Learning Partitions with Optimal Query and Round Complexities

Conference on Learning Theory (**COLT**) 2025

Hadley Black
UCSD

Arya Mazumdar
UCSD

Barna Saha
UCSD

Clustering via Crowdsourcing

Are these animals in the same genus?

- Can we offload the work of computing a clustering by asking simple questions to external individuals?

- **Pairwise same-cluster queries:** Are these two points of the same type?

Yes! *No.*

Learning Partitions with Queries

Problem statement

- Set U of n elements
- Hidden k-partition $X_1 \sqcup \cdots \sqcup X_k = U$
- Learn $X_1, \ldots, X_k$ **exactly** using same-set queries

Perspective & motivation

Practical clustering model:

- Leveraging crowd responses to simple questions enables

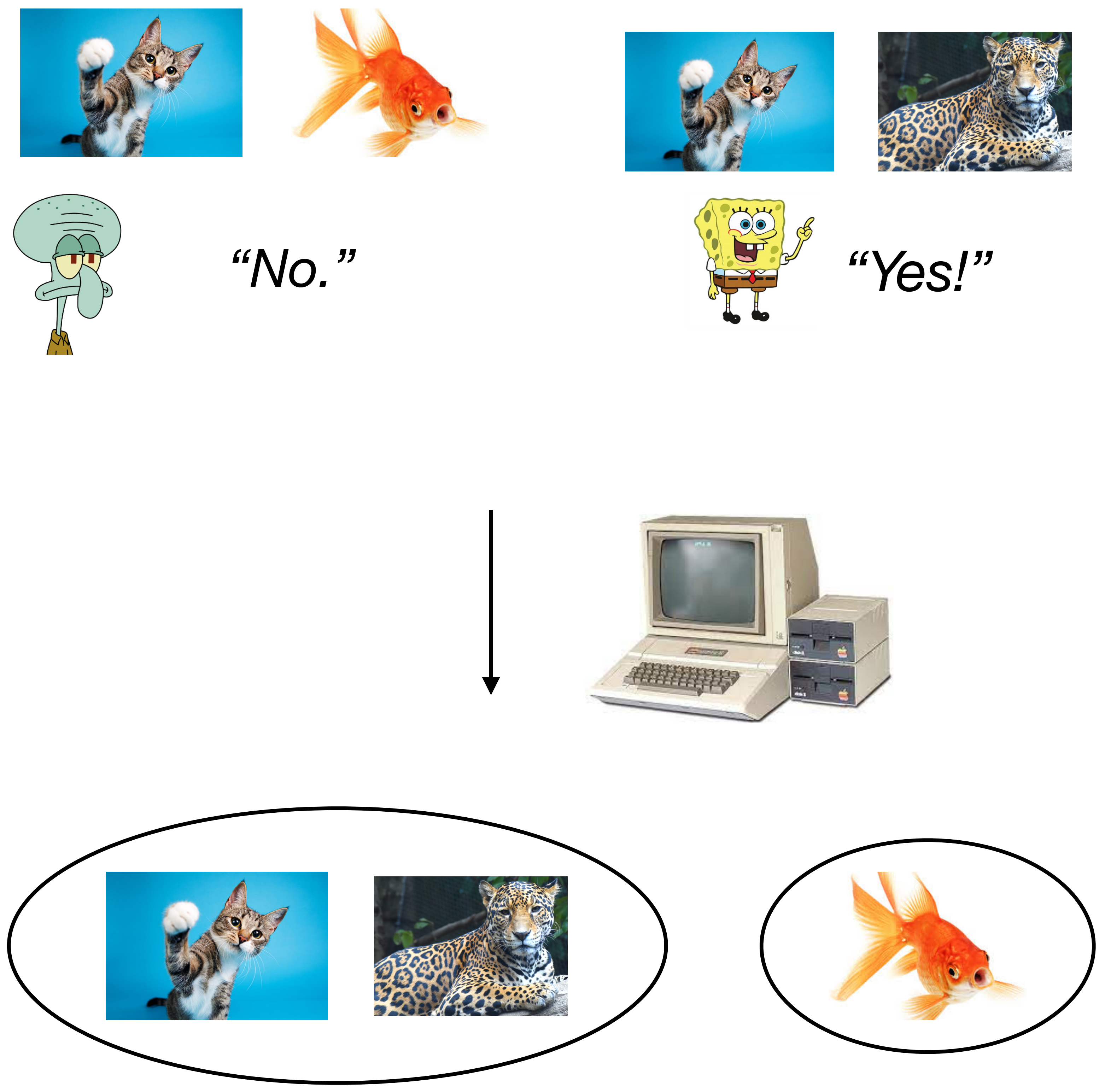
  (a) Label-invariance

  (b) Simple combinatorial setting where geometry has been removed ("offloaded" to the oracle)

Theoretical motivation:

- Partition learning is a fundamental problem
- Key aspects remained unexplored

Query profile



Learned clustering

Learning Partitions with Queries

Problem statement

- Set U of n elements
- Hidden k-partition $X_1 \sqcup \cdots \sqcup X_k = U$
- Learn $X_1, \ldots, X_k$ **exactly** using same-set queries

Considerations in this work

(1) Query complexity

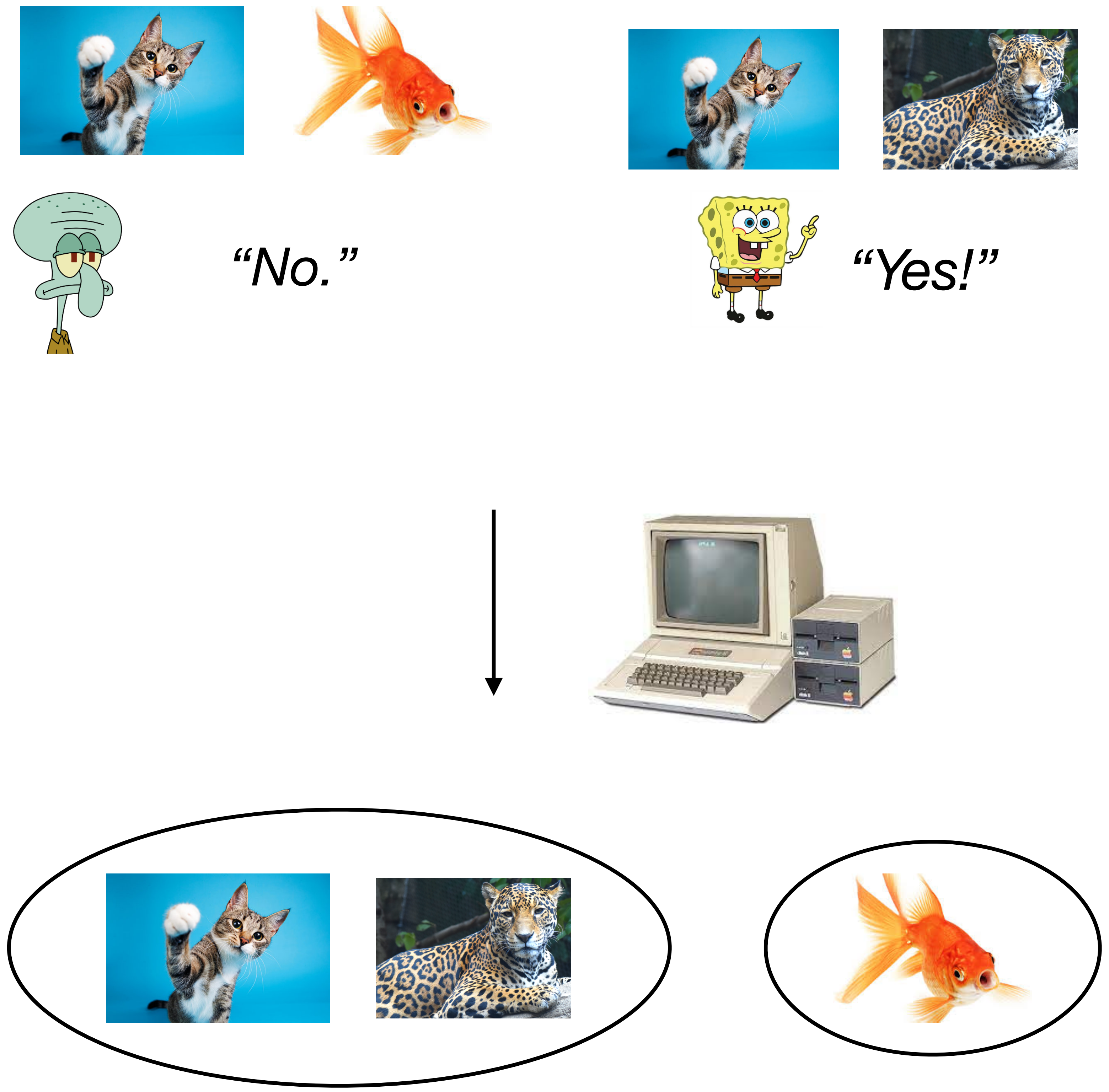
(2) Round complexity

- Responses may be slow
- Important to parallelize queries as much as possible

(3) "Size" complexity

- Consider generalized **subset** queries
- Oracle may not be able to handle large subsets

Query profile



Learned clustering

Learning Partitions with Pair Queries

Reyzin-Srivastava [ALT 07], Mazumdar-Saha [NeuIPS 17], Mazumdar-Saha [AAAI 17], Mazumdar-Pal [NeurIPS 17], Mitzenmacher-Tsouraskis [16], Saha-Subramanian [ESA 19], Pia-Ma-Tzamos [COLT 22], Bressan-Cesa-Bianchi-Lattanzi-Paudice [NeurIPS 20], Huleihal-Mazumdar-Médard-Pal [NeurIPS 19], etc…

- Set U of n elements
- Hidden k-partition $X_1 \sqcup \cdots \sqcup X_k = U$
- Learn $X_1, \ldots, X_k$ **exactly** using same-set queries

Tight query complexity bound

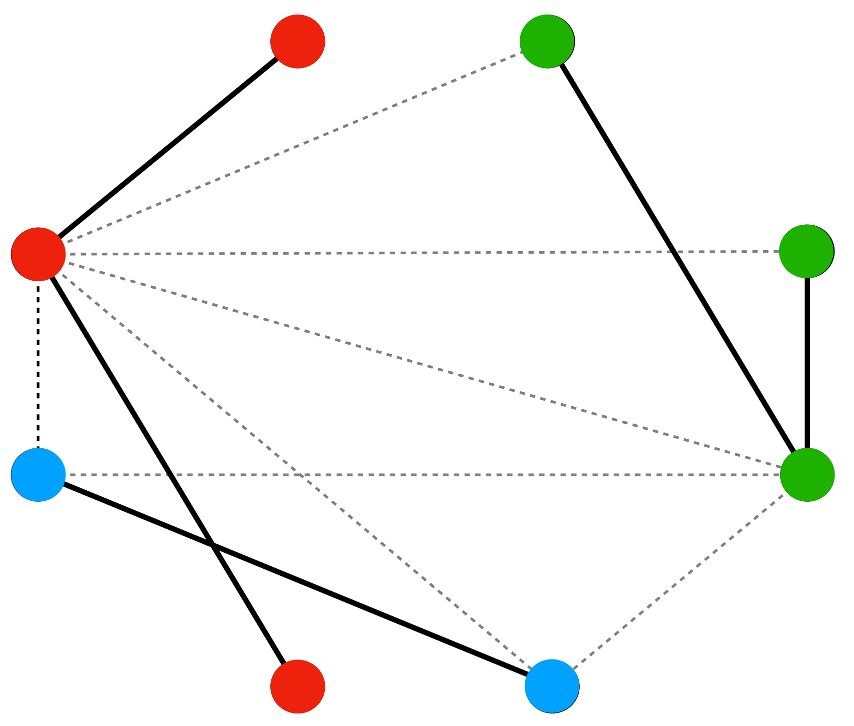
$$\Theta(nk)$$

Upper bound
Reyzin-Srivastava 07

Lower bound
Davidson-Khanna-Milo-Roy 14

Classic algorithm of Reyzin-Srivastava:

Learn clusters one-by-one



!!

$k-1$ rounds of adaptivity

Can we do better?

Question

What is the minimum number of rounds that suffice to achieve $O(nk)$ queries?

Question

Given a budget of r rounds, what is the optimal query complexity?

Result 1: Round Complexity of Pair Queries

- Set U of n elements
- Hidden k-partition $X_1 \sqcup \cdots \sqcup X_k = U$
- Learn $X_1, \ldots, X_k$ **exactly** using same-set queries

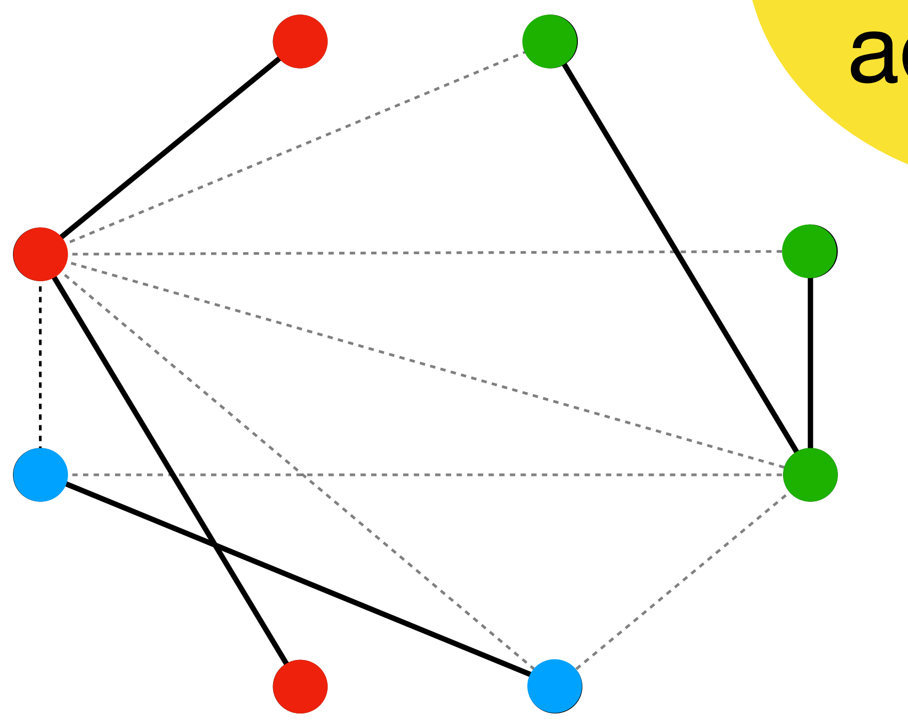
Theorem *

$$\Theta\left(n^{1+\frac{1}{2^r-1}} \cdot k^{1-\frac{1}{2^r-1}}\right)$$

Fully adaptive

$\Theta(nk)$

$O(\log\log n)$ $\cancel{k-1}$ rounds of adaptivity **!!**



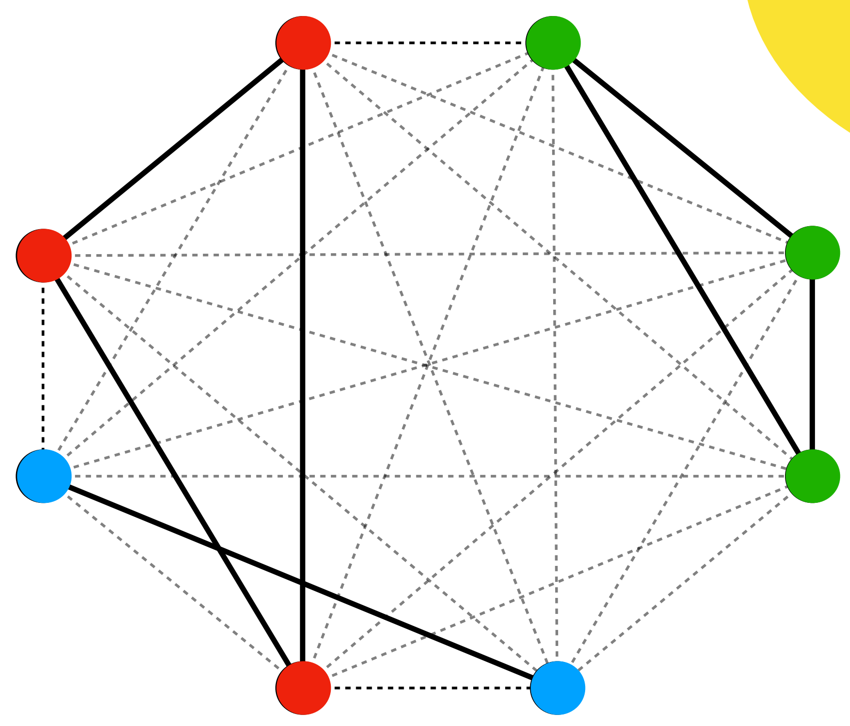
r ***rounds***?

A double exponential improvement when $k \geq n^{0.01}$

Non-adaptive

$\Theta(n^2)$

1 round of adaptivity



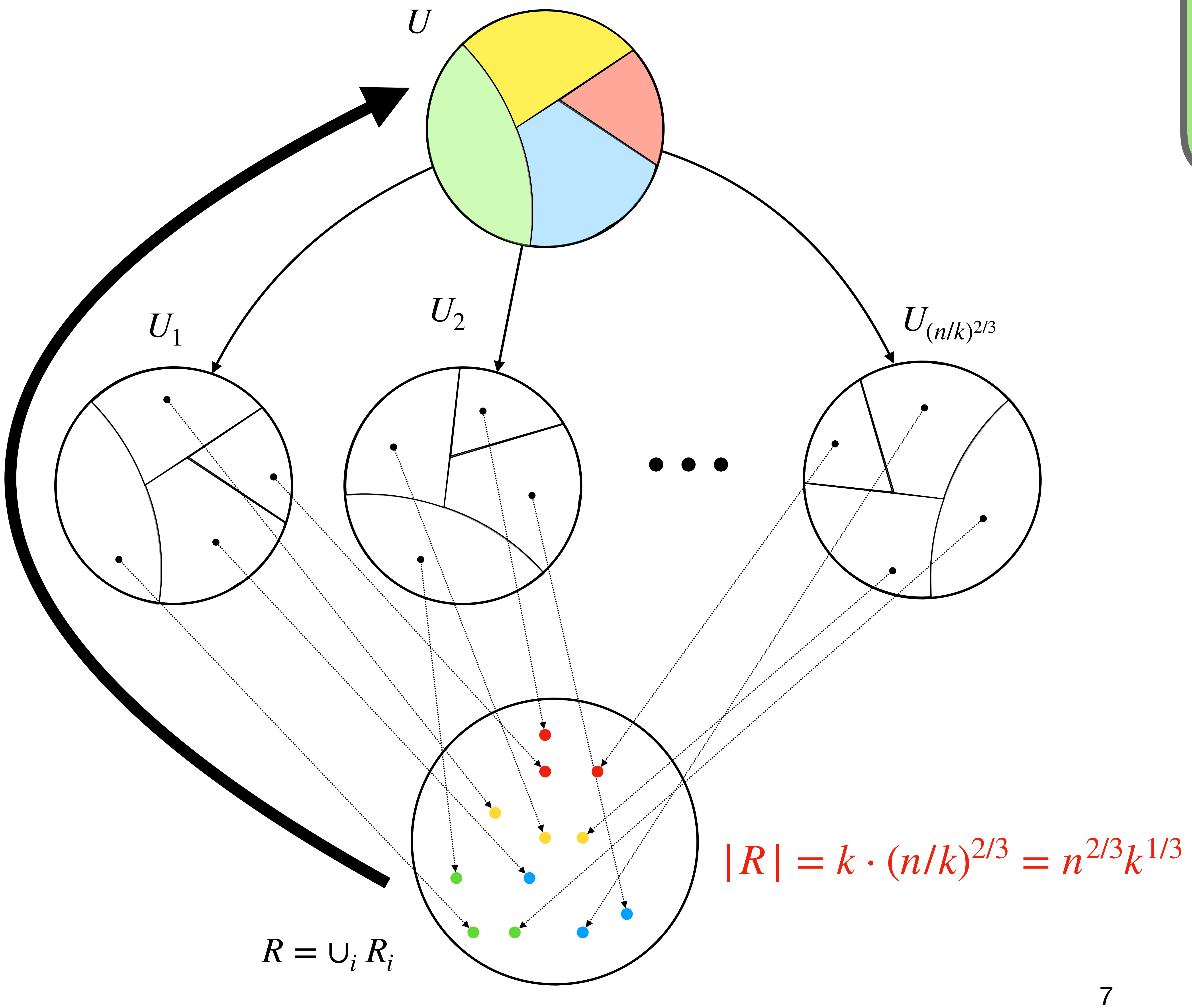
Fine print:

* Algorithm and lower bound are deterministic

* lower bound matches exactly for $r = O(1)$

* … but only ever off by a $r = O(\log\log n)$ factor

Algorithm: $r = 2$



- Split into $(n/k)^{2/3}$ sets of size $n^{1/3}k^{2/3}$
- **Round 1:** Run non-adaptive algorithm in each
- R_i = one representative from each cluster found in U_i
- **Round 2:** Run non-adaptive algorithm on $\cup_i R_i$

$\longrightarrow$ Combine partitions computed in round 1 using information in gained in round 2

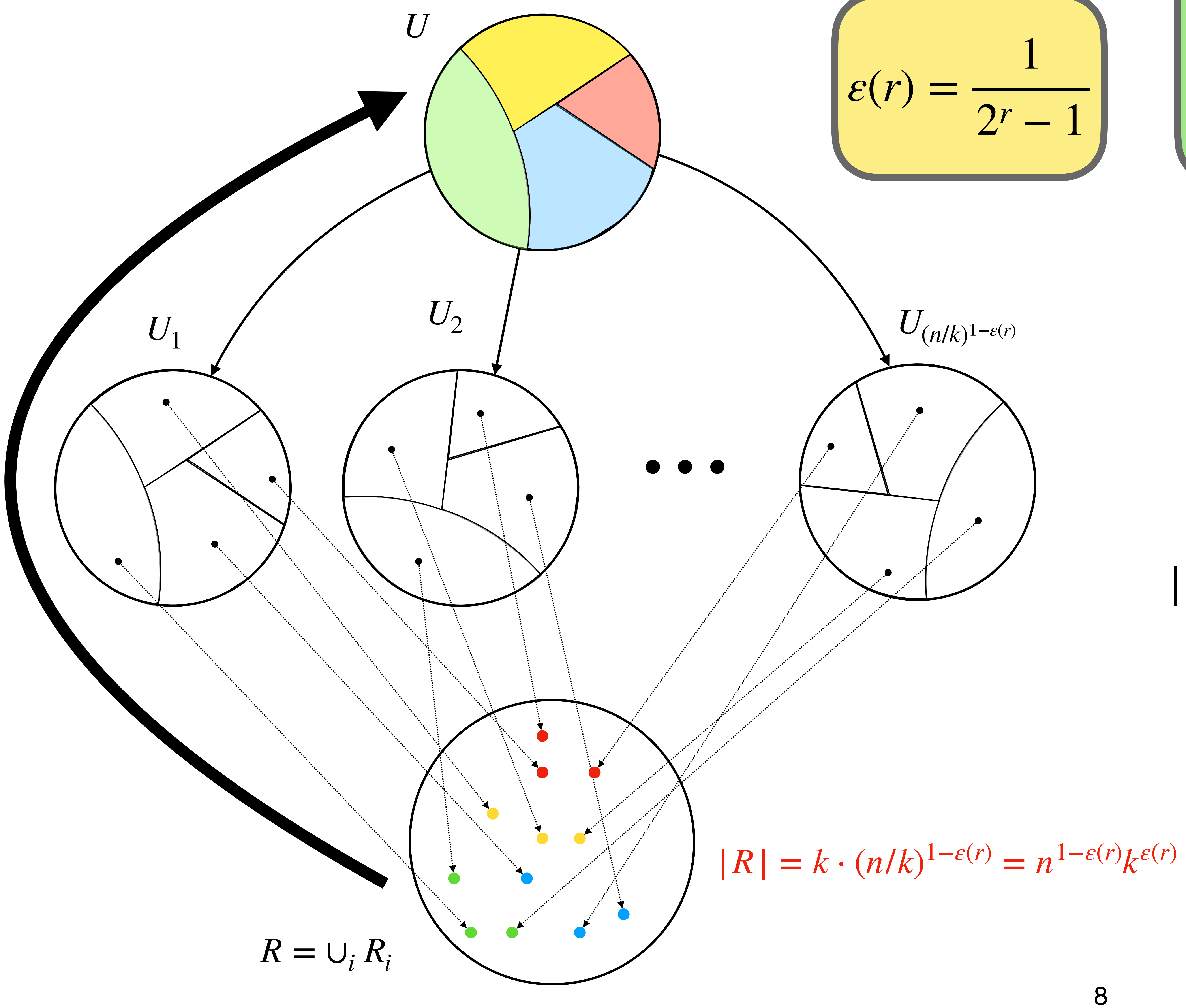
Round 1 queries

$$(n/k)^{2/3} \cdot \left(n^{1/3}k^{2/3}\right)^2 = n^{4/3}k^{2/3}$$

Round 2 queries

$$(k \cdot (n/k)^{2/3})^2 = n^{4/3}k^{2/3}$$

Algorithm: general r



$$\varepsilon(r) = \frac{1}{2^r - 1}$$

- Split into $(n/k)^{1-\varepsilon(r)}$ sets of size $n^{\varepsilon(r)}k^{1-\varepsilon(r)}$
- **Round 1:** Run non-adaptive algorithm in each
- $R_i =$ one representative from each cluster found in U_i
- **Round $2,\ldots,r$:** Run $r-1$ round algorithm on $\cup_i R_i$

Round 1 queries

$$(n/k)^{1-\varepsilon(r)} \cdot \left(n^{\varepsilon(r)}k^{1-\varepsilon(r)}\right)^2 = n^{1+\varepsilon(r)}k^{1-\varepsilon(r)}$$

Round $2,\ldots,r$ queries

$$|R|^{1+\varepsilon(r-1)}k^{1-\varepsilon(r-1)} = (k \cdot (n/k)^{1-\varepsilon(r)})^{1+\varepsilon(r-1)}k^{1-\varepsilon(r-1)}$$

$$= n^{1+\varepsilon(r)}k^{1-\varepsilon(r)}$$

Ugly expression… but the math works out

Note: setting constants appropriately allows to avoid an additional r factor in final query complexity

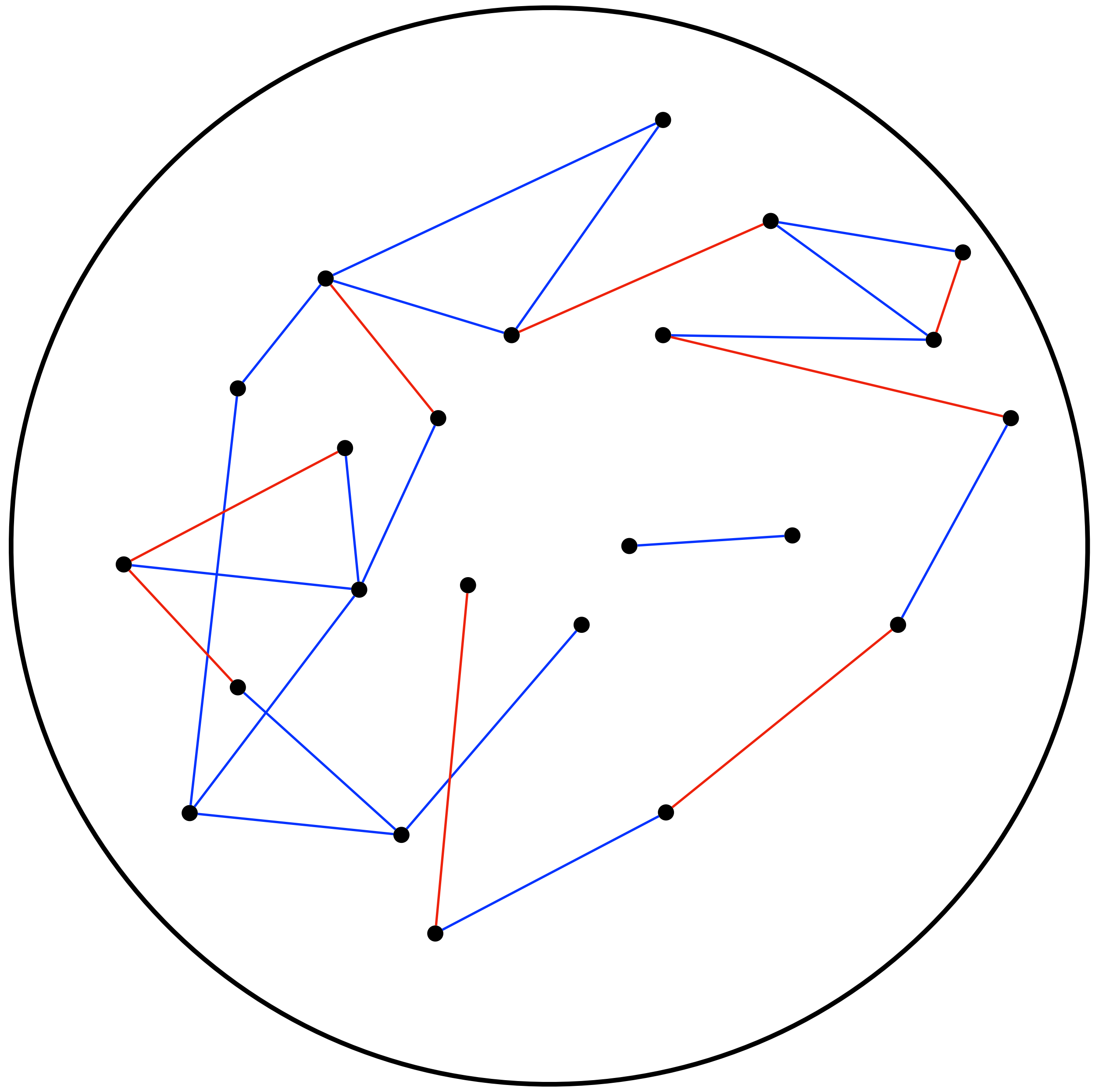
Lower bound high level ideas

$$\Omega\left(\frac{1}{r}\cdot n^{1+\frac{1}{2^r-1}}\cdot k^{1-\frac{1}{2^r-1}}\right)$$
$$\forall k \geq r+2$$

- Consider arbitrary **deterministic** algorithm
- Queries appearing in r rounds $Q = Q_1 \cup \underbrace{Q_2 \cup \cdots \cup Q_r}_{\text{Depend on previous query responses}} \subseteq \binom{U}{2}$

Fixed set

- View queries as **edges** in a graph over U



$G_2(U, Q_1 \cup Q_2)$

(The query graph after 2 rounds)

Idea: If $Z \subset U$ is

(a) an independent set (IS), **and**

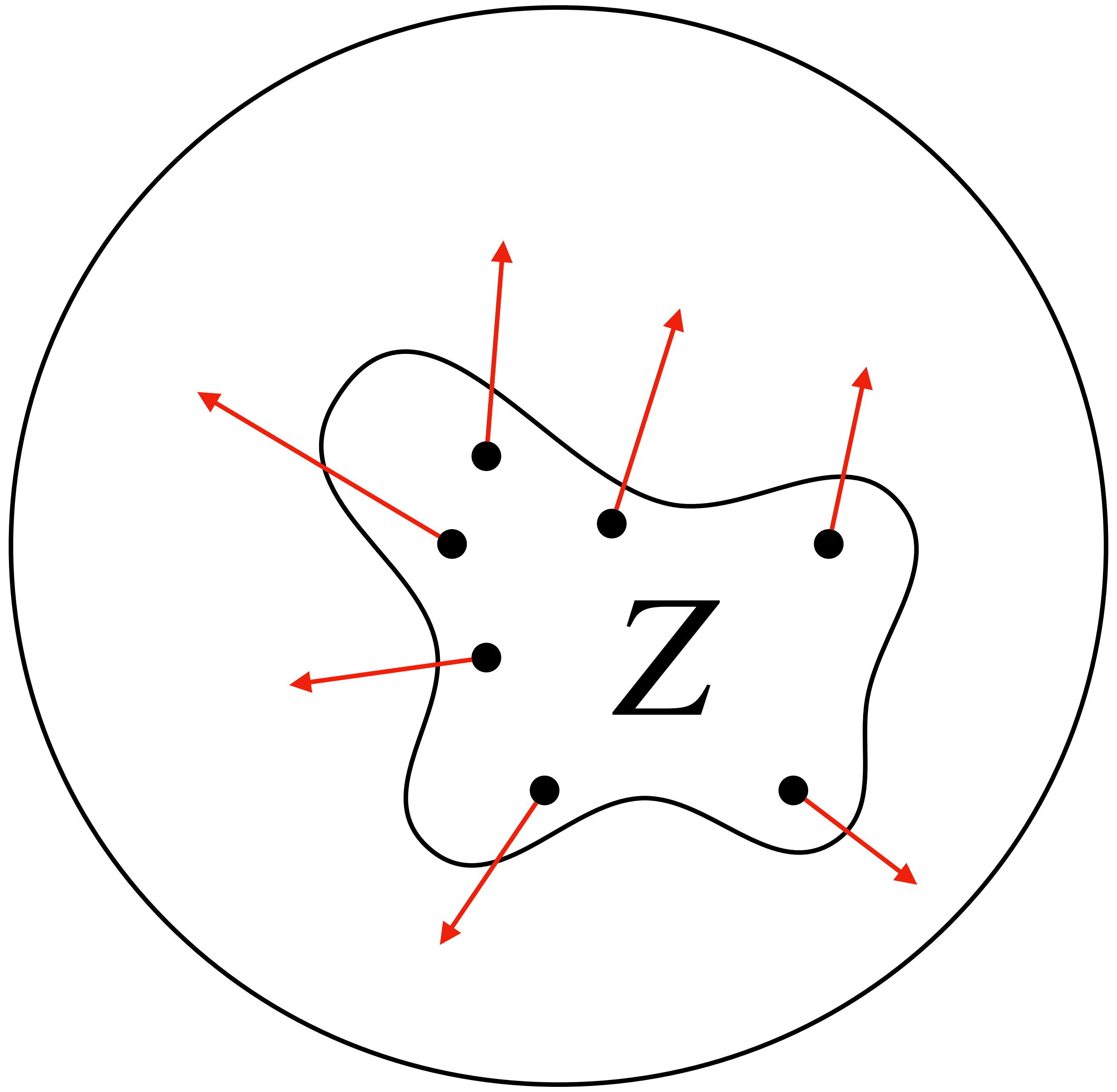
(b) every query that touches Z has returned "not same set",

then we have **not learned anything** about partition in Z

Turán's theorem: *!!*

$q \geq n$ queries so far $\Longrightarrow$

G contains an IS of size $\approx n^2/q$

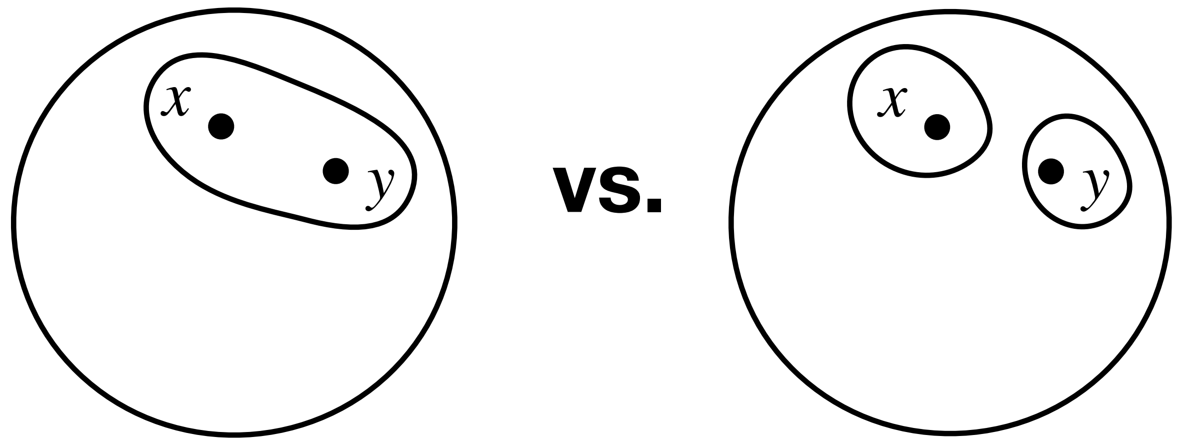


Warm-up: $\Omega\left(n^{1+\frac{1}{2^r-1}}\right)$, $k \geq r+2$

Base case: $r = 1, k = 3$:

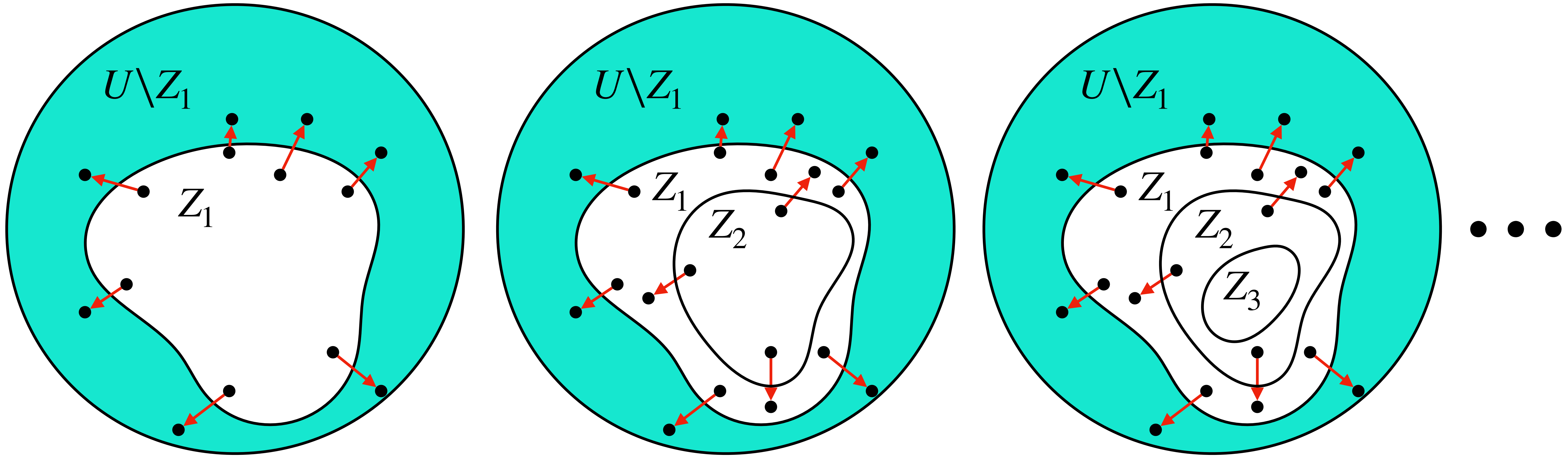
If $|Q| \ll n^2$, there exists $(x, y) \in \binom{U}{2} \backslash Q$ $\Longrightarrow$

Cannot distinguish



Induction: $r > 1, k = r+2$:

If $|Q_1| \ll n^{1+\frac{1}{2^r-1}}$, there exists an **IS** Z_1 in G_1 of size $\approx n^{1-\frac{1}{2^r-1}}$ by **Turán's theorem**



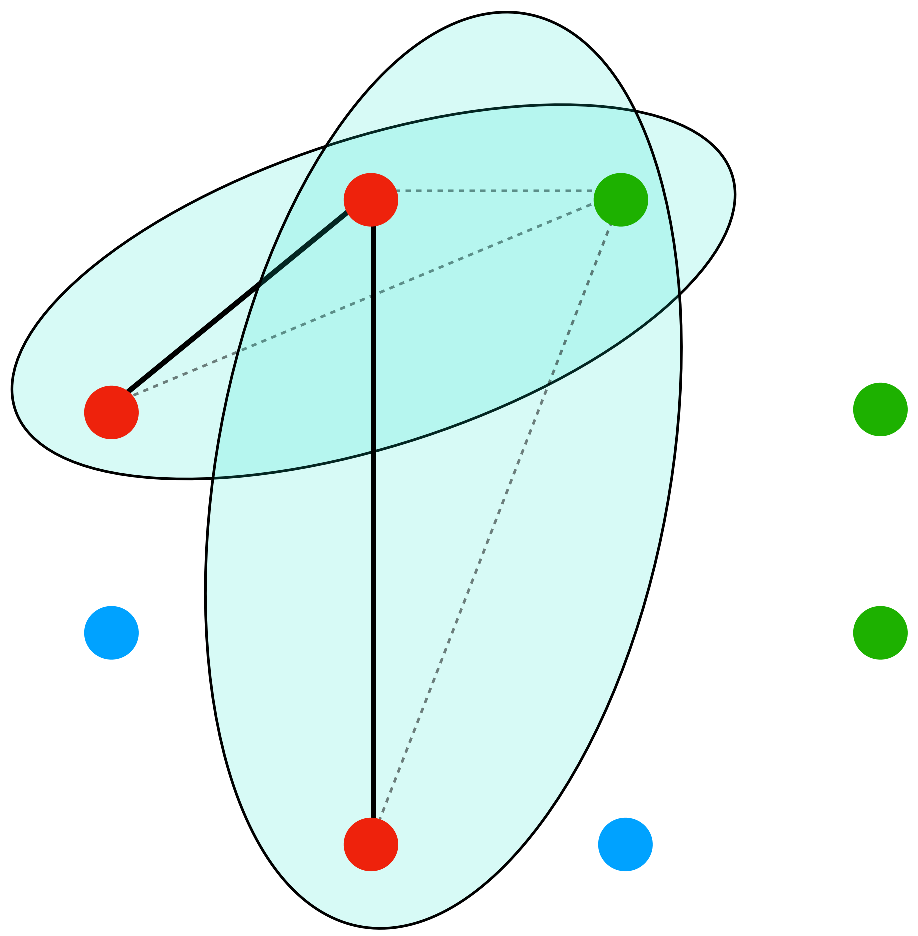
- Fix $U \backslash Z_1$ as one cluster
- Remaining $r-1$ rounds restricted in Z:
    - By induction, if $|Q_2 \cup \cdots \cup Q_r| \ll |Z_1|^{1+\frac{1}{2^{r-1}-1}} = n^{1+\frac{1}{2^r-1}}$, then there exists two partitions P_1, P_2 over Z_1 into $r+1$ sets that are **not distinguished**

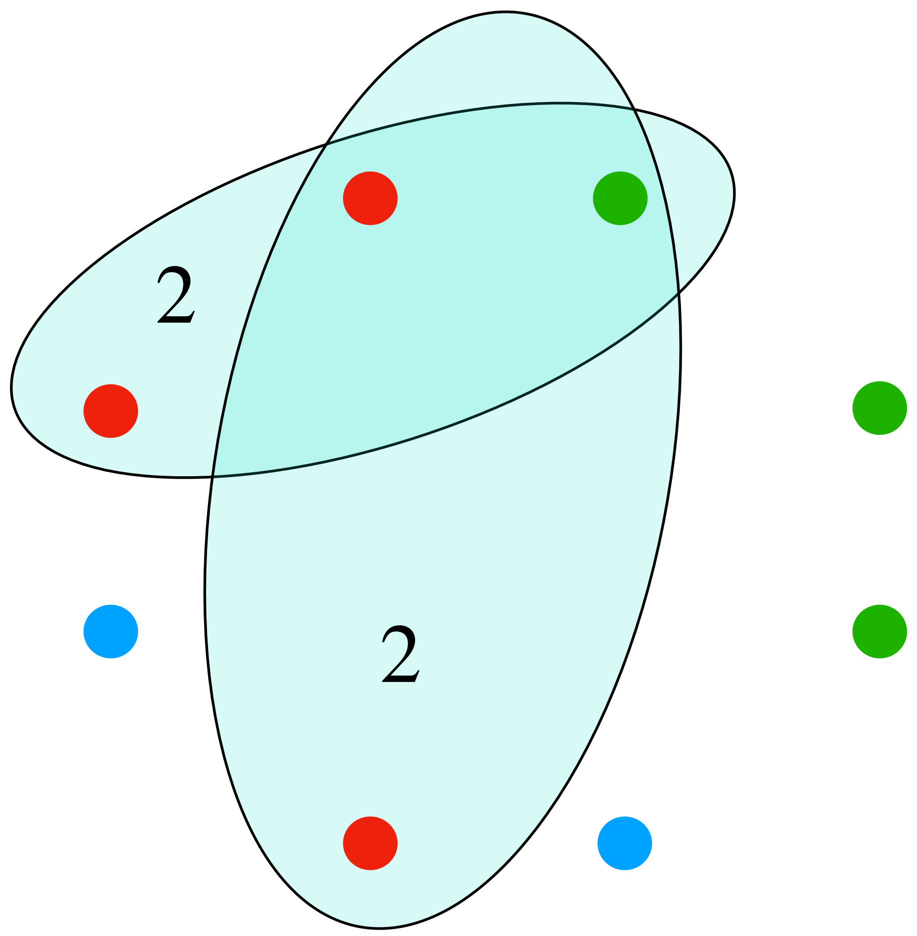
Bringing in dependence on k is significantly more challenging, but core ideas are similar

Generalizing to Subset Queries

Chakrabarty-Liao [FSTTCS 24], Black-Lee-Mazumdar-Saha [NeurIPS 24]

- Set U of n elements
- Hidden k-partition $X_1 \sqcup \cdots \sqcup X_k = U$
- How many **subset queries of size at most s** to learn $X_1, \ldots, X_k$ **exactly**?





Strong	**Weak**
Returns full description of partition on S	Returns # clusters intersecting S

$s = n$ — 1 query is sufficient

Not practical

$O(n)$ **adaptive** [CL24] $\Omega(n)$ info-theory

$\widetilde{O}(n)$ **non-adaptive** [BLMS24]

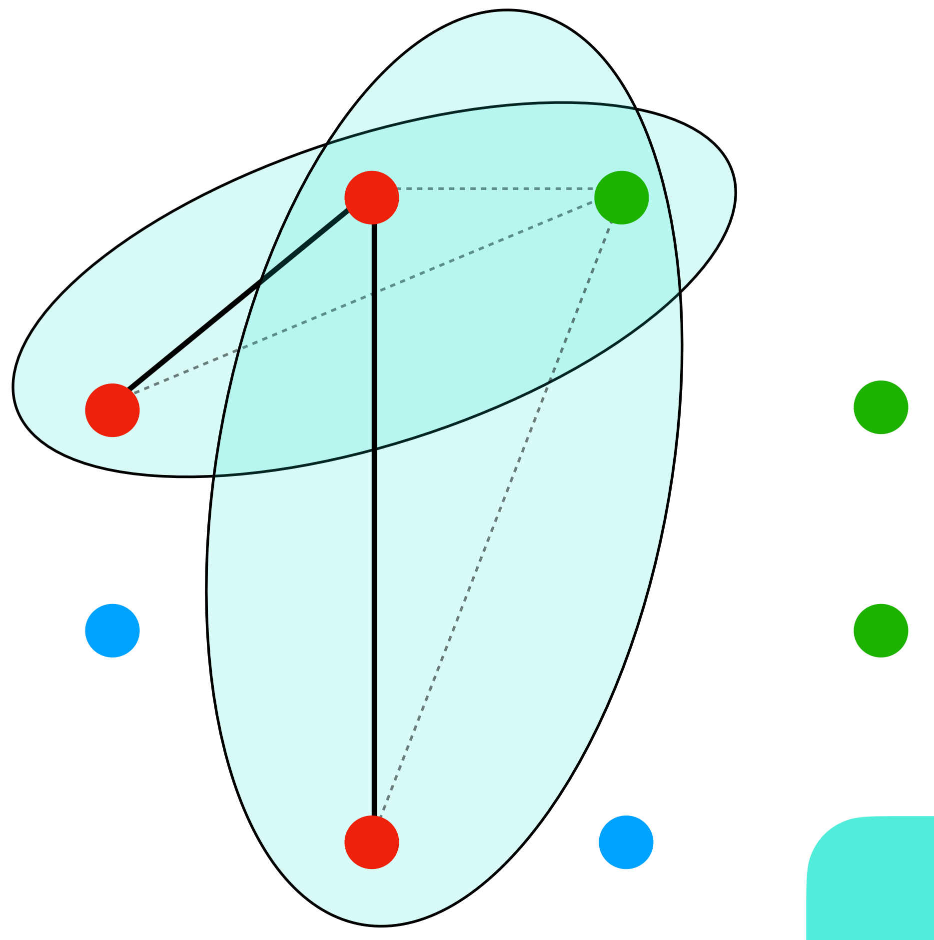
Question: What is the minimum query size s needed to achieve $\widetilde{O}(n)$ queries?

Basic observation: s^2 pair queries simulate 1 strong subset query

$\Longrightarrow$ $\Omega(nk/s^2)$ **adaptive**, $\Omega(n^2/s^2)$ **non-adaptive** $\xrightarrow{+\text{ info theory}}$ $\Omega(nk/s^2 + n)$ **adaptive**, $\Omega(n^2/s^2 + n)$ **non-adaptive**

ENCORE

Result 2: Size Complexity of Subset Queries (Non-adaptive)



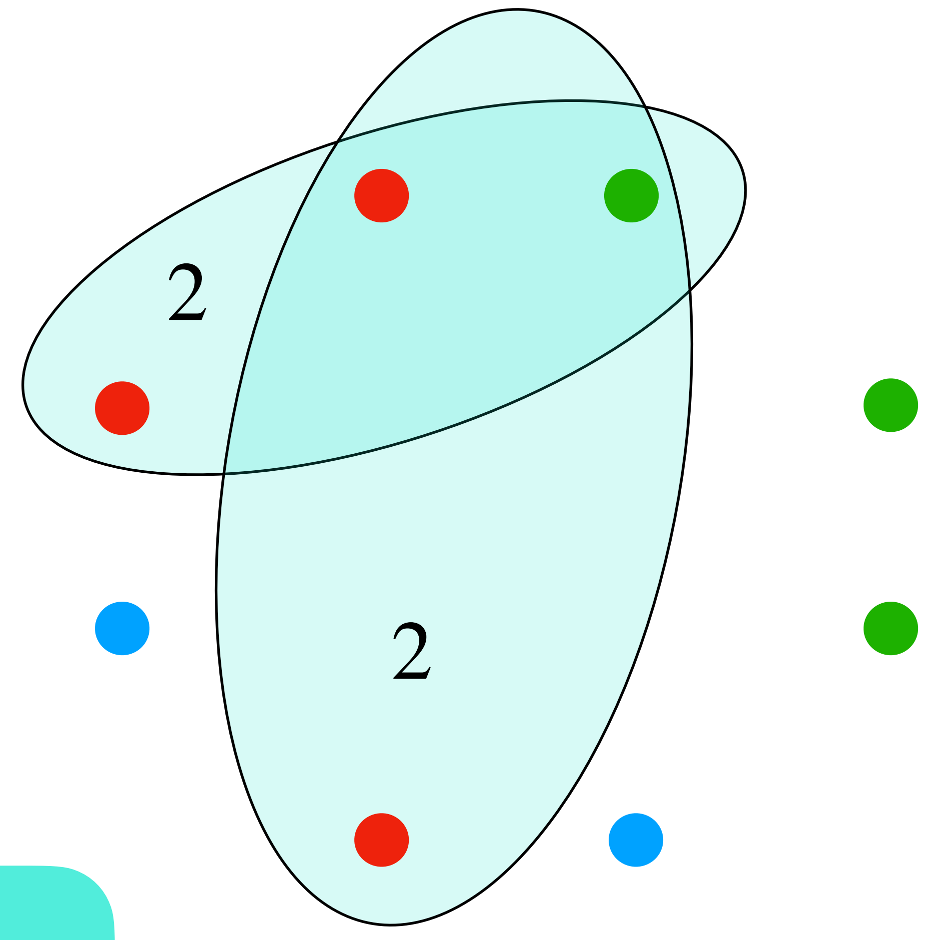
Strong

Returns full description of partition on S

$\Omega(n^2/s^2)$ $\xrightarrow{\text{+ info theory}}$ $\Omega(n^2/s^2 + n)$

Weak

Returns # clusters intersecting S



Question

When $s \leq \sqrt{n}$, are weak queries just as useful as strong queries?

Question

Is the information-theoretic optimum attainable with only $\sqrt{n}$-sized queries?

Yes!* Despite, exponentially less information from weak queries

* Up to $\log$-factors

Theorem (non-adaptive)

$O(n^2/s^2)$ **strong** queries for all $s \leq n$

Theorem (non-adaptive)

$\widetilde{O}(n^2/s^2)$ **weak** queries for all $s \leq \sqrt{n}$

General theorems for r-rounds, s-size

Theorem (strong queries)

$$\Theta\left(\max\left(\frac{n^{1+\frac{1}{2^r-1}}k^{1-\frac{1}{2^r-1}}}{s^2}, \frac{n}{s}\right)\right)$$

Theorem (weak queries)

$$\widetilde{\Theta}\left(\max\left(\frac{n^{1+\frac{1}{2^r-1}}k^{1-\frac{1}{2^r-1}}}{s^2}, n\right)\right)$$

Info-theory bounds

Equal for s up until info-theory bound is reached for weak queries:

$$s \leq \sqrt{n^{\frac{1}{2^r-1}} \cdot k^{1-\frac{1}{2^r-1}}}$$

ENCORE

Summary

- We revisit the classic problem of partition learning with pair-wise queries / crowdsource clustering
    - Obtain tight bounds in terms of **round-complexity**
    - Practical consideration: **query parallelization**
- Consider generalized **subset** queries
    - Obtain tight bounds in terms of **allowed query size**
    - Practical consideration: large queries infeasible
    - Up to reasonable size threshold:
        - Oracle that **counts** # intersected clusters “as useful” as oracle that returns entire clustering

<u>Unexplored direction</u>
What is the right **noise model** for subset queries?

ENCORE