

Approximating Klee's Measure and a Lower Bound for **Union Volume Estimation**

Karl Bringmann

André Nusser

Kasper Green Larsen

Eva Rotenberg

Yanheng Wang

Approximating Klee's Measure and
a Lower Bound for **Union Volume Estimation**

$$\text{vol} \left(\bigcup_{i=1}^n X_i \right) \approx ?$$

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

$$\text{vol} \left(\bigcup_{i=1}^n X_i \right) \approx ?$$

boxes polytopes ...

0 Oracle Model

$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$
algorithm

oracle X_1, \dots, X_n

0 Oracle Model

$(1 \pm \varepsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle X_1, \dots, X_n

0 Oracle Model

$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$

algorithm

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle X_1, \dots, X_n

Karp-Luby-Madras: $O\left(\frac{n}{\varepsilon^2}\right)$ time algorithm!

0 Oracle Model

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle X_1, \dots, X_n

Karp-Luby-Madras: $O\left(\frac{n}{\varepsilon^2}\right)$ time algorithm!

We show: $\tilde{O}\left(n + \frac{1}{\varepsilon^2}\right)$ time possible when X_i 's are boxes

0 Oracle Model

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle X_1, \dots, X_n

Karp-Luby-Madras: $O\left(\frac{n}{\varepsilon^2}\right)$ time algorithm!

We show: $\tilde{O}\left(n + \frac{1}{\varepsilon^2}\right)$ time possible when X_i 's are boxes

$\Omega\left(\frac{n}{\varepsilon^2}\right)$ time necessary in general

0 Oracle Model

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle X_1, \dots, X_n

Karp-Luby-Madras: $O\left(\frac{n}{\varepsilon^2}\right)$ time algorithm!

1

We show: $\tilde{O}\left(n + \frac{1}{\varepsilon^2}\right)$ time possible when X_i 's are boxes

2

$\Omega\left(\frac{n}{\varepsilon^2}\right)$ time necessary in general

3

1 Ideas Behind KLM

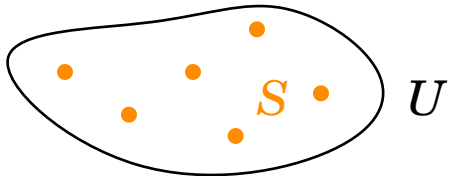
1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if
$$\mathbf{E} |S \cap U'| = p \operatorname{vol}(U'), \quad \forall U' \subseteq U$$

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E}|S \cap U'| = p \operatorname{vol}(U'), \quad \forall U' \subseteq U$$

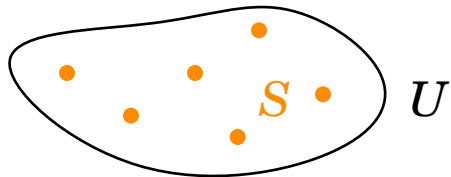


density p everywhere

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E}|S \cap U'| = p \operatorname{vol}(U'), \quad \forall U' \subseteq U$$



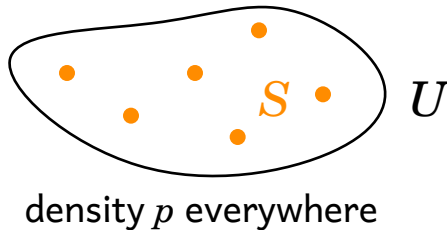
density p everywhere

[draw $N \sim \operatorname{Pois}(p \operatorname{vol}(U))$
repeat N times:
sample a random $x \in U$]

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \text{vol}(U'), \quad \forall U' \subseteq U$$



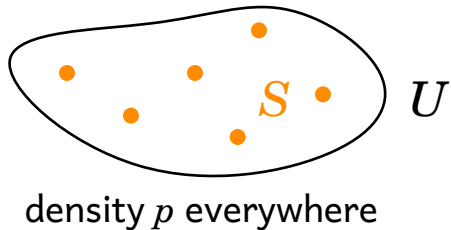
[draw $N \sim \text{Pois}(p \text{vol}(U))$
repeat N times:
sample a random $x \in U$]

Why care?

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \text{vol}(U'), \quad \forall U' \subseteq U$$



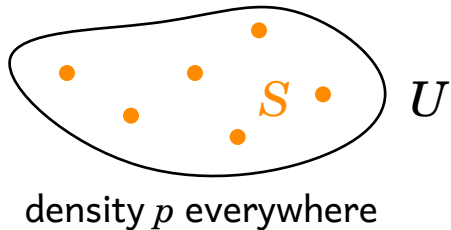
[draw $N \sim \text{Pois}(p \text{vol}(U))$
repeat N times:
sample a random $x \in U$]

Why care? Suppose can p -sample from $U = \bigcup_{i=1}^n X_i$

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \text{vol}(U'), \quad \forall U' \subseteq U$$



[draw $N \sim \text{Pois}(p \text{vol}(U))$
repeat N times:
sample a random $x \in U$]

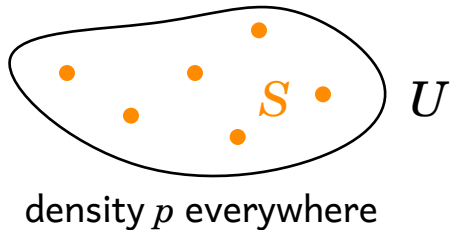
Why care? Suppose can p -sample from $U = \bigcup_{i=1}^n X_i$

$$\text{Exponential guess } \frac{10}{\varepsilon^2 \text{vol}(U)} \leq p \leq \frac{20}{\varepsilon^2 \text{vol}(U)}$$

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \text{vol}(U'), \quad \forall U' \subseteq U$$



[draw $N \sim \text{Pois}(p \text{vol}(U))$
repeat N times:
sample a random $x \in U$]

Why care? Suppose can p -sample from $U = \bigcup_{i=1}^n X_i$

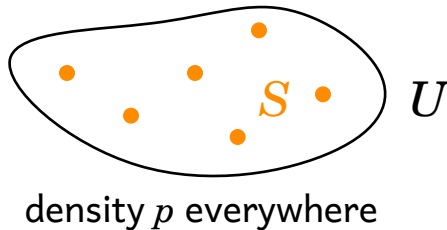
$$\text{Exponential guess } \frac{10}{\varepsilon^2 \text{vol}(U)} \leq p \leq \frac{20}{\varepsilon^2 \text{vol}(U)}$$

$$|S| \sim \text{Pois}(p \text{vol}(U))$$

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \text{ vol}(U'), \quad \forall U' \subseteq U$$



[draw $N \sim \text{Pois}(p \text{ vol}(U))$
repeat N times:
sample a random $x \in U$]

Why care? Suppose can p -sample from $U = \bigcup_{i=1}^n X_i$

$$\text{Exponential guess } \frac{10}{\varepsilon^2 \text{ vol}(U)} \leq p \leq \frac{20}{\varepsilon^2 \text{ vol}(U)}$$

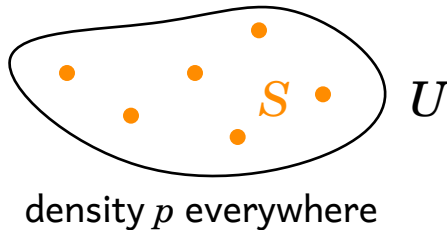
$$|S| \sim \text{Pois}(p \text{ vol}(U))$$

$$\mathbf{P} \left[\frac{|S|}{p} \in (1 \pm \varepsilon) \text{ vol}(U) \right] \geq \frac{9}{10}$$

1 Ideas Behind KLM

Definition: Random subset $S \subset U$ is a p -sample of U if

$$\mathbf{E} |S \cap U'| = p \operatorname{vol}(U'), \quad \forall U' \subseteq U$$



[draw $N \sim \operatorname{Pois}(p \operatorname{vol}(U))$
repeat N times:
sample a random $x \in U$]

Why care? Suppose can p -sample from $U = \bigcup_{i=1}^n X_i$

$$\text{Exponential guess } \frac{10}{\varepsilon^2 \operatorname{vol}(U)} \leq p \leq \frac{20}{\varepsilon^2 \operatorname{vol}(U)}$$

$$|S| \sim \operatorname{Pois}(p \operatorname{vol}(U))$$

$$\mathbf{P} \left[\frac{|S|}{p} \in (1 \pm \varepsilon) \operatorname{vol}(U) \right] \geq \frac{9}{10}$$

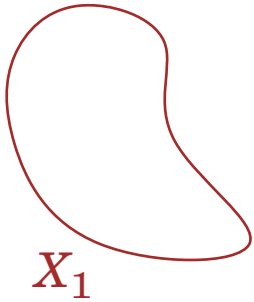
$$\mathbf{E}[|S|] \leq \frac{20}{\varepsilon^2}$$

1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$

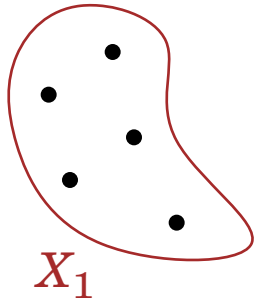
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



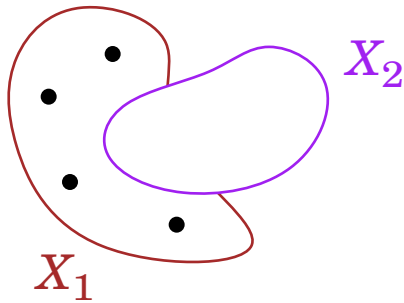
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



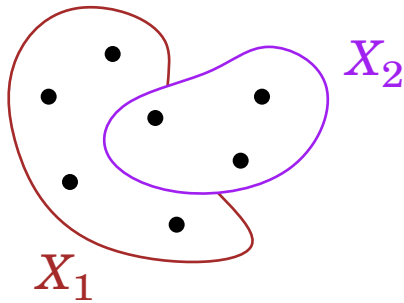
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



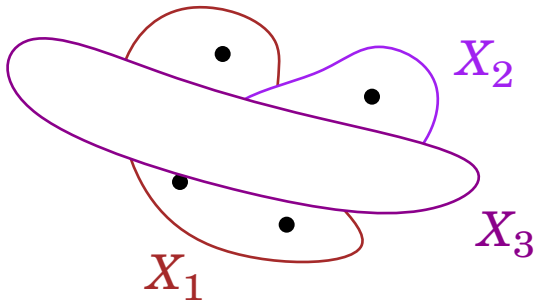
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



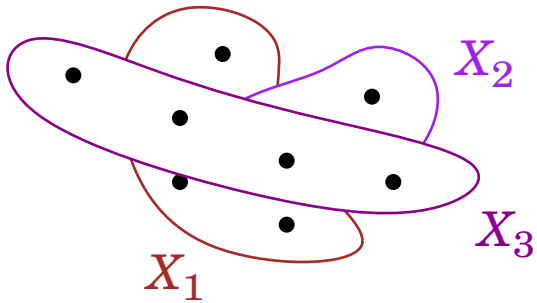
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



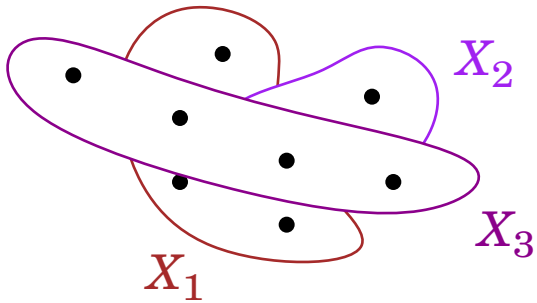
1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$



1 Ideas Behind KLM

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$

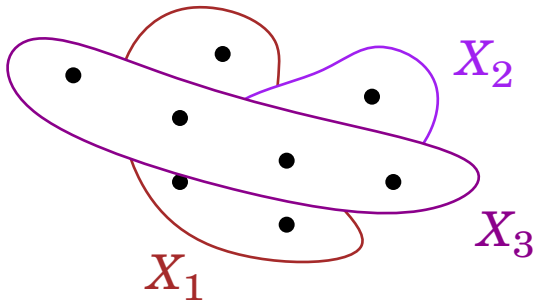


$$\left[\begin{array}{l} S := \emptyset \\ \text{for } i = 1, \dots, n: \\ \quad S := S \setminus X_i \\ \quad S := S \cup (p\text{-sample of } X_i) \end{array} \right]$$

2 Improved Algorithm for Boxes

axis-aligned; coordinates given

Remaining task: p -sample from $U := \bigcup_{i=1}^n X_i$

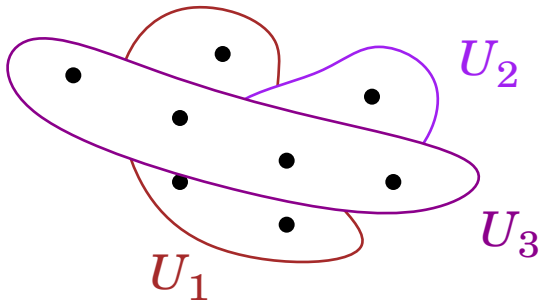


```
[  
   $S := \emptyset$   
  for  $i = 1, \dots, n$ :  
     $S := S \setminus X_i$   
     $S := S \cup (p\text{-sample of } X_i)$   
]
```

2 Improved Algorithm for Boxes

axis-aligned; coordinates given

Remaining task: p -sample from $U := \bigcup_{i=1}^g U_i$

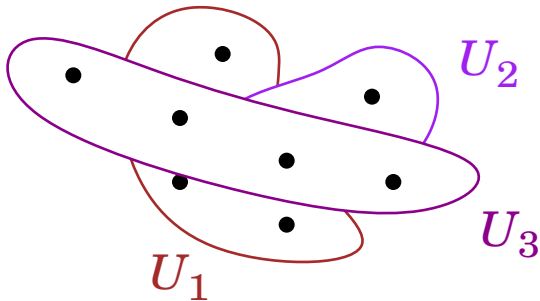


$$\left[\begin{array}{l} S := \emptyset \\ \text{for } i = 1, \dots, g: \\ \quad S := S \setminus U_i \\ \quad S := S \cup (p\text{-sample of } U_i) \end{array} \right]$$

2 Improved Algorithm for Boxes

axis-aligned; coordinates given

Remaining task: p -sample from $U := \bigcup_{i=1}^g U_i$?

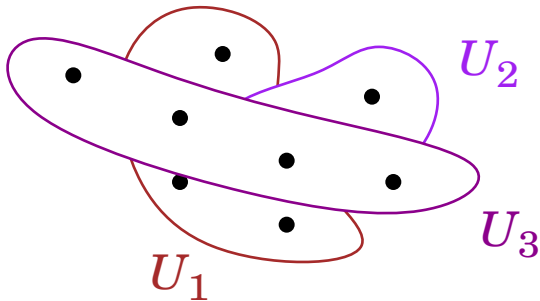


$$\left[\begin{array}{l} S := \emptyset \\ \text{for } i = 1, \dots, g: \\ \quad S := S \setminus U_i \text{ ?} \\ \quad S := S \cup (p\text{-sample of } U_i) \text{ ?} \end{array} \right]$$

2 Improved Algorithm for Boxes

axis-aligned; coordinates given

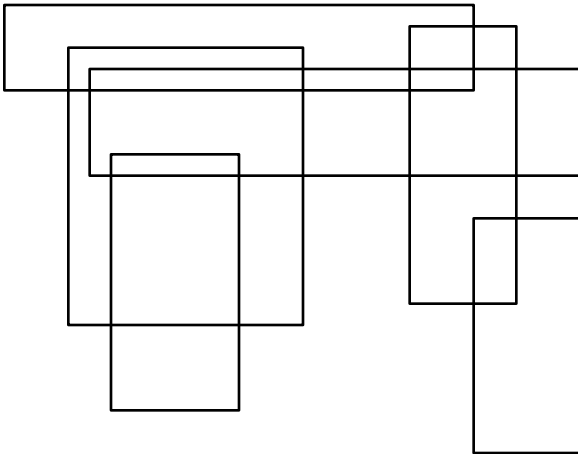
Remaining task: p -sample from $U := \bigcup_{i=1}^g U_i$?



$$\left[\begin{array}{l} S := \emptyset \\ \text{for } i = 1, \dots, g: \\ \quad S := S \setminus U_i \\ \quad S := S \cup (p\text{-sample of } U_i) ? \end{array} \right]$$

2 Improved Algorithm for Boxes

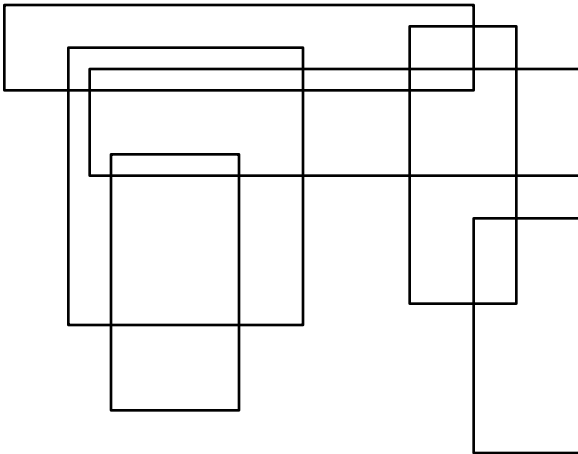
axis-aligned; coordinates given



2 Improved Algorithm for Boxes

axis-aligned; coordinates given

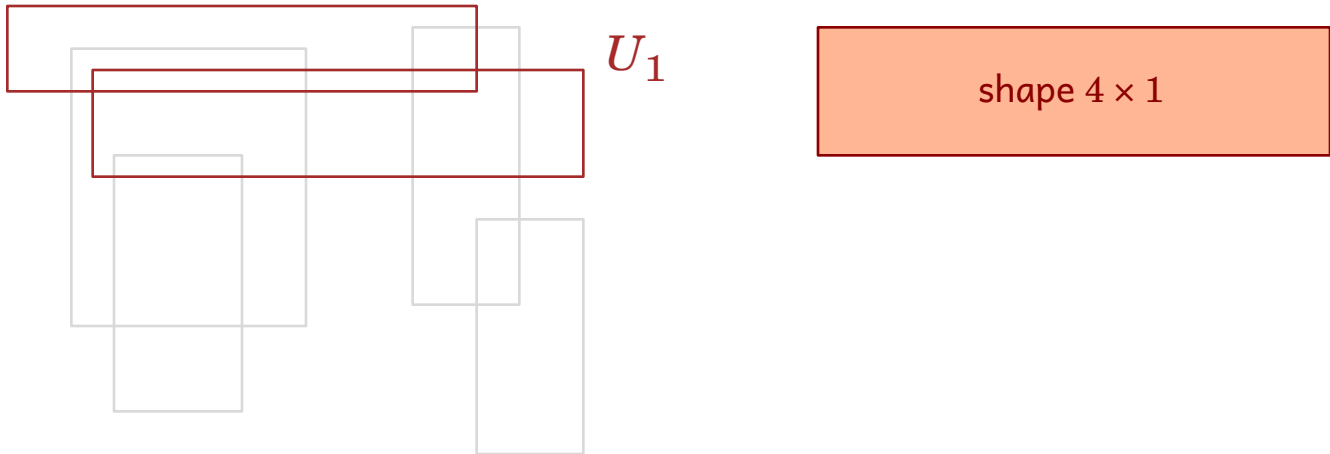
Group boxes by shapes (round to power of two)



2 Improved Algorithm for Boxes

axis-aligned; coordinates given

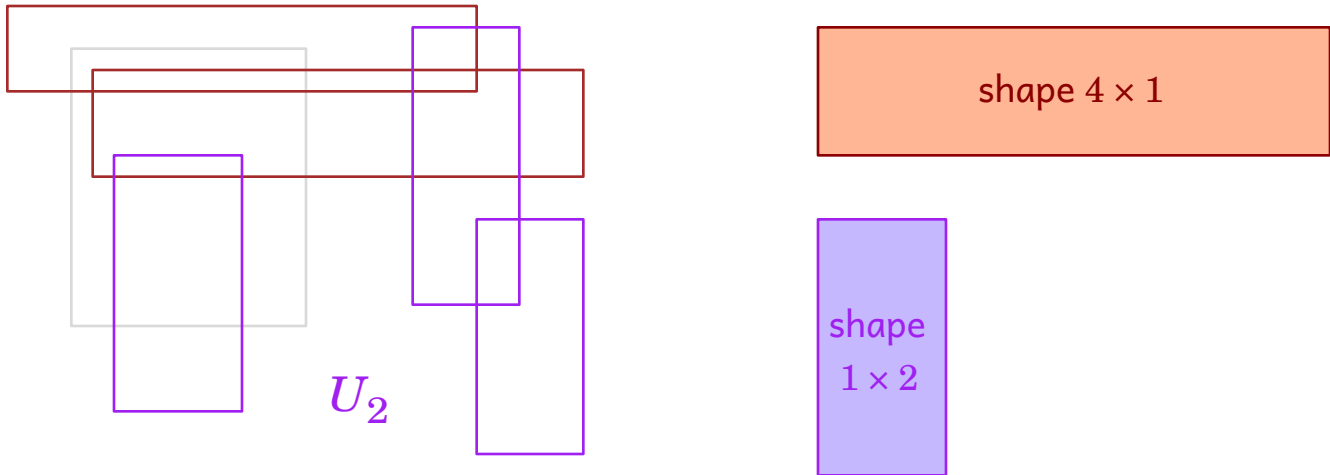
Group boxes by shapes (round to power of two)



2 Improved Algorithm for Boxes

axis-aligned; coordinates given

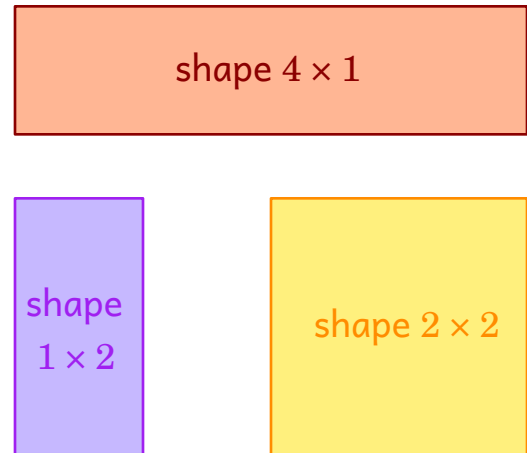
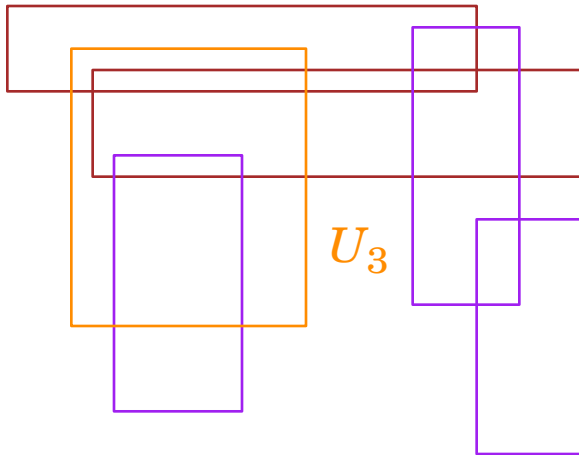
Group boxes by shapes (round to power of two)



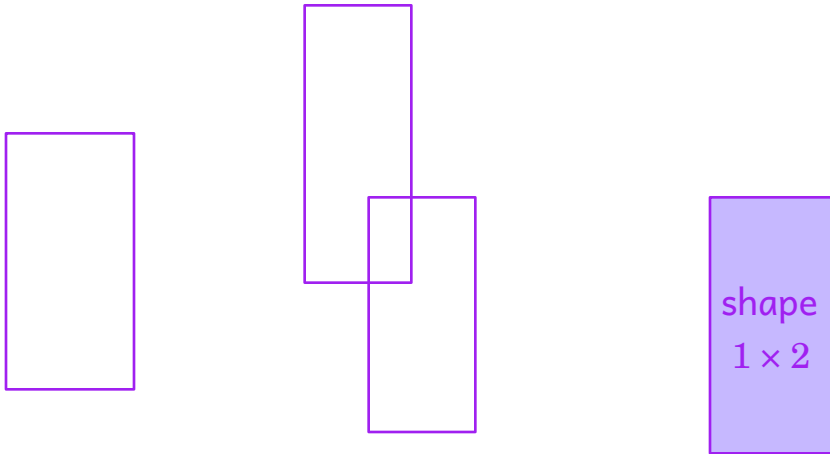
2 Improved Algorithm for Boxes

axis-aligned; coordinates given

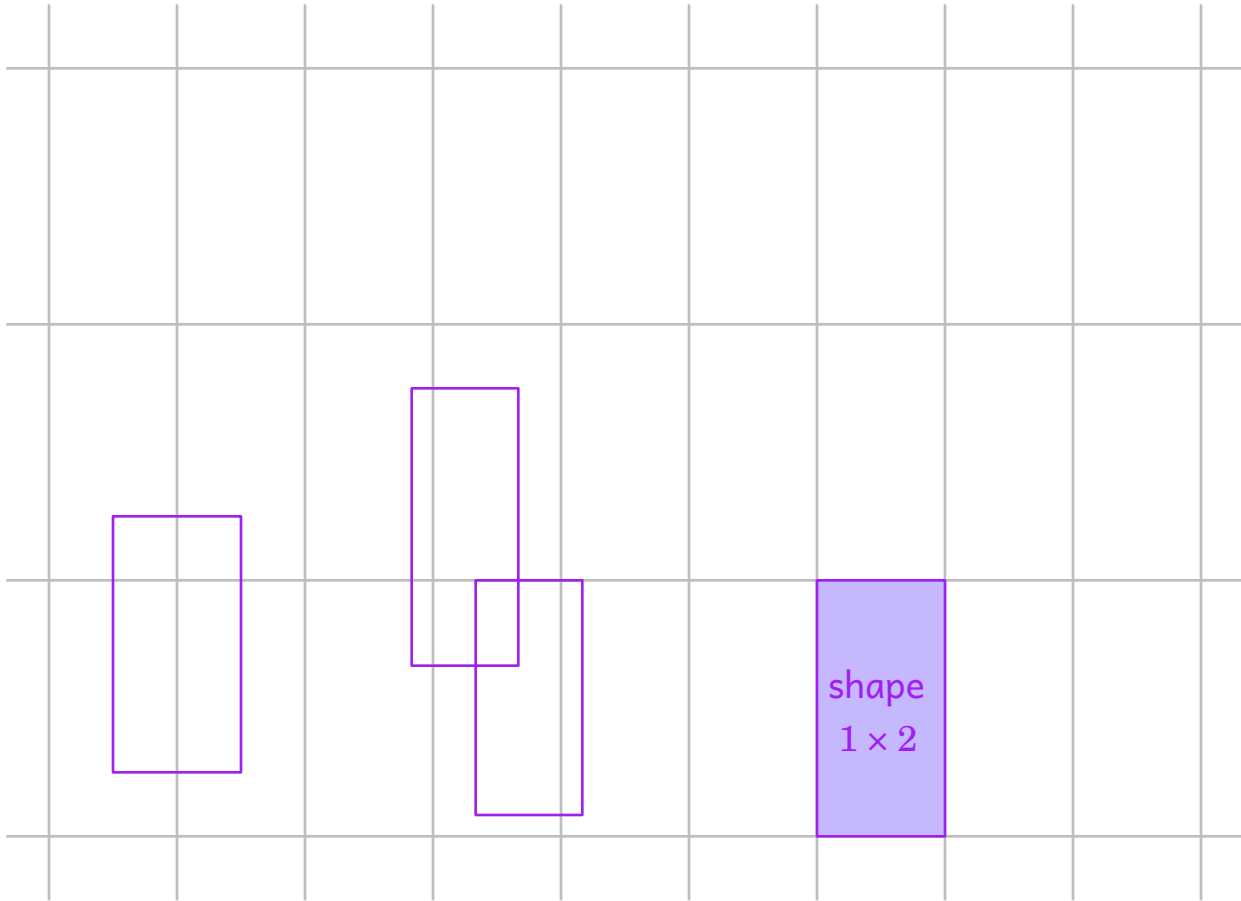
Group boxes by shapes (round to power of two)



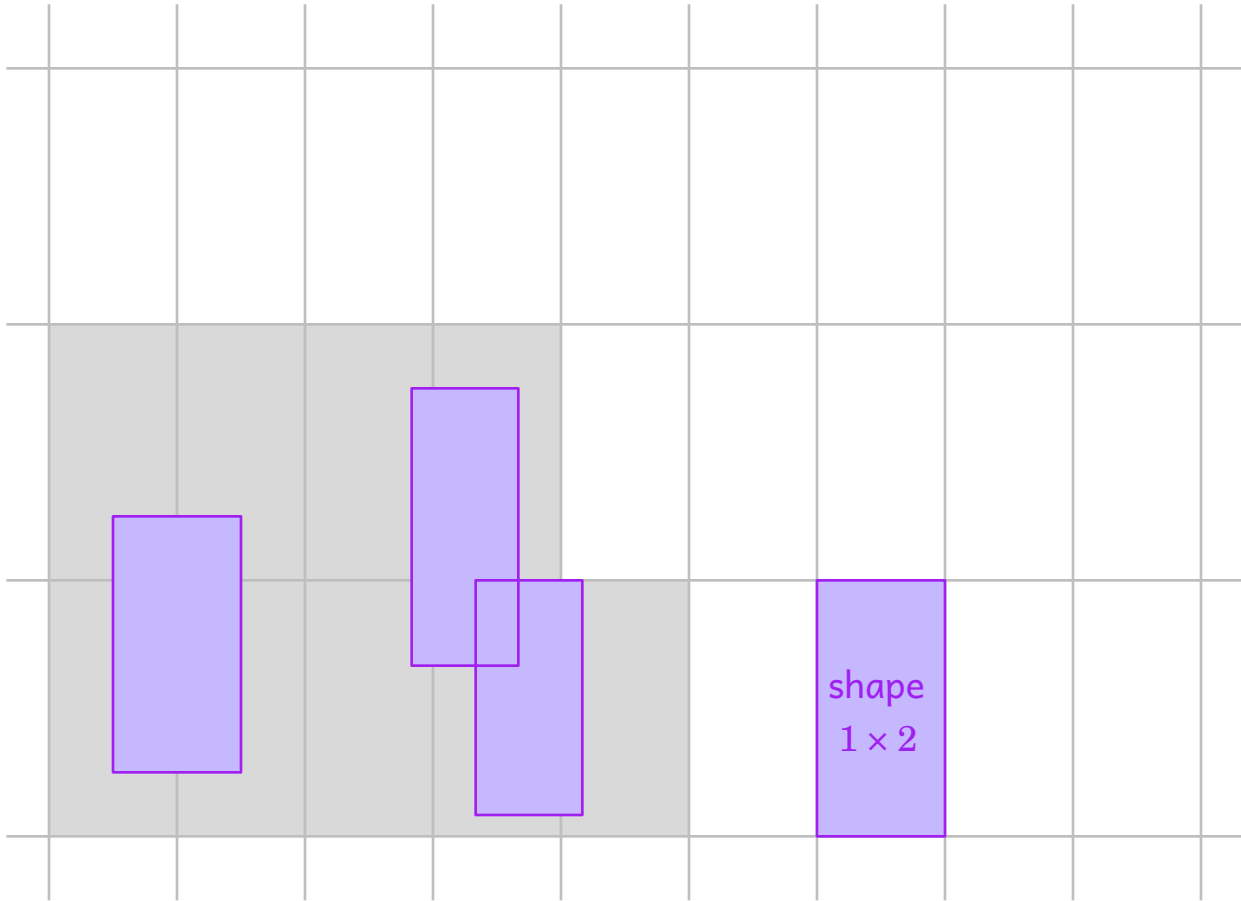
2 Improved Algorithm for Boxes



2 Improved Algorithm for Boxes

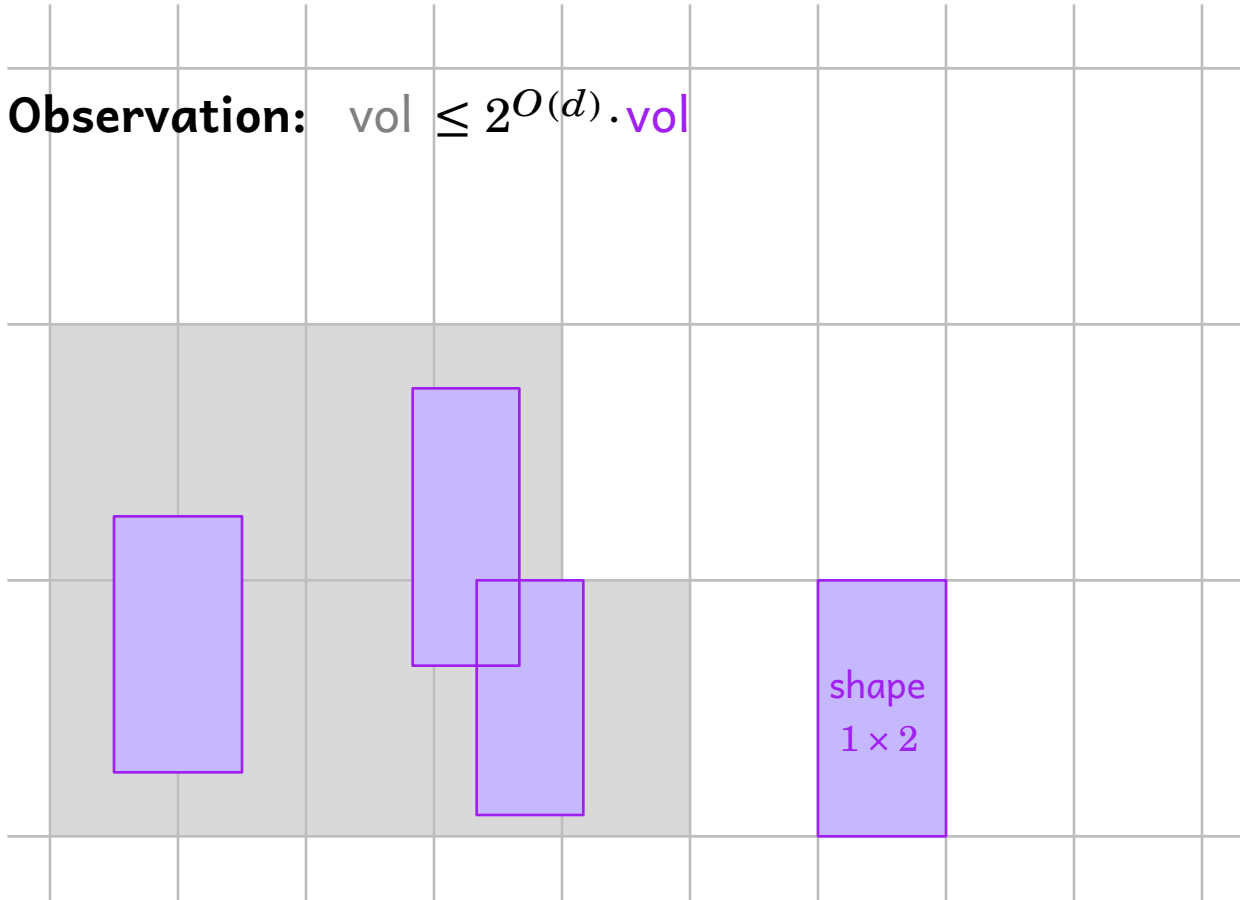


2 Improved Algorithm for Boxes



2 Improved Algorithm for Boxes

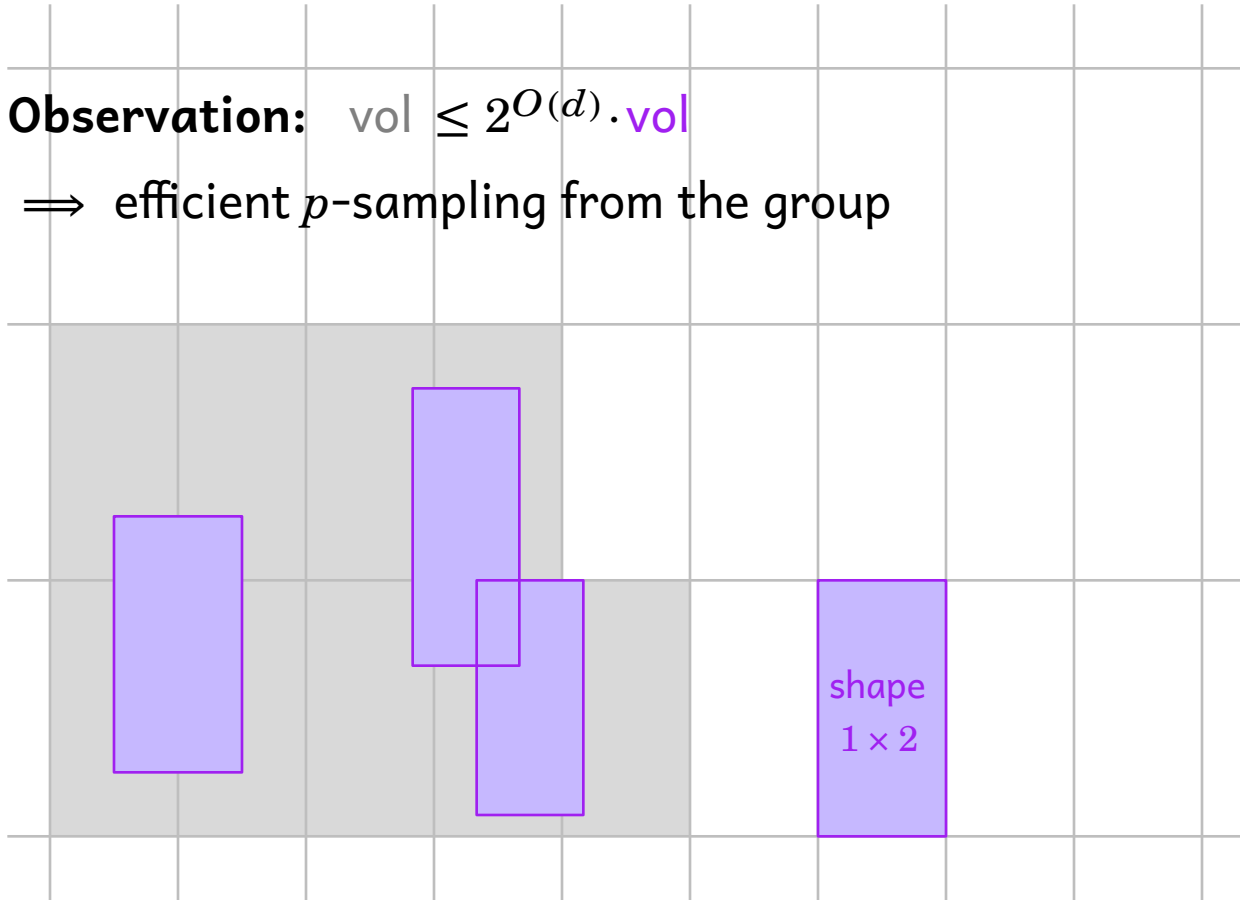
Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$



2 Improved Algorithm for Boxes

Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$

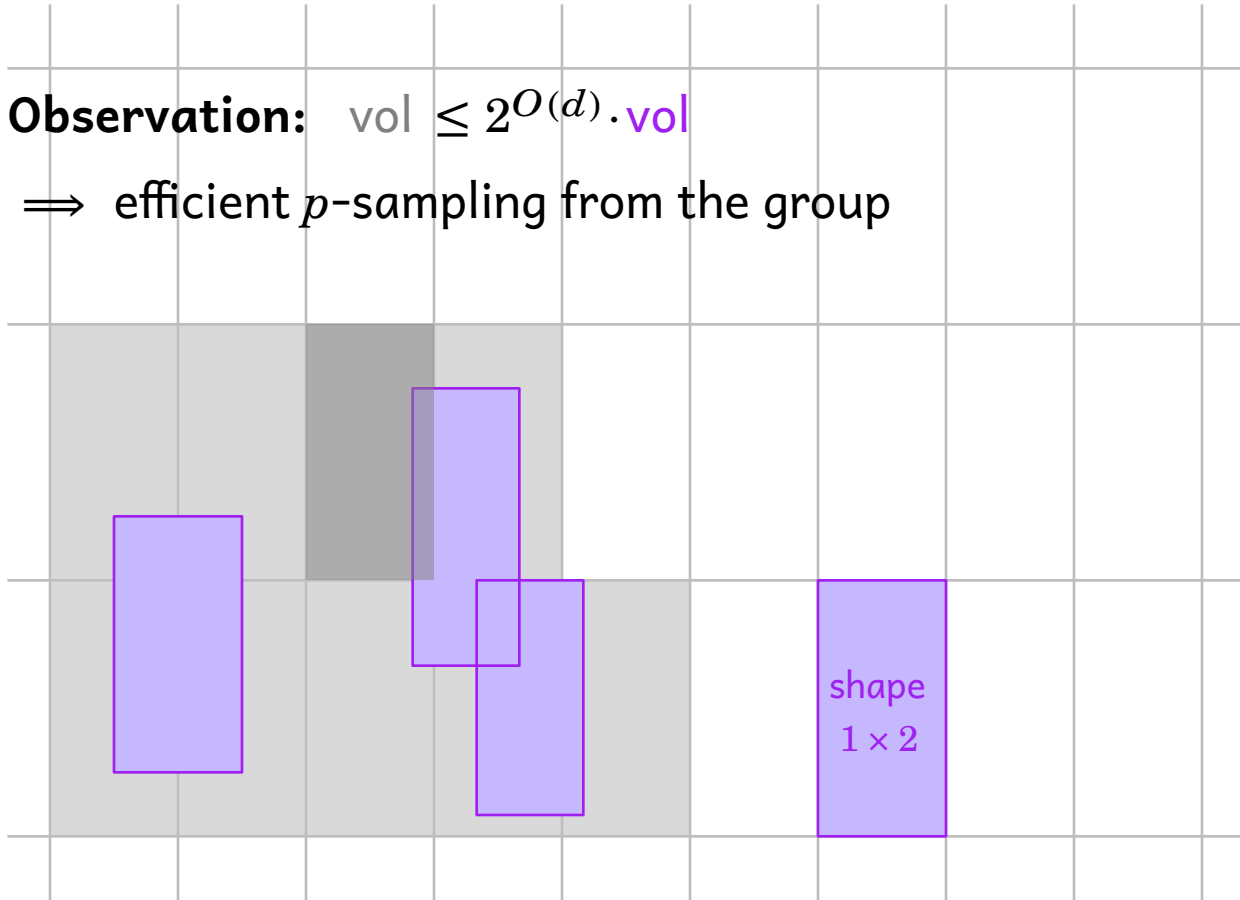
\Rightarrow efficient p -sampling from the group



2 Improved Algorithm for Boxes

Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$

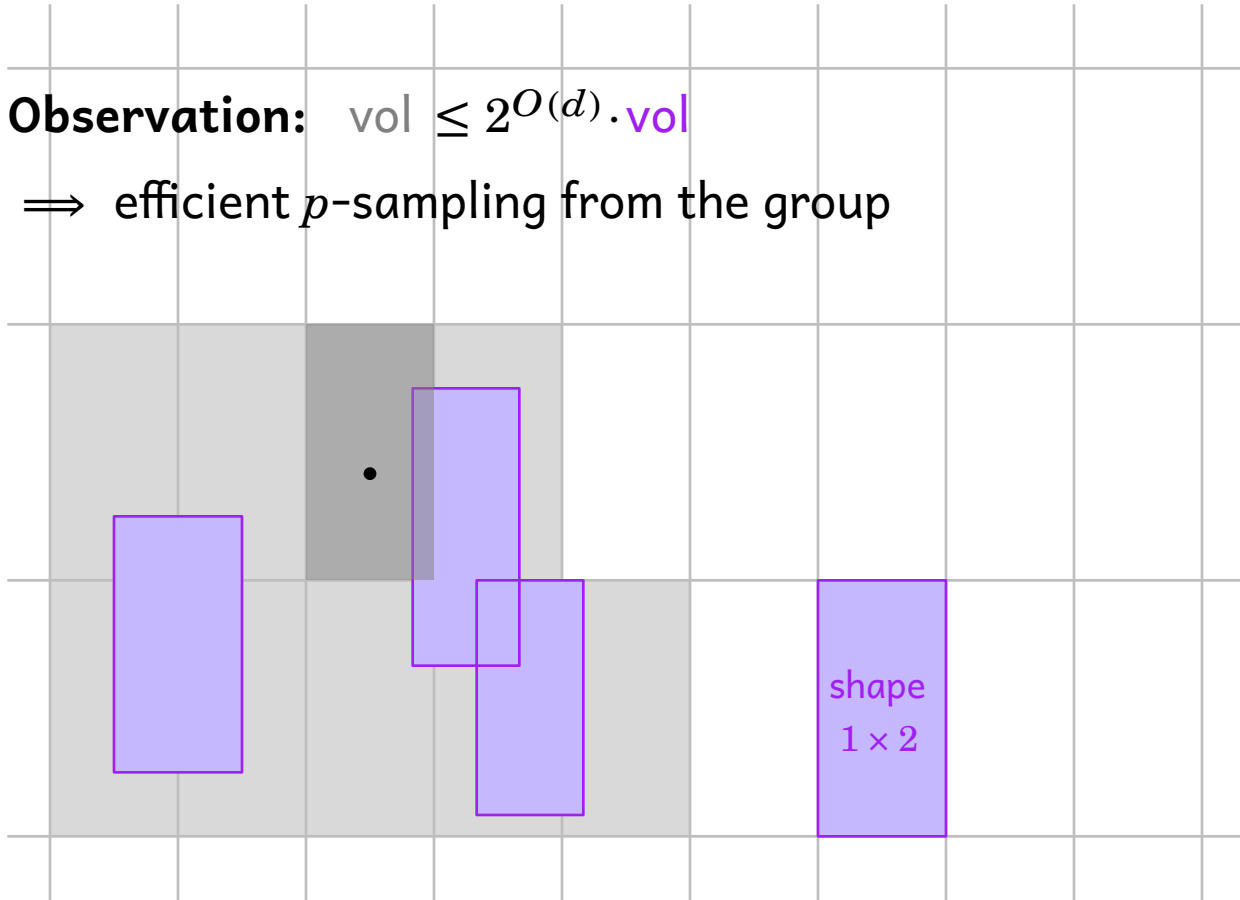
\Rightarrow efficient p -sampling from the group



2 Improved Algorithm for Boxes

Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$

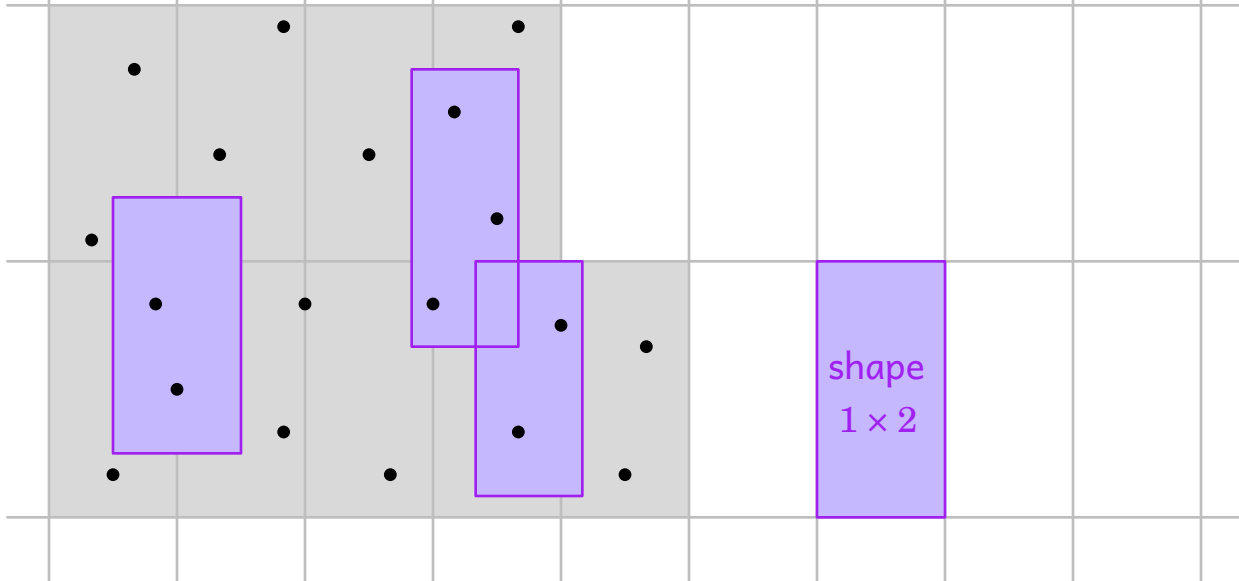
\Rightarrow efficient p -sampling from the group



2 Improved Algorithm for Boxes

Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$

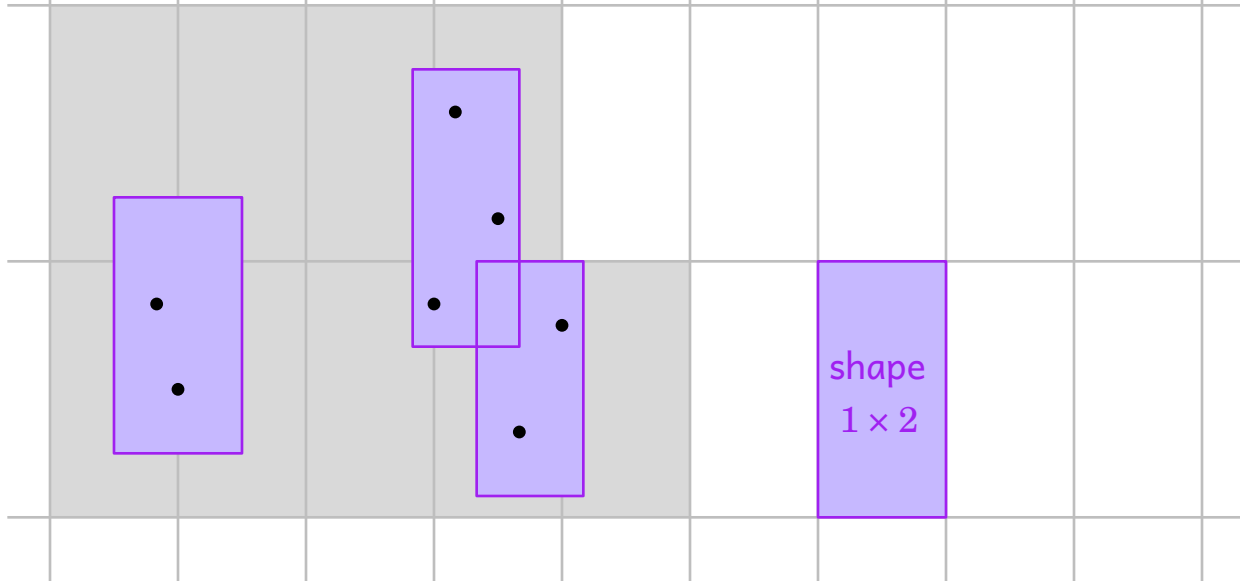
\Rightarrow efficient p -sampling from the group



2 Improved Algorithm for Boxes

Observation: $\text{vol} \leq 2^{O(d)} \cdot \text{vol}$

\Rightarrow efficient p -sampling from the group



2 Improved Algorithm for Boxes

What we have done so far:

2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$

2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$

2 Improved Algorithm for Boxes

What we have done so far:

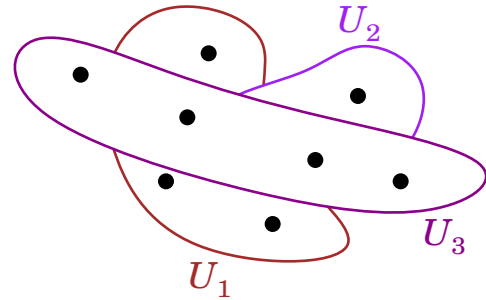
- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!



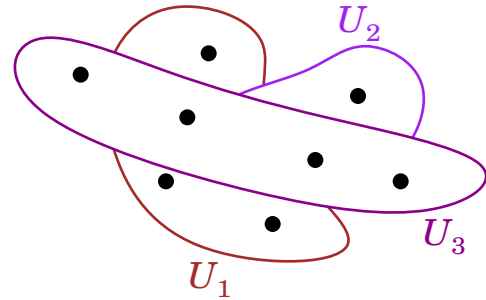
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

$$\mathbf{E}[\text{time}] = \tilde{O}(\mathbf{E}[\#\text{sampled points}])$$



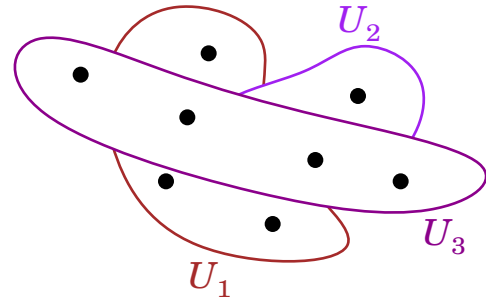
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

$$\begin{aligned} \mathbf{E}[\text{time}] &= \tilde{O}(\mathbf{E}[\#\text{sampled points}]) \\ &\leq \tilde{O}(gp \text{ vol}(U)) = \tilde{O}(g/\epsilon^2) \end{aligned}$$



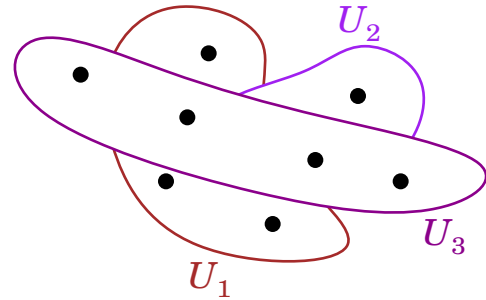
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

$$\begin{aligned} \mathbf{E}[\text{time}] &= \tilde{O}(\mathbf{E}[\#\text{sampled points}]) \\ &\leq \tilde{O}(gp \text{vol}(U)) = \tilde{O}(g/\epsilon^2) \end{aligned}$$



If boxes bounded by n^{10} , then $g \leq (\log n)^d$

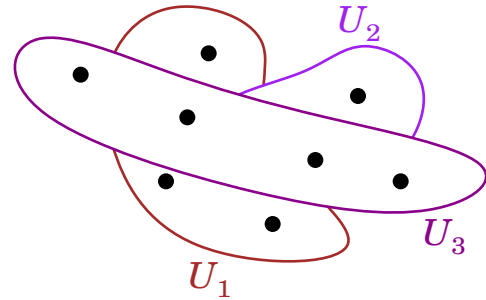
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

$$\begin{aligned} \mathbf{E}[\text{time}] &= \tilde{O}(\mathbf{E}[\#\text{sampled points}]) \\ &\leq \tilde{O}(gp \text{vol}(U)) = \tilde{O}(g/\epsilon^2) \end{aligned}$$



If boxes bounded by n^{10} , then $g \leq (\log n)^d$

Without assumption, $g = \Theta(n)$ possible

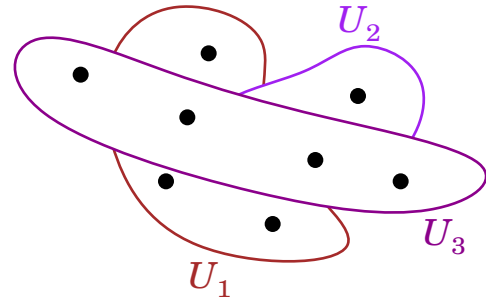
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

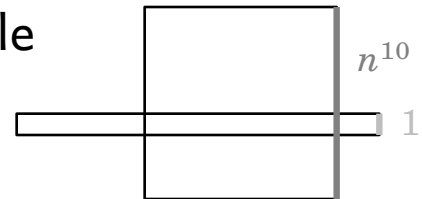
$$\begin{aligned} \mathbf{E}[\text{time}] &= \tilde{O}(\mathbf{E}[\#\text{sampled points}]) \\ &\leq \tilde{O}(gp \text{vol}(U)) = \tilde{O}(g/\epsilon^2) \end{aligned}$$



If boxes bounded by n^{10} , then $g \leq (\log n)^d$

Without assumption, $g = \Theta(n)$ possible

But dissimilar shapes overlap little:



2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

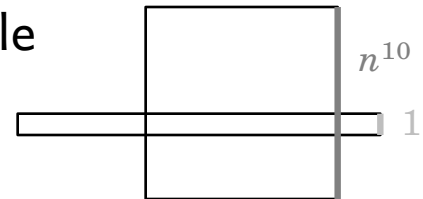
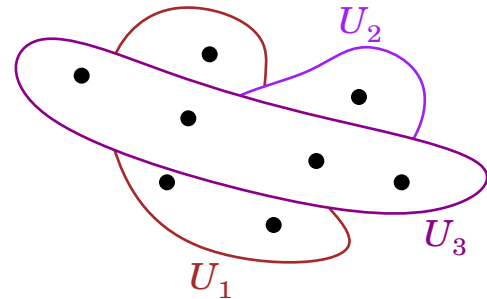
$$\mathbf{E}[\text{time}] = \tilde{O}(\mathbf{E}[\#\text{sampled points}])$$

$$\text{overcount} \leq \tilde{O}(gp \text{ vol}(U)) = \tilde{O}(g/\epsilon^2)$$

If boxes bounded by n^{10} , then $g \leq (\log n)^d$

Without assumption, $g = \Theta(n)$ possible

But dissimilar shapes overlap little:



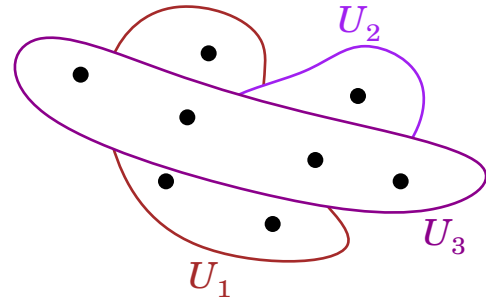
2 Improved Algorithm for Boxes

What we have done so far:

- group boxes by shapes $\Rightarrow U_1, U_2, \dots, U_g$
- efficient membership test $x \in U_i$
- efficient p -sample from U_i

Groups as objects!

$$\begin{aligned} \mathbf{E}[\text{time}] &= \tilde{O}(\mathbf{E}[\#\text{sampled points}]) \\ &\leq \tilde{O}(\hat{g}p \text{vol}(U)) = \tilde{O}(\hat{g}/\varepsilon^2) \end{aligned}$$



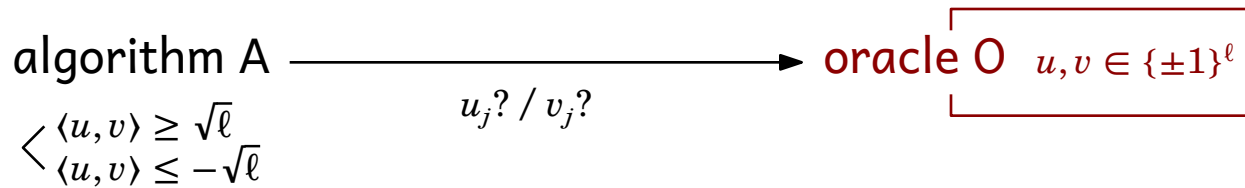
3 Lower Bound for Objects

algorithm A

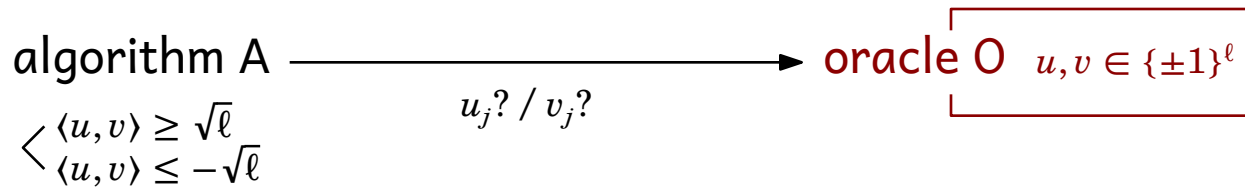
$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

oracle O $u, v \in \{\pm 1\}^\ell$

3 Lower Bound for Objects

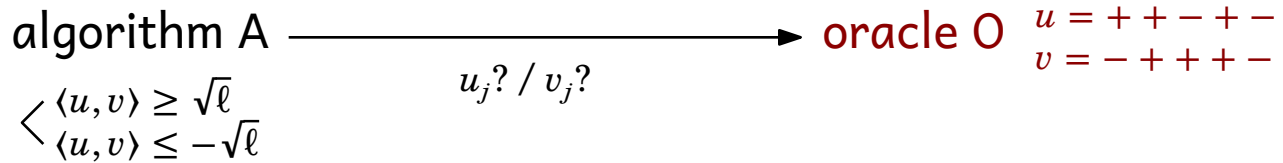


3 Lower Bound for Objects





Fact: $\Omega(\ell)$ queries to oracle O are needed

3 Lower Bound for Objects



Fact: $\Omega(\ell)$ queries to oracle O are needed

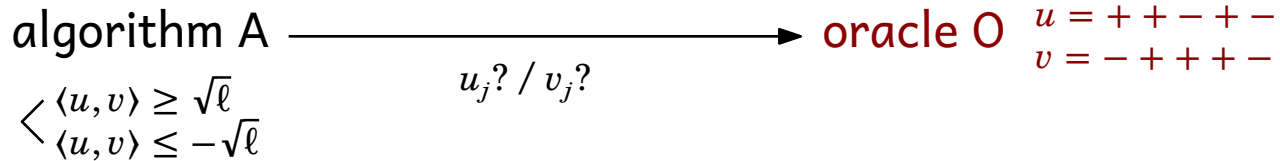
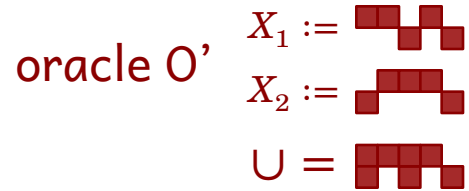
3 Lower Bound for Objects

oracle O' $X_1 :=$ 
 $X_2 :=$ 

algorithm A $\xrightarrow{u_j? / v_j?}$ oracle O $u = + + - + -$
 $v = - + + + -$
 $\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$

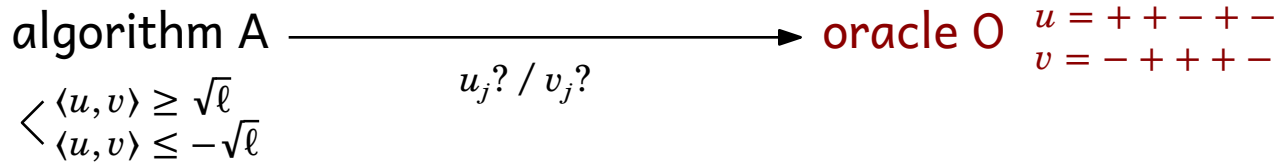
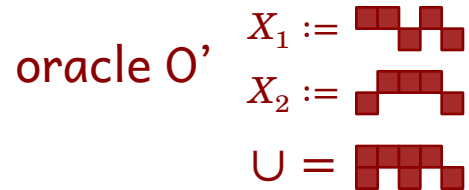
Fact: $\Omega(\ell)$ queries to oracle O are needed

3 Lower Bound for Objects



Fact: $\Omega(\ell)$ queries to oracle O are needed

3 Lower Bound for Objects



Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2 \text{vol}(U)$

3 Lower Bound for Objects

$(1 \pm \varepsilon) \text{vol}(\bigcup_{i=1}^n X_i)$
 algorithm A'

oracle O'

$X_1 :=$ 

$X_2 :=$ 

$U =$ 

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

oracle O

$u = + + - + -$

$v = - + + + -$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2 \text{vol}(U)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$$

algorithm A'

$$\varepsilon := \frac{1}{4\sqrt{\ell}}$$

oracle O'

$$X_1 := \begin{array}{ccccccc} \blacksquare & \blacksquare & & \blacksquare & \blacksquare & & \\ & & & & & & \\ & & & & & & \blacksquare \end{array}$$

$$X_2 := \begin{array}{ccccccc} & & & \blacksquare & \blacksquare & \blacksquare & \\ & & & & & & \\ \blacksquare & & & & & & \blacksquare \end{array}$$

$$U = \begin{array}{ccccccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \\ & & & & & & \\ & & & & & & \blacksquare \end{array}$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2\text{vol}(U)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$$

algorithm A'

$$\varepsilon := \frac{1}{4\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$X_1 := \begin{array}{cccc} \blacksquare & \blacksquare & & \blacksquare \\ & & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$$X_2 := \begin{array}{cccc} & & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$$U = \begin{array}{cccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2\text{vol}(U)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$$

algorithm A'

$$\varepsilon := \frac{1}{4\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$X_1 := \begin{array}{ccccccc} \blacksquare & \blacksquare & & \blacksquare & \blacksquare & & \\ & & & & & & \\ & & & & & & \blacksquare \end{array}$$

$$X_2 := \begin{array}{ccccccc} & & & \blacksquare & \blacksquare & \blacksquare & \\ & & & & & & \\ & & & & & & \blacksquare \end{array}$$

$$U = \begin{array}{ccccccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \\ & & & & & & \\ & & & & & & \blacksquare \end{array}$$

$u_j? / v_j?$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2\text{vol}(U)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{4\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$X_1 := \begin{array}{cccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$$X_2 := \begin{array}{cccc} & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$$U = \begin{array}{cccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$\leq q$ queries
 $u_j? / v_j?$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2\text{vol}(U)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol}(\cup_{i=1}^n X_i)$$

algorithm A'

$$\varepsilon := \frac{1}{4\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$X_1 := \begin{array}{cccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \blacksquare \end{array}$$

$$X_2 := \begin{array}{cccc} & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & & & \blacksquare \end{array}$$

$$U = \begin{array}{cccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{array}$$

$\leq q$ queries
 $u_j? / v_j?$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

Fact: $\Omega(\ell)$ queries to oracle O are needed

Observation: $\langle u, v \rangle = 3\ell - 2\text{vol}(U)$

Conclusion: $q \geq \Omega(\ell) = \Omega(1/\varepsilon^2)$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$\bigcup_{i \leq n/2} X_i$$

$$\bigcup_{i > n/2} X_i$$

$$\bigcup$$



$u_j? / v_j?$

oracle O

$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$\bigcup_{i \leq n/2} X_i$$

$$\bigcup_{i > n/2} X_i$$

$$\bigcup$$



$u_j? / v_j?$

oracle O

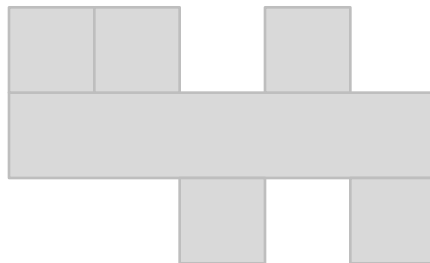
$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$$\bigcup_{i \leq n/2} X_i$$



3 Lower Bound for Objects

$(1 \pm \epsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm A'

$$\epsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

\bigcup



$u_j? / v_j?$

oracle O

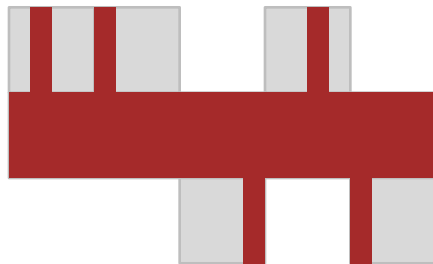
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



X_1

3 Lower Bound for Objects

$(1 \pm \epsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm A'

$$\epsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

\bigcup



$u_j? / v_j?$

oracle O

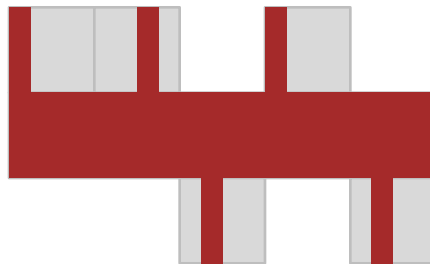
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



X_2

3 Lower Bound for Objects

$(1 \pm \epsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm A'

$$\epsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

\bigcup



$u_j? / v_j?$

oracle O

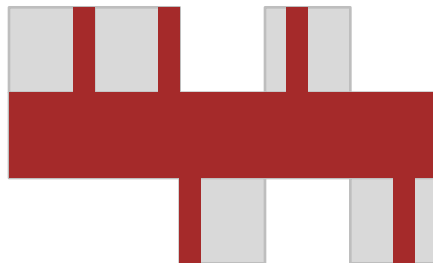
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



X_3

3 Lower Bound for Objects

$(1 \pm \varepsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

\bigcup



$u_j? / v_j?$

oracle O

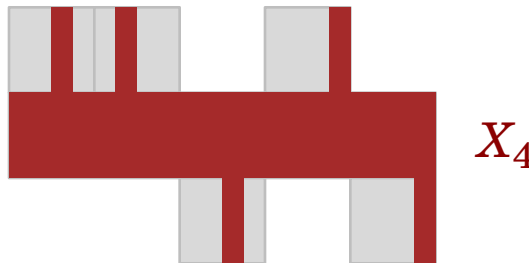
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



3 Lower Bound for Objects

$(1 \pm \varepsilon) \text{vol}(\bigcup_{i=1}^n X_i)$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$ →
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

\bigcup



$u_j? / v_j?$

oracle O

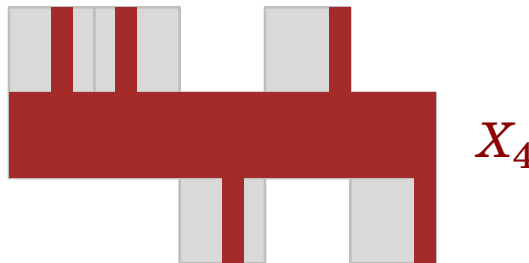
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$\bigcup_{i \leq n/2} X_i$$

$$\bigcup_{i > n/2} X_i$$

$$\bigcup$$



$u_j? / v_j?$

oracle O

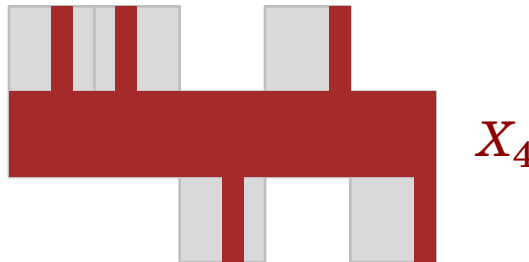
$$u = + + - + -$$

$$v = - + + + -$$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$$\bigcup_{i \leq n/2} X_i$$



3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$\bigcup_{i \leq n/2} X_i$

$\bigcup_{i > n/2} X_i$

U



$u_j? / v_j?$

oracle O

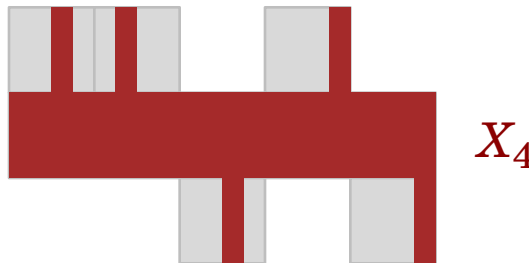
$u = + + - + -$

$v = - + + + -$

algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$\bigcup_{i \leq n/2} X_i$



3 Lower Bound for Objects

$$(1 \pm \varepsilon) \text{vol} \left(\bigcup_{i=1}^n X_i \right)$$

algorithm A'

$$\varepsilon := \frac{1}{6\sqrt{\ell}}$$

q queries

- $\text{vol}(X_i) = ?$
- $x \in X_i?$
- random $x \in X_i$

oracle O'

$$\bigcup_{i \leq n/2} X_i$$

$$\bigcup_{i > n/2} X_i$$

$$\bigcup$$



$\leq q/n$ queries
 $u_j? / v_j?$

oracle O

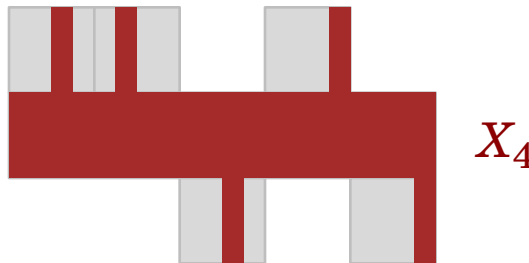
$$u = + + - + -$$

$$v = - + + + -$$

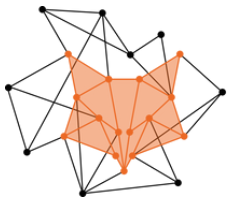
algorithm A

$$\begin{cases} \langle u, v \rangle \geq \sqrt{\ell} \\ \langle u, v \rangle \leq -\sqrt{\ell} \end{cases}$$

$$\bigcup_{i \leq n/2} X_i$$



X_4

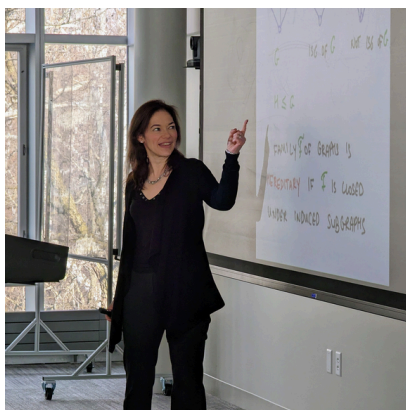


ADFOCS'25

25th Max Planck Advanced Course on the Foundations of Computer Science

18–22 August 2025, Saarbrücken, Germany

Graph Decompositions and Efficient Algorithms



Induced Subgraphs

Maria Chudnovsky

Princeton University



Structural Sparsity

Michał Pilipczuk

University of Warsaw



Expander Hierarchies

Thatchaphol Saranurak

University of Michigan



Early registration deadline: 14 July 2025

Organizers: Cosmina Croitoru, Danupon Nanongkai, Daniel Neuen,
and Marek Sokolowski

adfocs@mpi-inf.mpg.de



MAX PLANCK INSTITUTE
FOR INFORMATICS