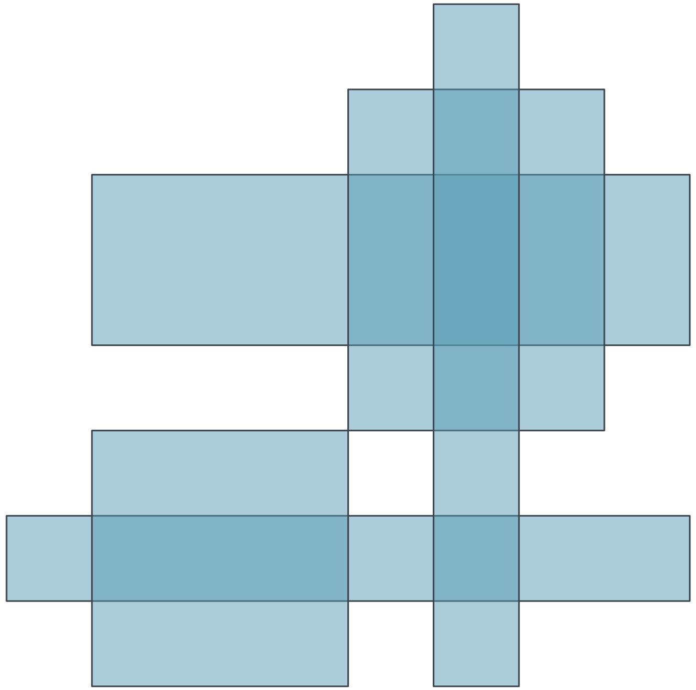
Algorithm:

build orthogonal range data structure
create grid, mark active cells $G_1, \ldots G_T$
$K \sim \text{Pois}(p \cdot \text{Volume}(\bigcup_i G_i))$
for $i \in [K]$
  $G \leftarrow$ random cell
  $q \leftarrow$ random point from G
  if q contained in box: add q to samples
return samples

Correctness?

Upper Bound

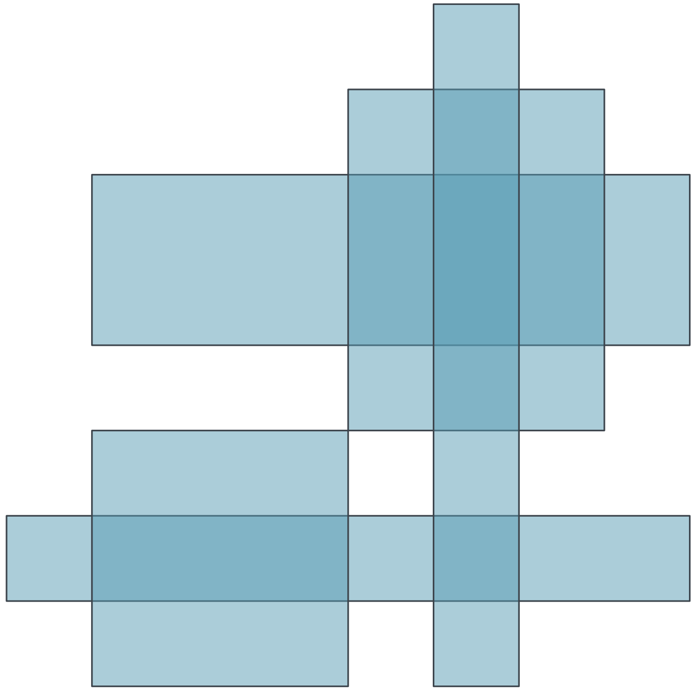
p-Sample: General Case



Algorithm:

Upper Bound

p-Sample: General Case



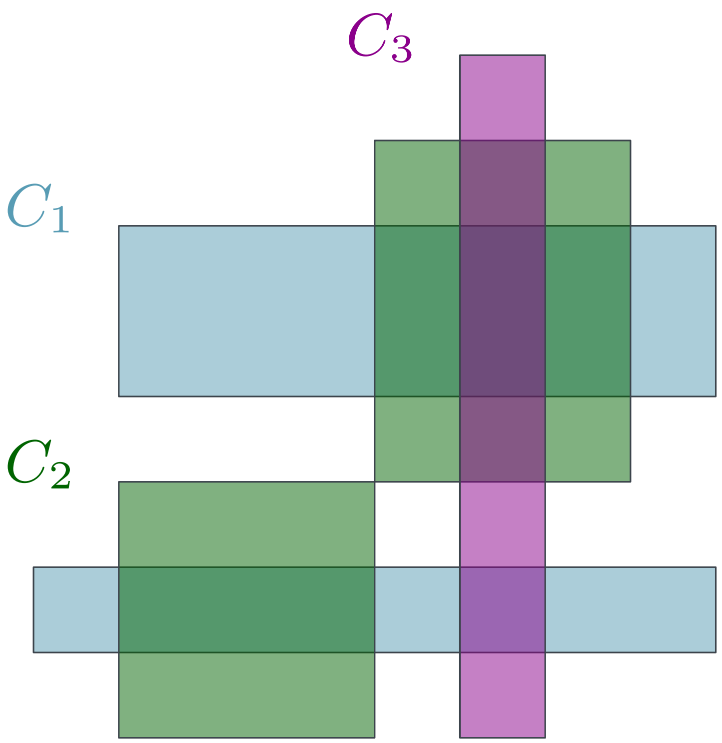
Algorithm:

build orthogonal range data structure

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



Algorithm:

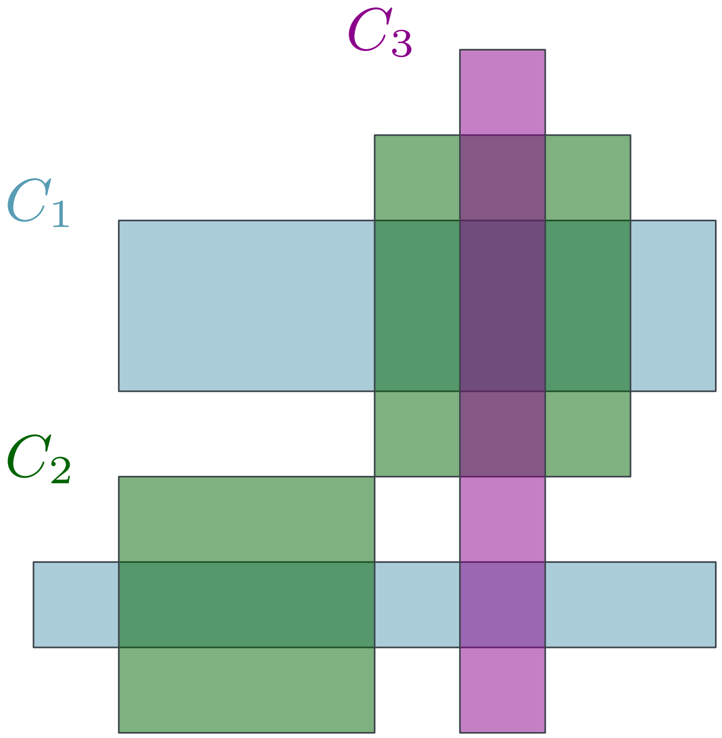
build orthogonal range data structure

partition input into classes $C_1, \ldots, C_m$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



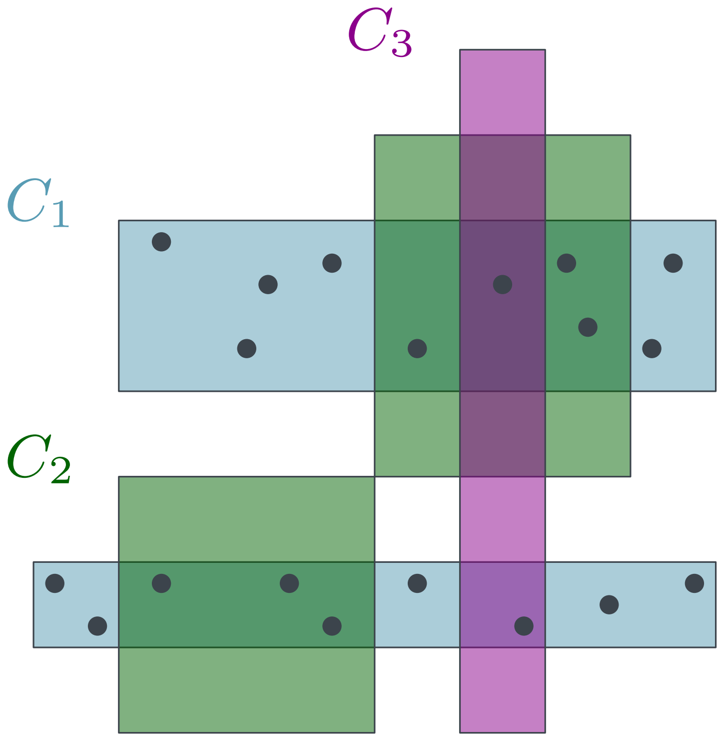
Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
$S_i \leftarrow p$-Sample from C_i
$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



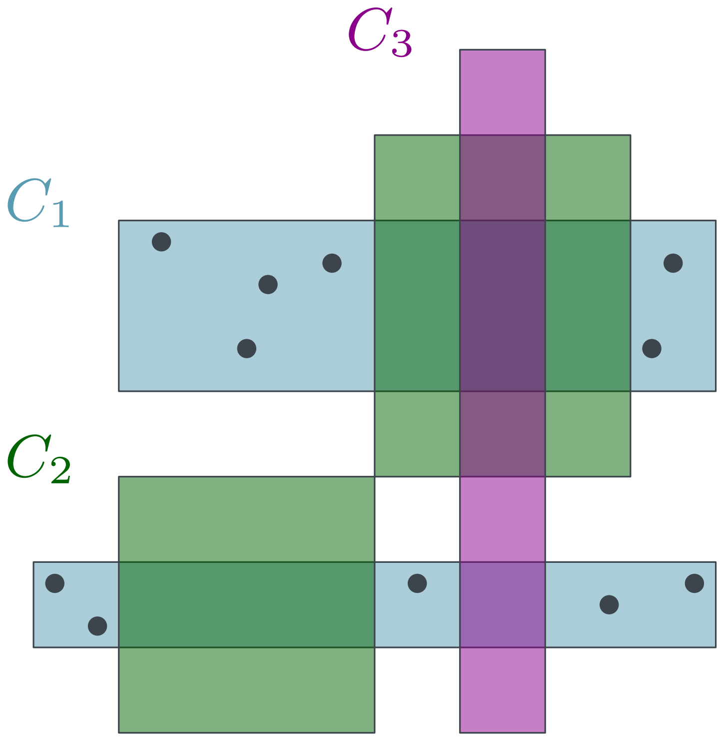
Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
$S_i \leftarrow p$-Sample from C_i
$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



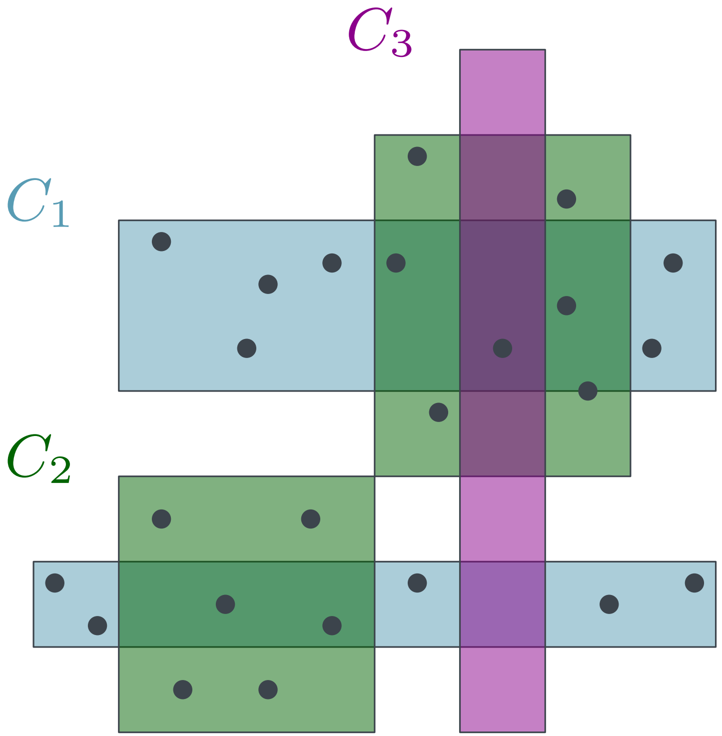
Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
$S_i \leftarrow p$-Sample from C_i
$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



Algorithm:

build orthogonal range data structure

partition input into classes $C_1, \ldots, C_m$

for each $i \in [m]$:

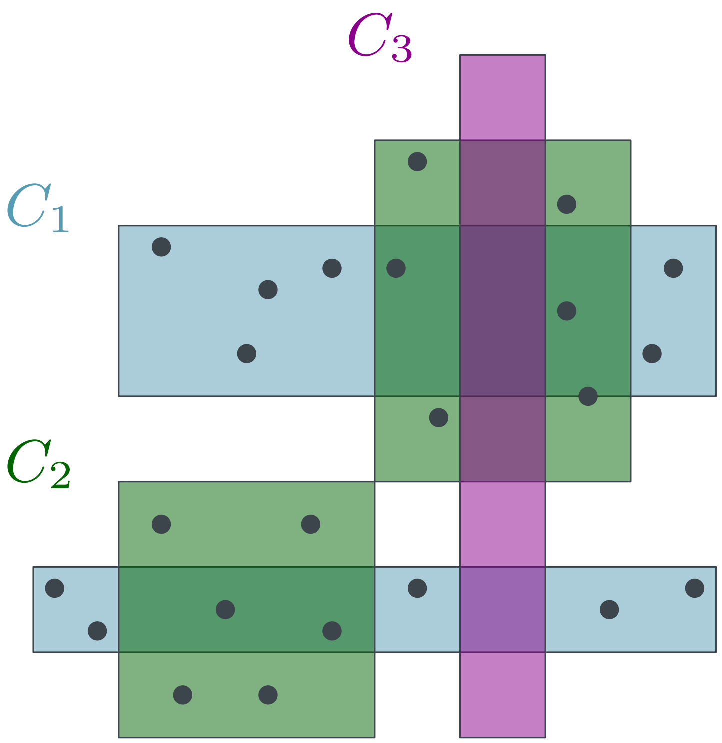
$S_i \leftarrow p$-Sample from C_i

$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



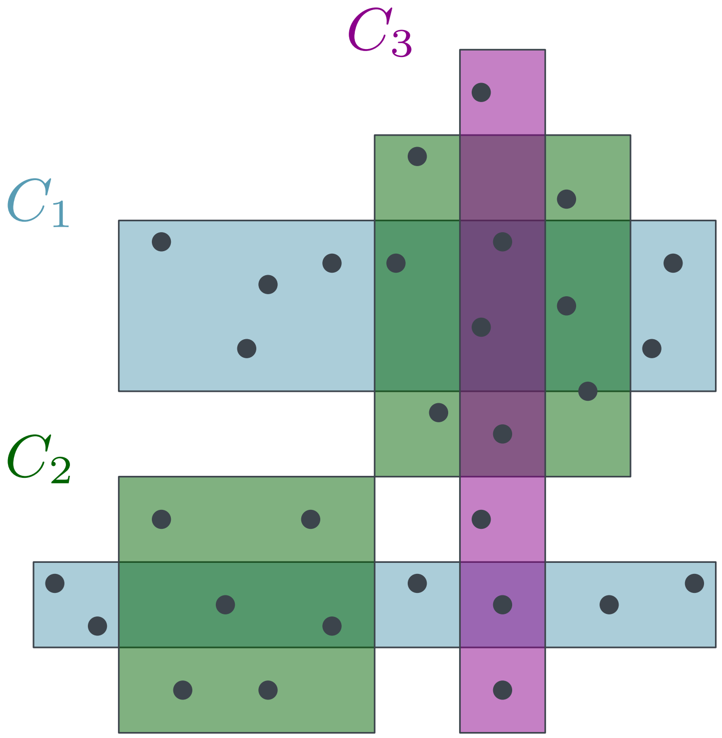
Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
  $S_i \leftarrow p$-Sample from C_i
  $S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



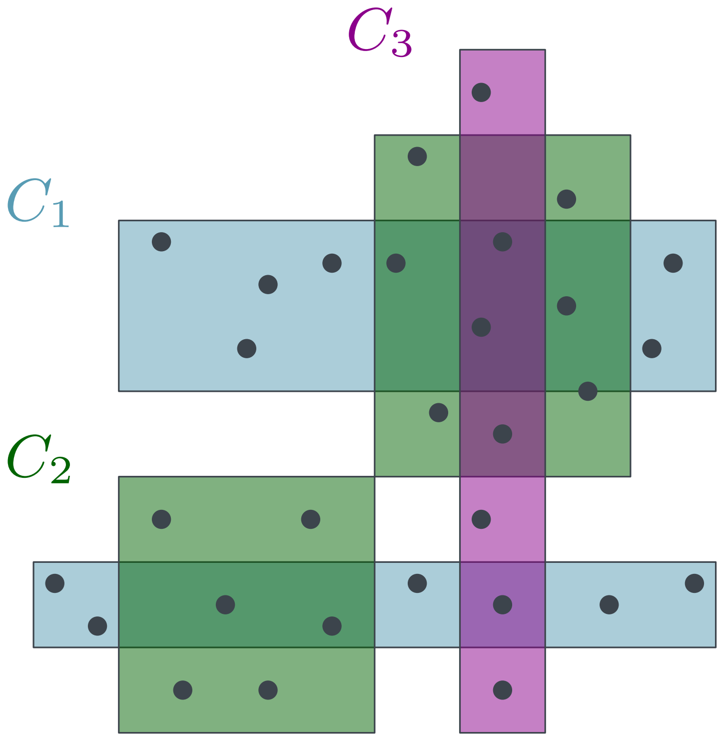
Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
    $S_i \leftarrow p$-Sample from C_i
    $S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



Algorithm:

build orthogonal range data structure

partition input into classes $C_1, \ldots, C_m$

for each $i \in [m]$:

$S_i \leftarrow p$-Sample from C_i

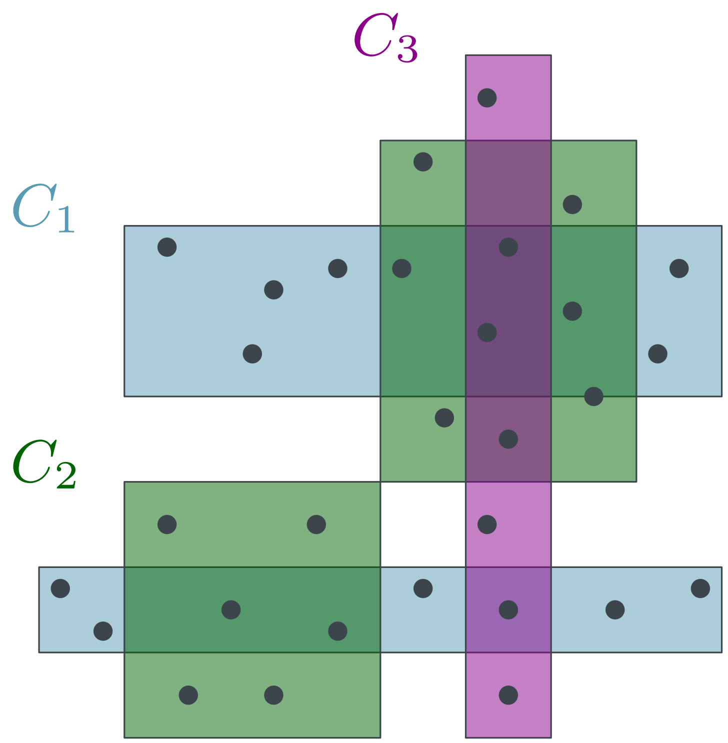
$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

output $\sum_{t=1}^{m} |S_t|/p$

Upper Bound

p-Sample: General Case

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.



Algorithm:

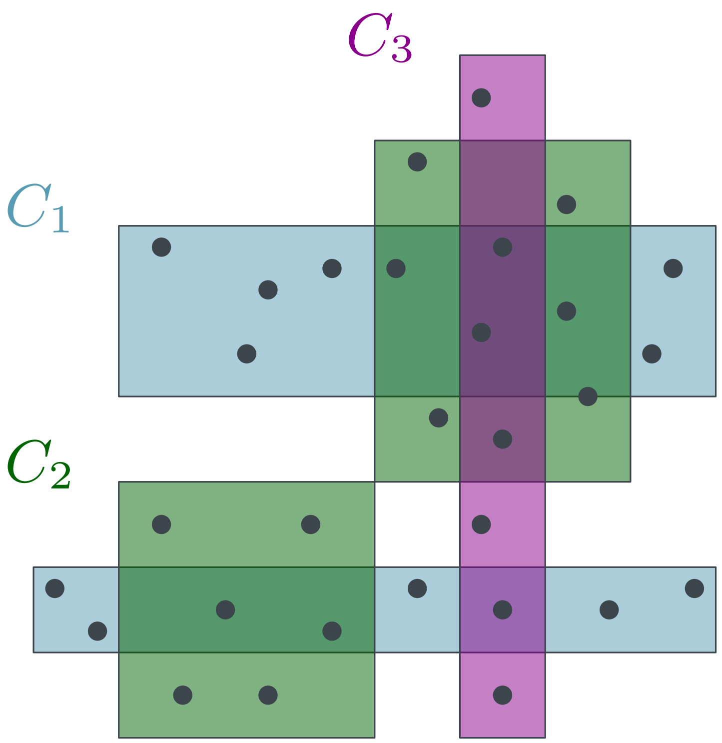
build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
  $S_i \leftarrow p$-Sample from C_i
  $S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$
output $\sum_{t=1}^{m} |S_t|/p$

Correctness?

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



Algorithm:

build orthogonal range data structure

partition input into classes $C_1, \ldots, C_m$

for each $i \in [m]$:

$S_i \leftarrow p$-Sample from C_i

$S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$

output $\sum_{t=1}^{m} |S_t|/p$

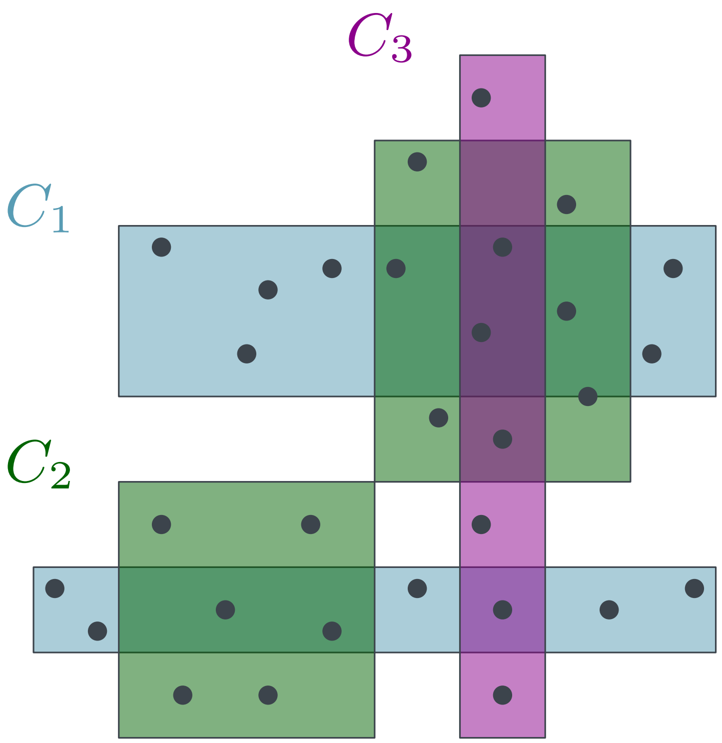
Correctness?

Running Time?

Upper Bound

Class: All boxes have sizes in $[s_1, 2s_1] \times [s_2, 2s_2] \times \cdots \times [s_d, 2s_d]$.

p-Sample: General Case



Algorithm:

build orthogonal range data structure
partition input into classes $C_1, \ldots, C_m$
for each $i \in [m]$:
  $S_i \leftarrow p$-Sample from C_i
  $S_i \leftarrow S_i \setminus (C_{i+1} \cup \cdots \cup C_m)$
output $\sum_{t=1}^{m} |S_t|/p$

Correctness?

Running Time?

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Proof: Let $C_j \ni q$, with $\mathrm{Vol}(C_j)$ maximal.

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Proof: Let $C_j \ni q$, with $\mathrm{Vol}(C_j)$ maximal.

#*similar* classes containing q: $\log^{O(d)} n$

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Proof: Let $C_j \ni q$, with $\mathrm{Vol}(C_j)$ maximal.

#*similar* classes containing q: $\log^{O(d)} n$
#*dissimilar* classes containing q: $O(1)$

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Proof: Let $C_j \ni q$, with $\mathrm{Vol}(C_j)$ maximal.

#*similar* classes containing q: $\log^{O(d)} n$
#*dissimilar* classes containing q: $O(1)$

Running Time?

$$O((n + |S|) \log^{O(d)} n)$$

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Upper Bound

p-Sample: General Case

Lemma

A random $q \in \bigcup_i C_i$ is contained in at most $\log^{O(d)} n$ classes in expectation.

Proof: Let $C_j \ni q$, with $\mathrm{Vol}(C_j)$ maximal.

#*similar* classes containing q: $\log^{O(d)} n$
#*dissimilar* classes containing q: $O(1)$

Running Time?

$$O((n + |S|) \log^{O(d)} n)$$

$|S| \leftarrow 1/\varepsilon^2$

Definition:

Classes C_i, C_j are **similar** if all side lengths are within a factor n^3.

Observations:

Each class C_i is similar to at most $\log^{O(d)} n$ other classes.

If C_i, C_j are dis-similar, then

$$\mathrm{Vol}(C_i \cap C_j) \leq \frac{\mathrm{Vol}(C_i \cup C_j)}{n}.$$

Conclusion

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

Conclusion

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

Klee's Measure Problem

"can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time"

Conclusion

Union Volume Estimation

"requires $\Omega(n/\varepsilon^2)$ time"

Klee's Measure Problem

"can be solved in $\tilde{O}(n + \frac{1}{\varepsilon^2})$ time"

Questions?