Parameter-free Locality Sensitive Hashing for Spherical Range Reporting

Thomas D. Ahle, Martin Aumüller, Rasmus Pagh

IT University of Copenhagen

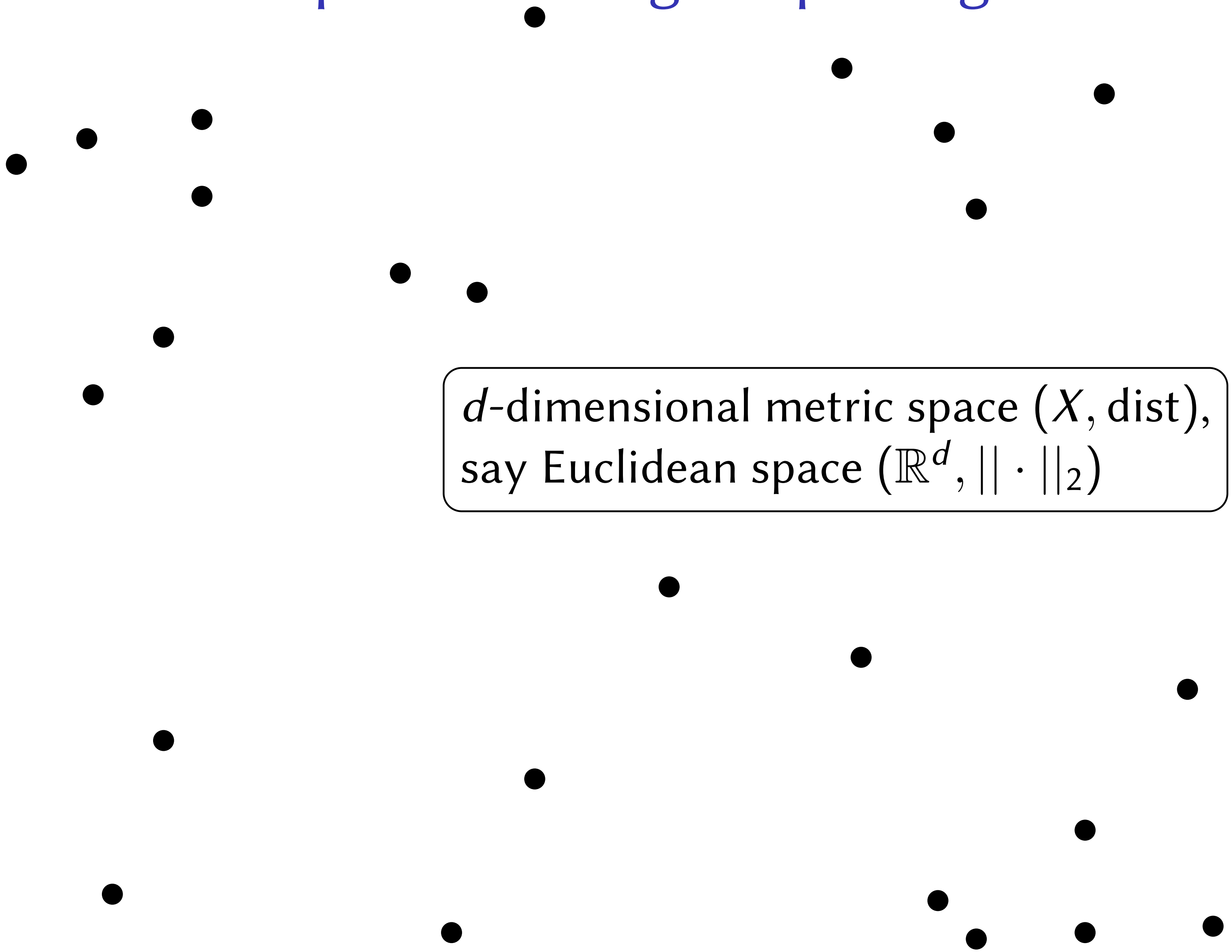
Jan 16, 2017

SODA 2017

Supported by

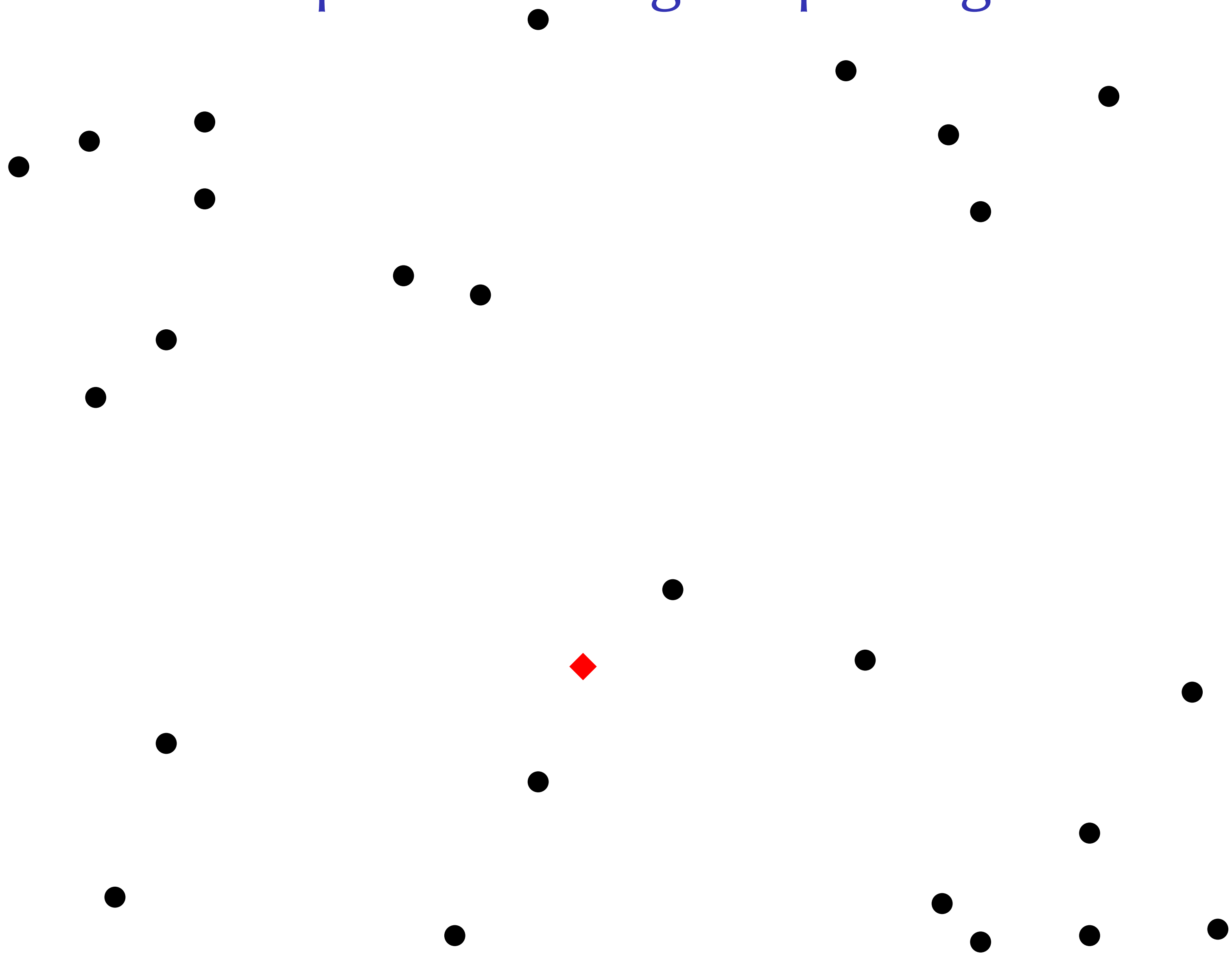
European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world

r-Spherical Range Reporting

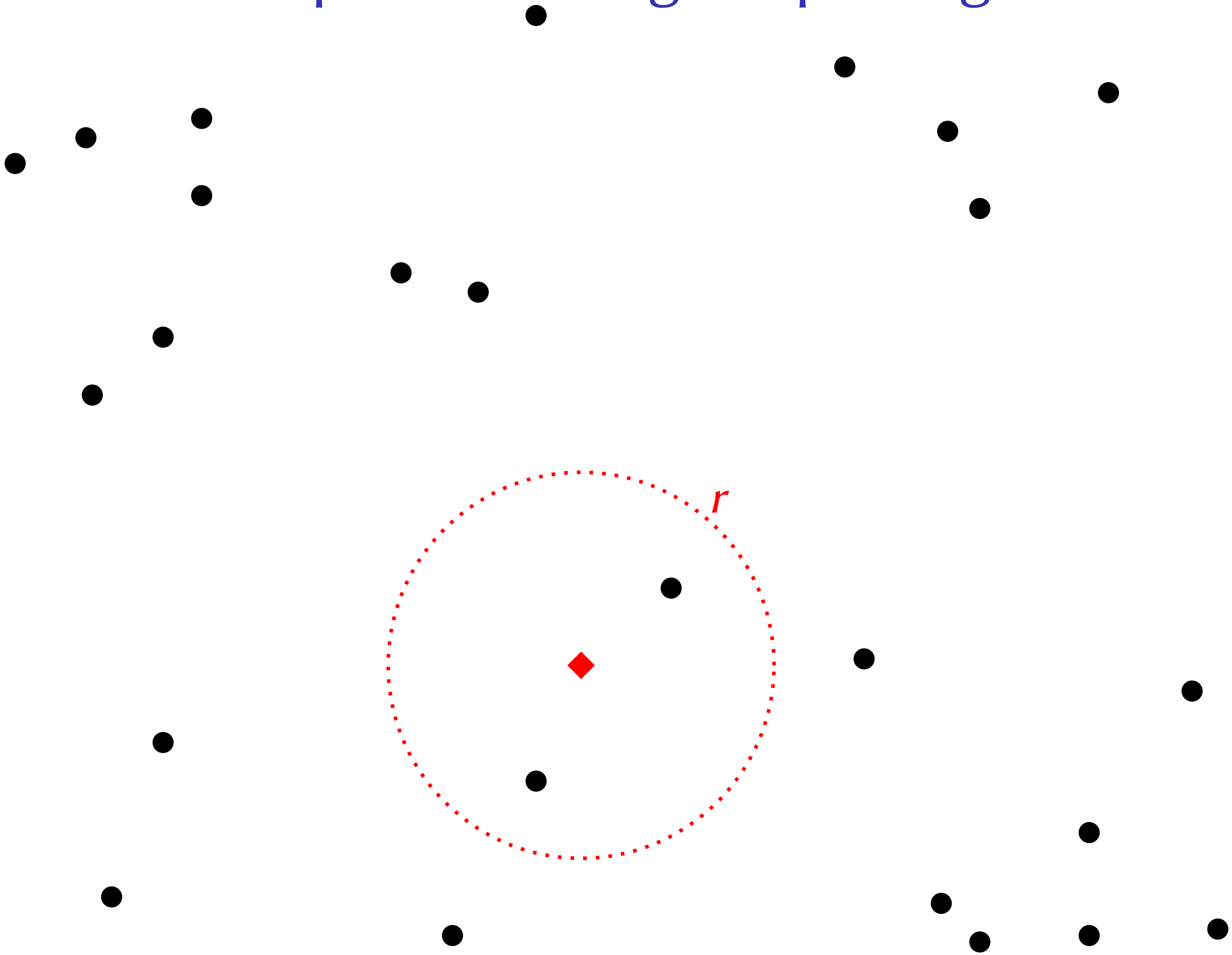


d-dimensional metric space (X, dist),
say Euclidean space $(\mathbb{R}^d, ||\cdot||_2)$

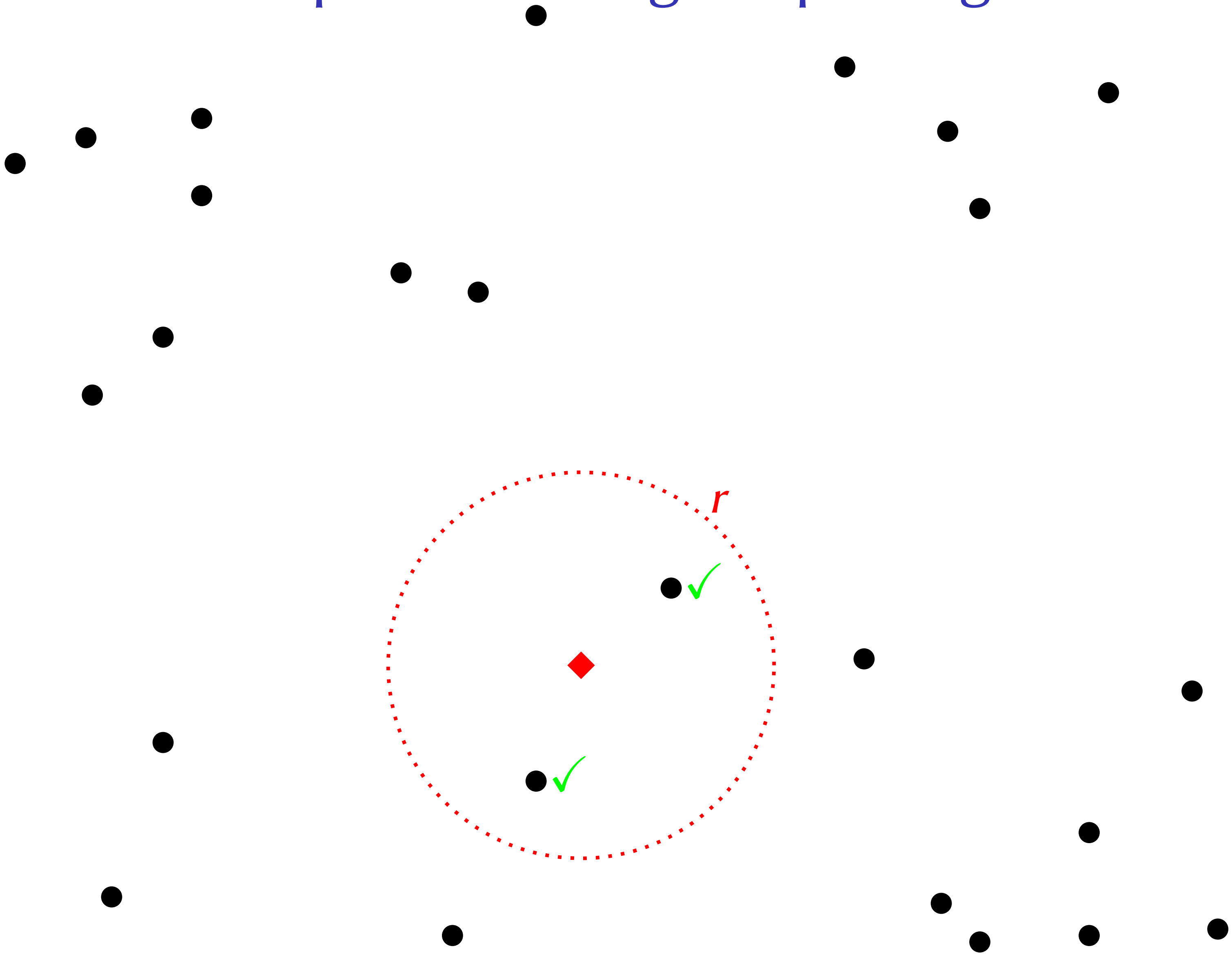
r-Spherical Range Reporting



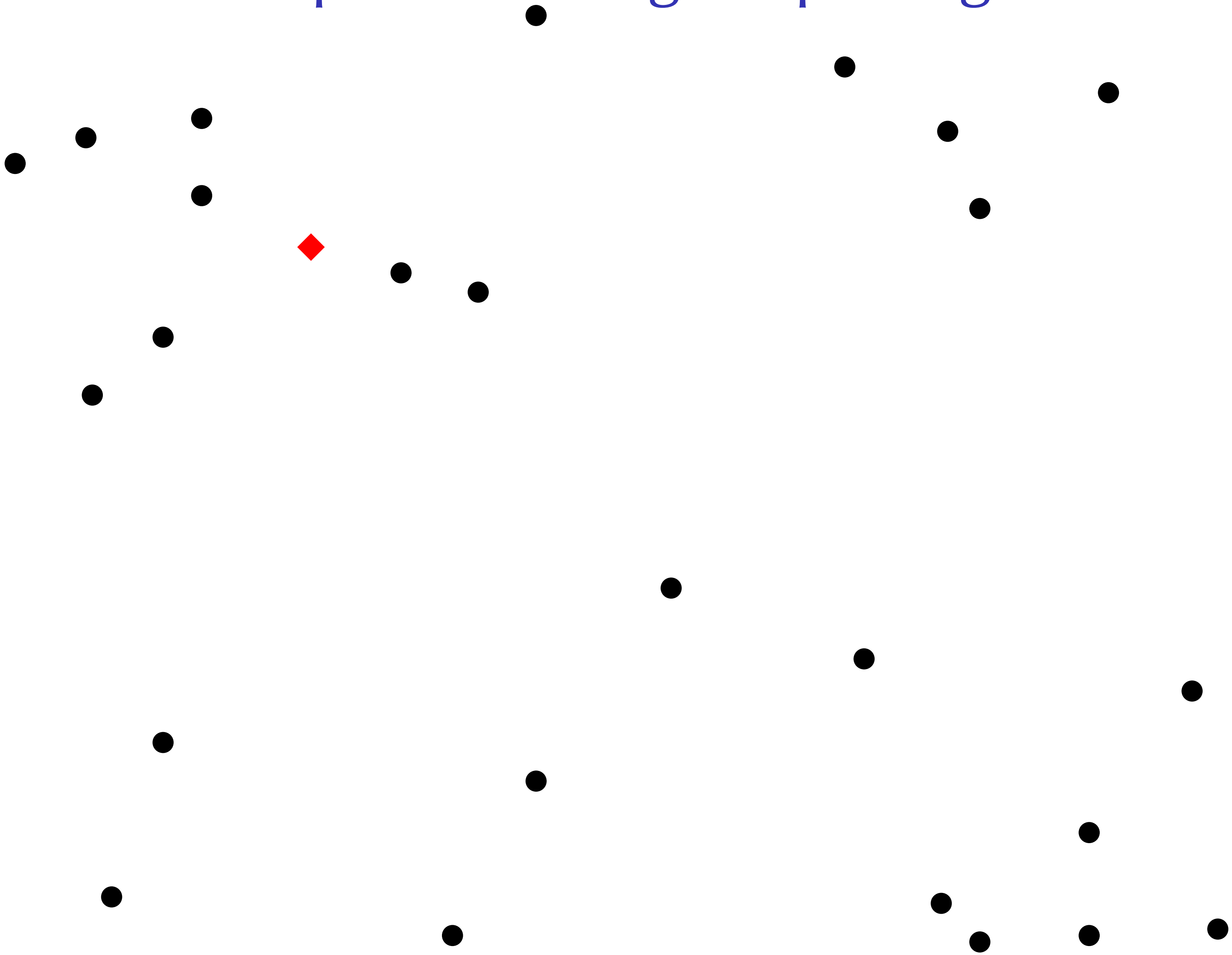
r-Spherical Range Reporting



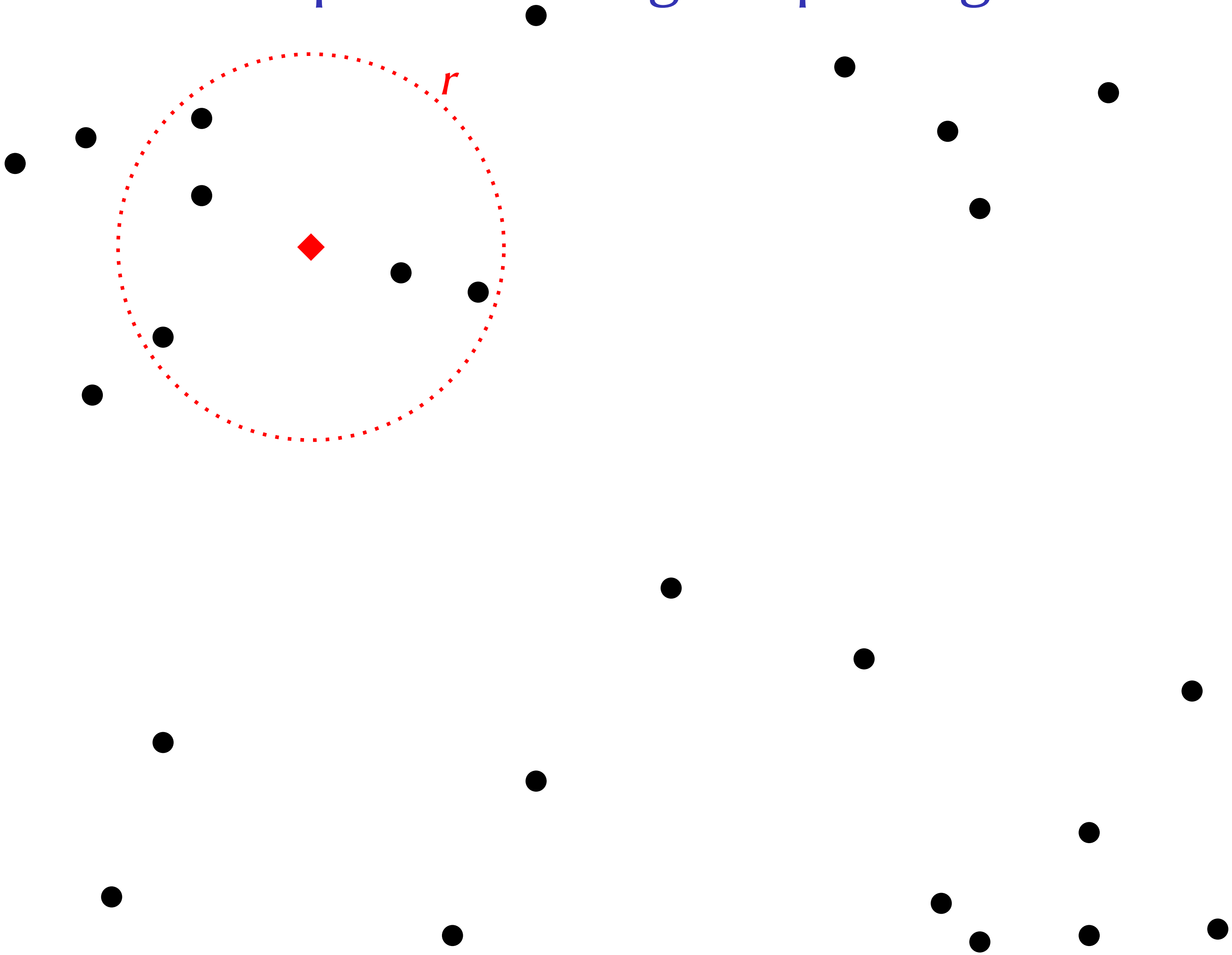
r-Spherical Range Reporting



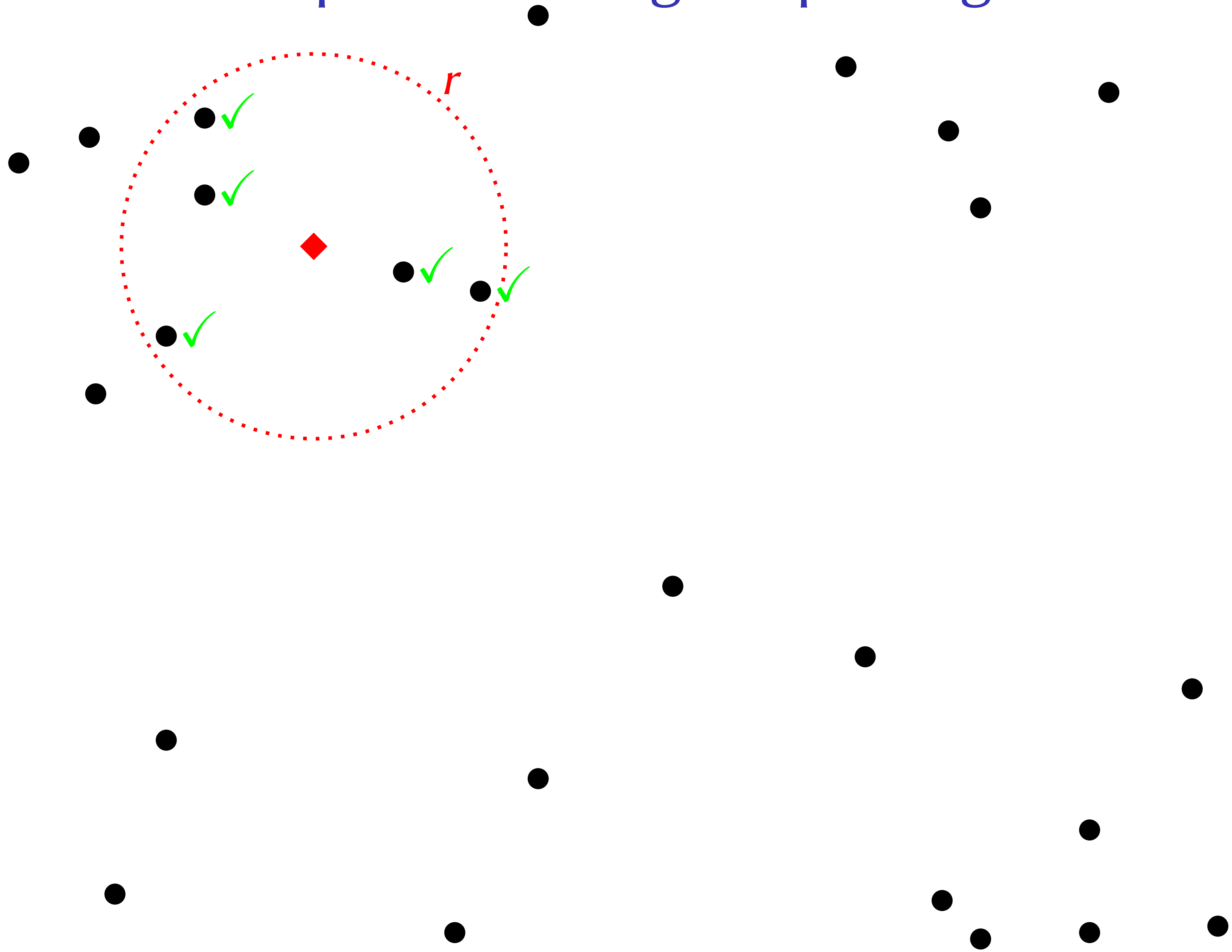
r-Spherical Range Reporting



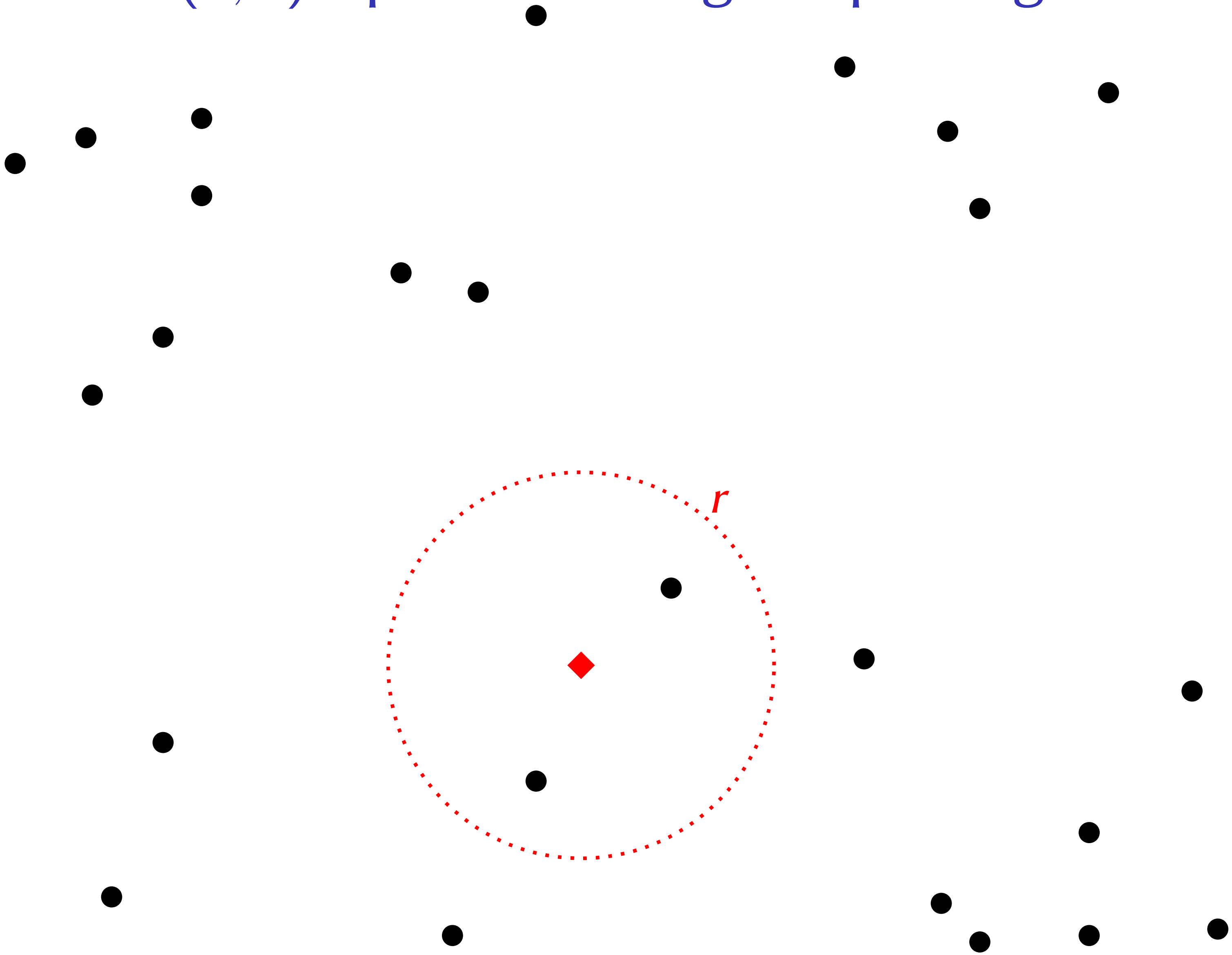
r-Spherical Range Reporting



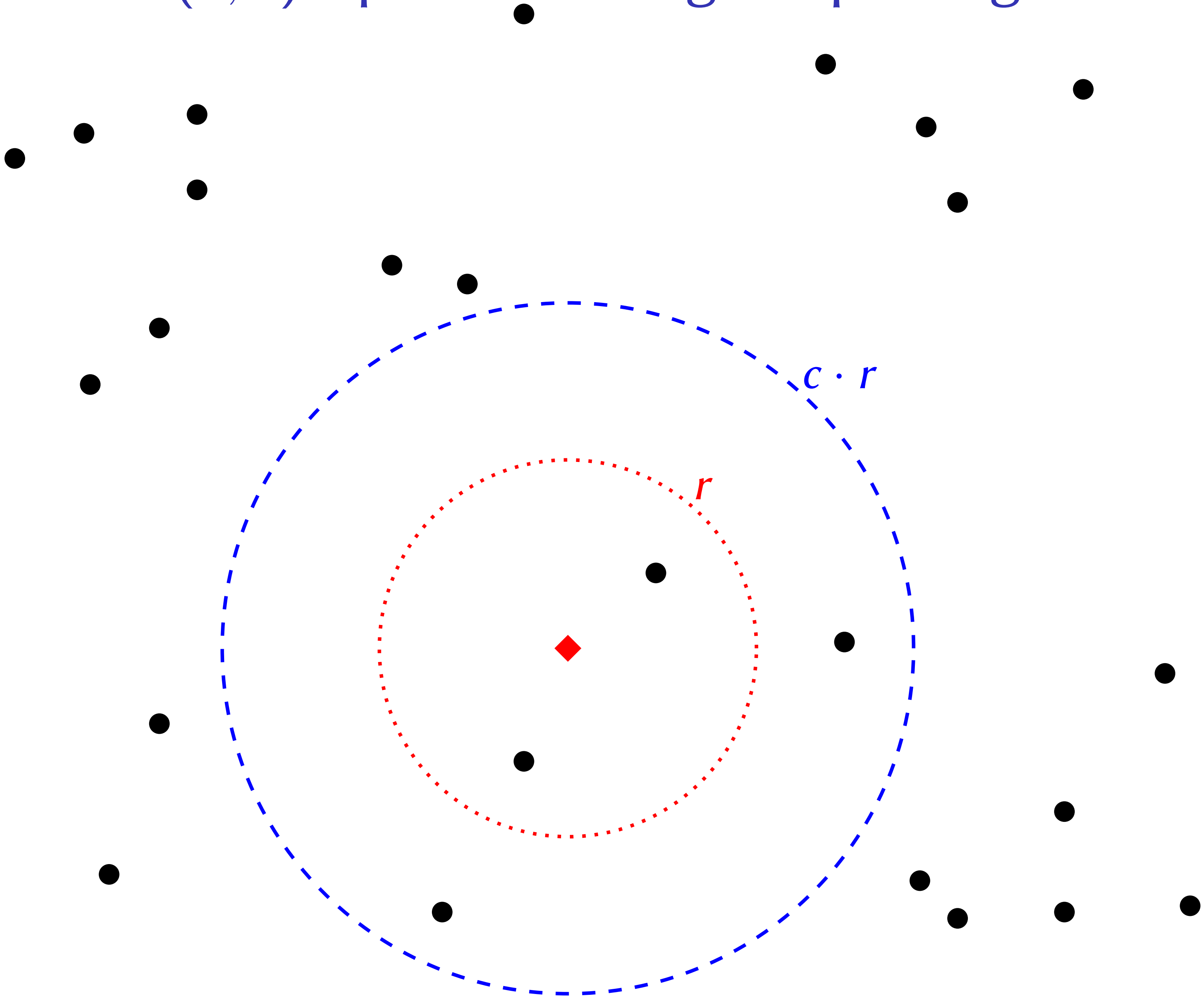
r-Spherical Range Reporting



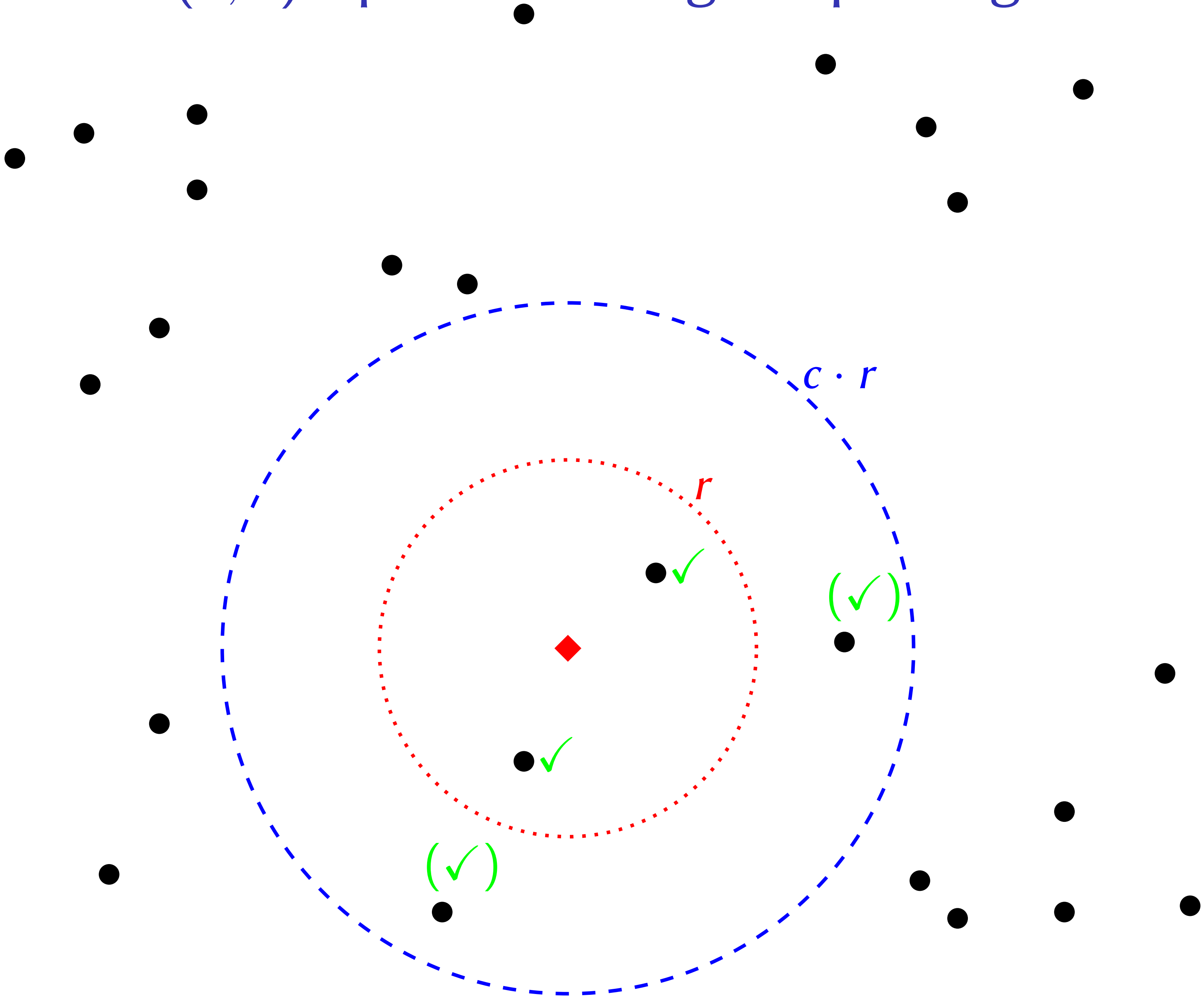
(c, r)-Spherical Range Reporting



(c, r)-Spherical Range Reporting



(c, r)-Spherical Range Reporting

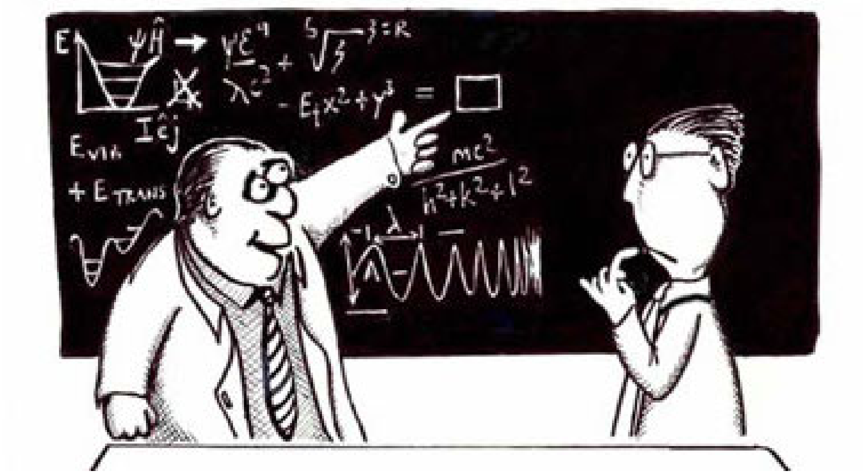


Related Work & Difficulty

- Curse of dimensionality: Solving exact problem takes linear time (brute-force scan) or space/time exponential in dimension
- small-medium dimensions:
  - tree-based approaches (Arya et al., 2010)
  - space and/or running time $O((1/\varepsilon)^d)$ for $1+\varepsilon$ approximation
- high dimensions:
  - "locality-sensitive hashing (LSH)"-based approaches for reporting (Indyk, 2000), (Andoni, 2009)
  - ρ parameter tied to the LSH family, e.g., $1/c^2$ for Euclidean space
  - Running time: $O(n^\rho \cdot t)$ for output size t
  - Space: $O(n^{1+\rho})$

Adaptive LSH Algorithms

LSH Theory vs LSH Practice



“You need $n^\rho = 513$ repetitions!”

“10 repetitions work just as fine!”

Want

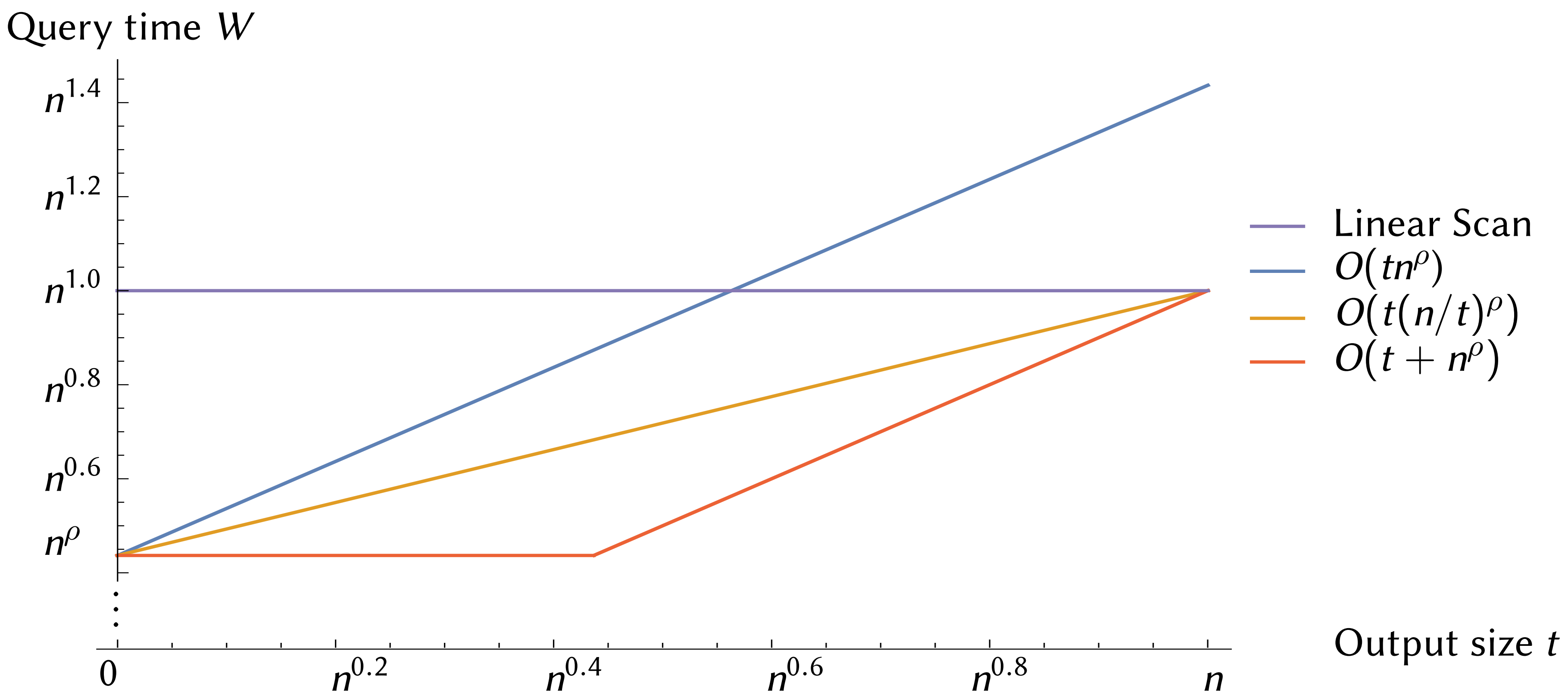
- algorithm adapts to query
- has theoretical guarantees
- use available space best

Recent related work: (Har-Peled & Mahabadi, SODA 2017)

Our Results (in presentation)

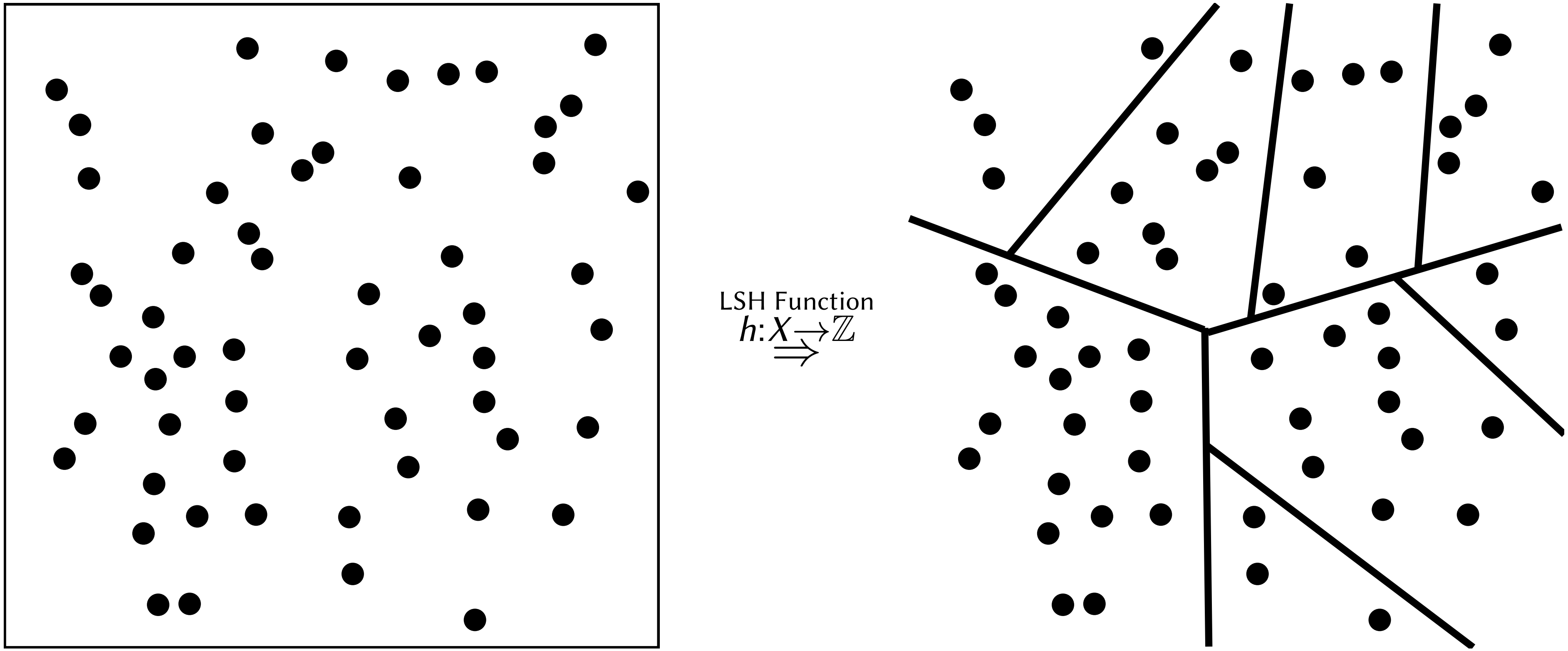
1. **Oracle access to output size *t***
   → can solve Spherical Range Reporting using LSH in time $O(t(n/t)^\rho)$
2. **No oracle?**
   → "Multi-Level LSH" with adaptive query algorithm finds "best LSH parameters" with little overhead

Plot of Running Times



Standard LSH Approach: LSH Function

- random space partition

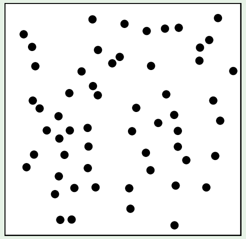


- Characteristics:
  - p_1: (lower bound on) collision probability of two points at distance $\leq r$
  - p_2: (upper bound on) collision probability of two points at distance $\geq cr$
  - strength of the LSH: $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$

Standard LSH Approach: LSH Data Structure

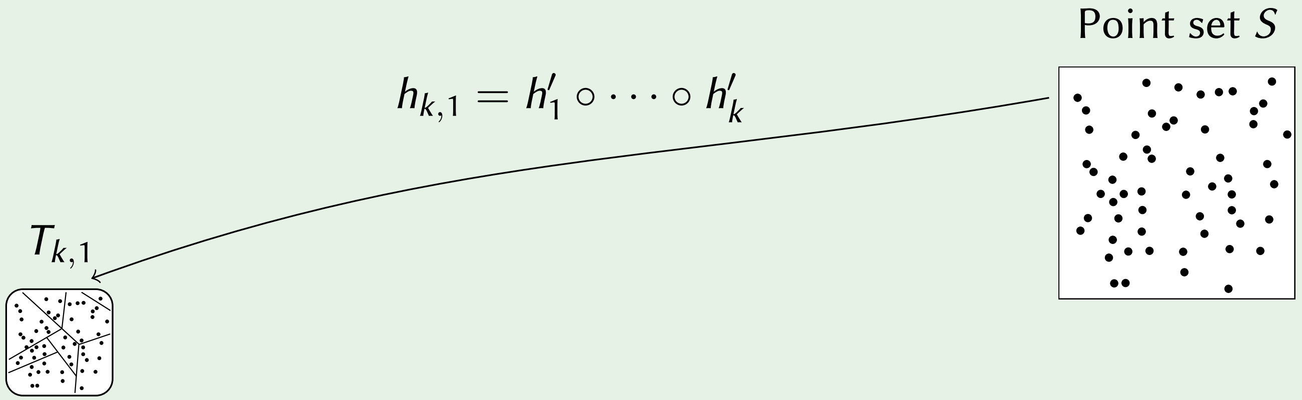
1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times

Point set S



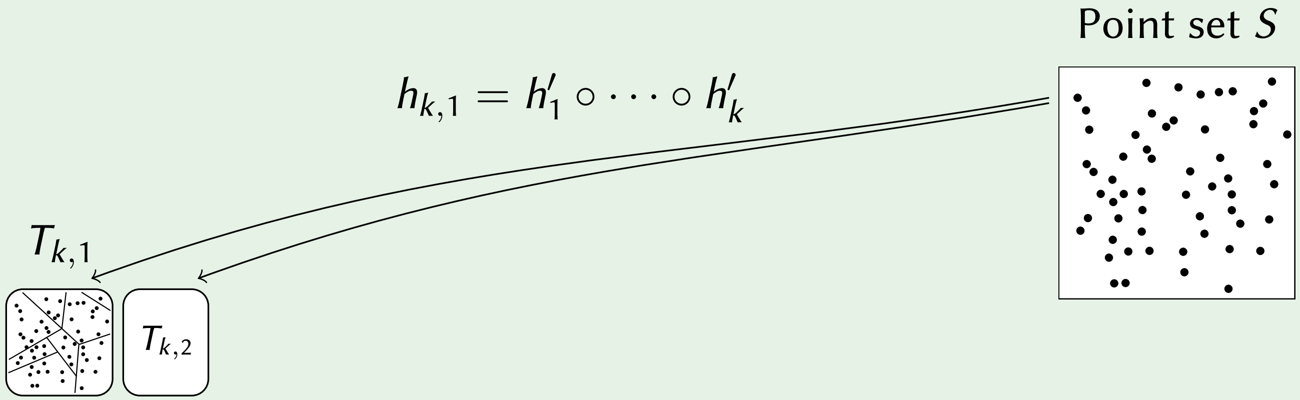
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



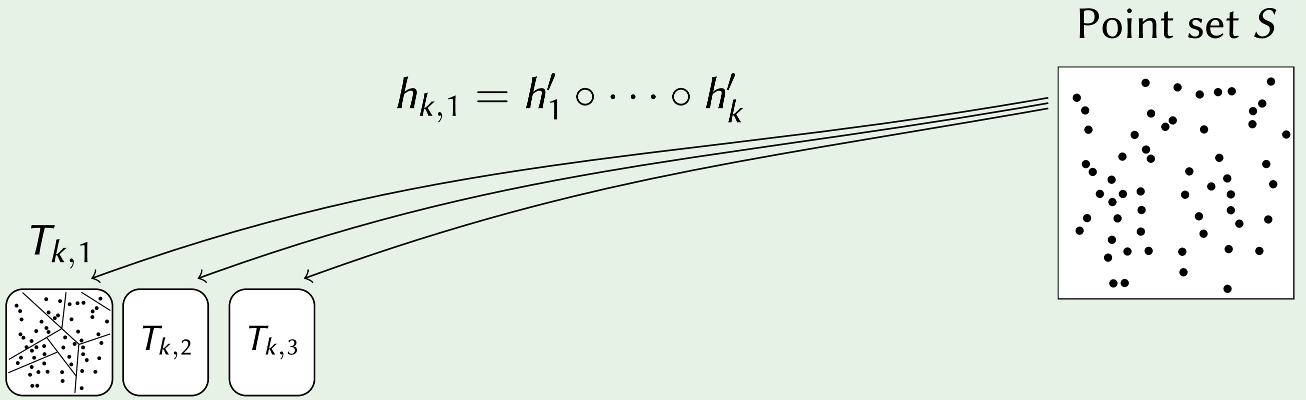
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



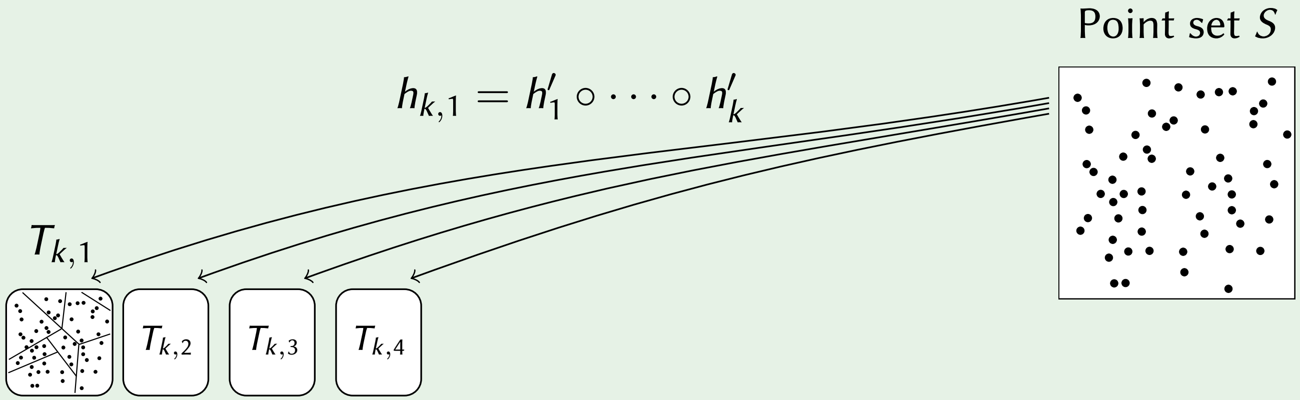
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



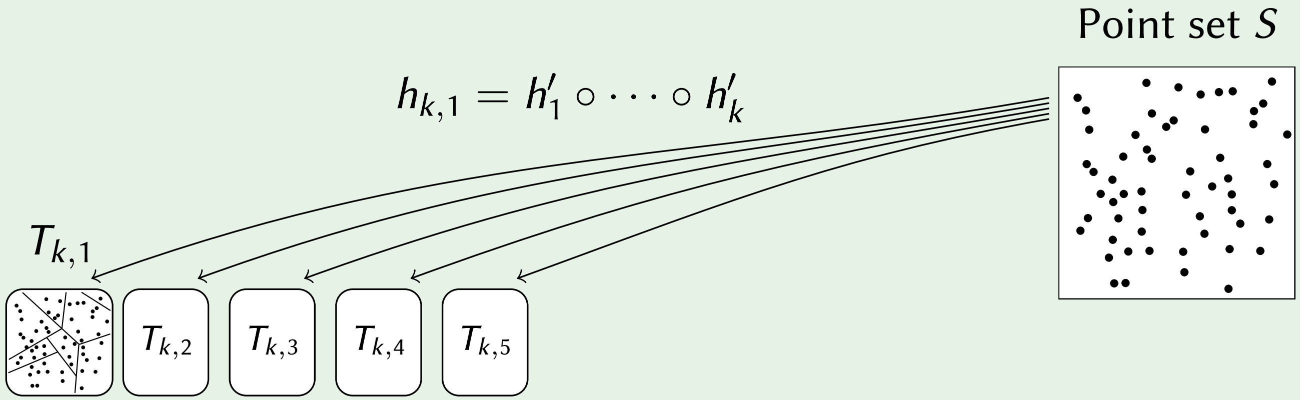
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



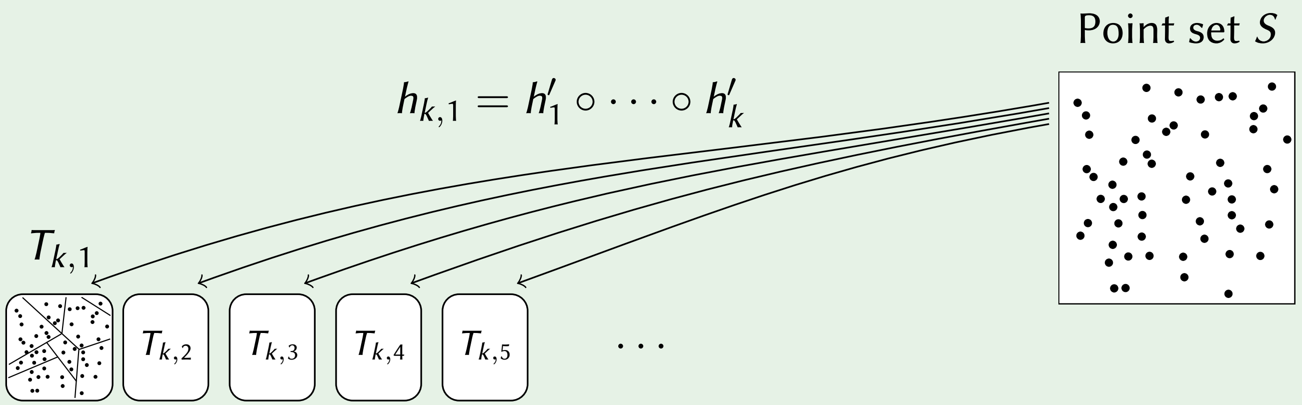
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



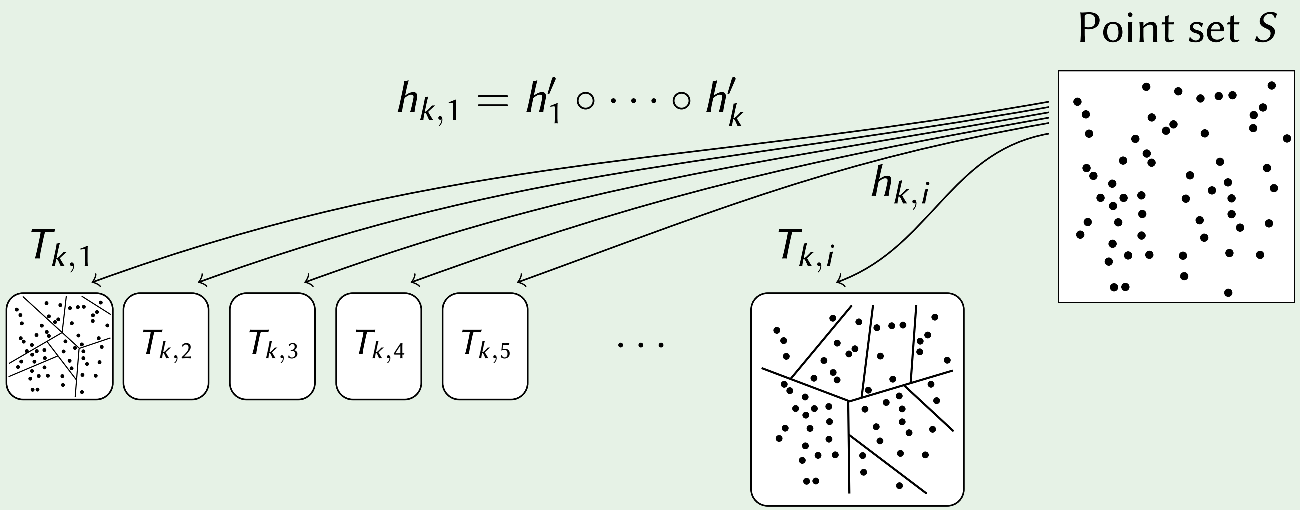
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



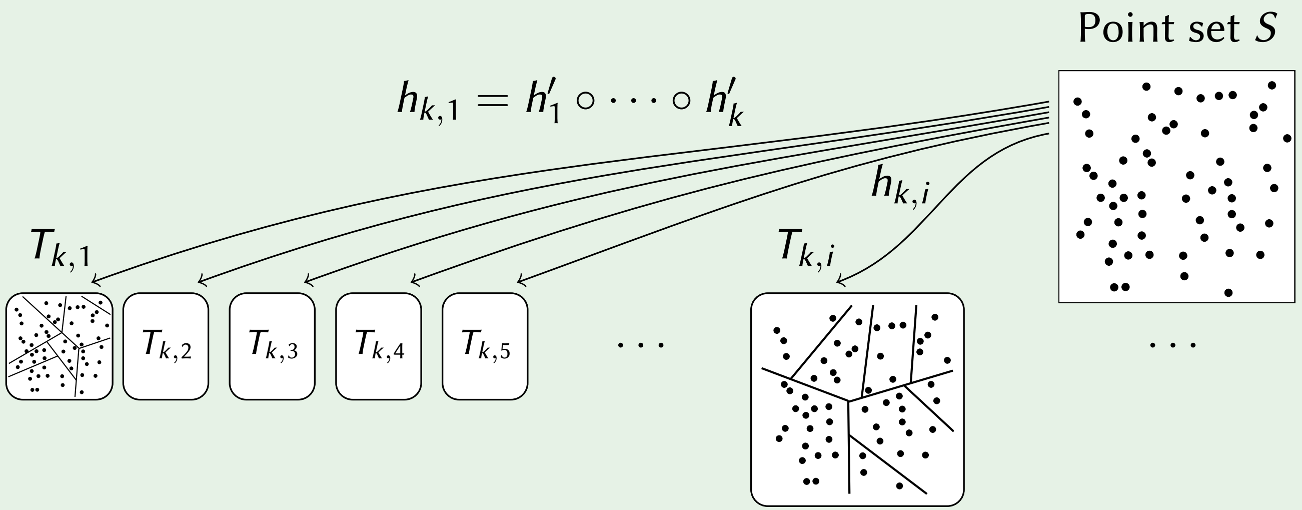
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



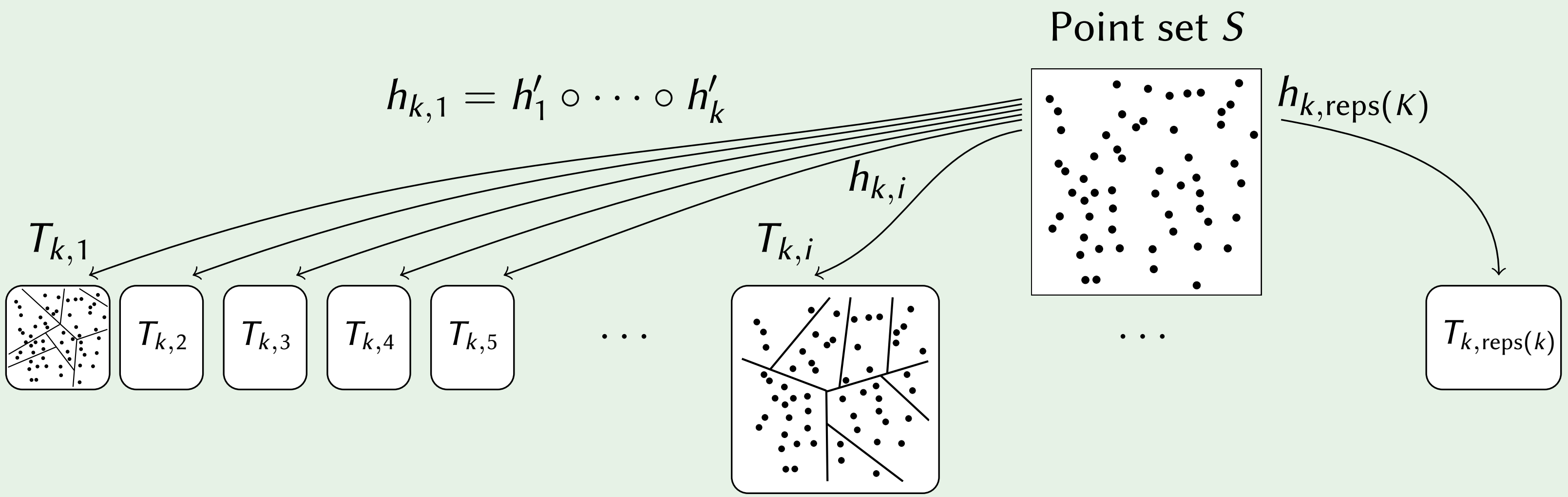
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times



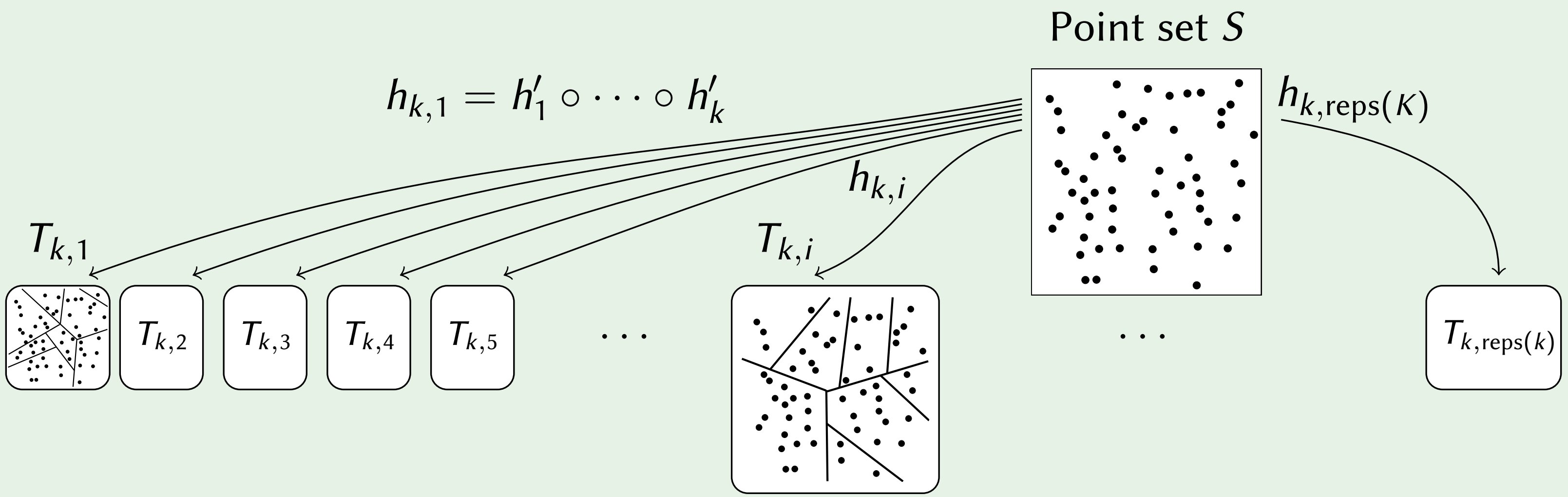
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\mathrm{reps}(k) := p_1^{-k}$ many times



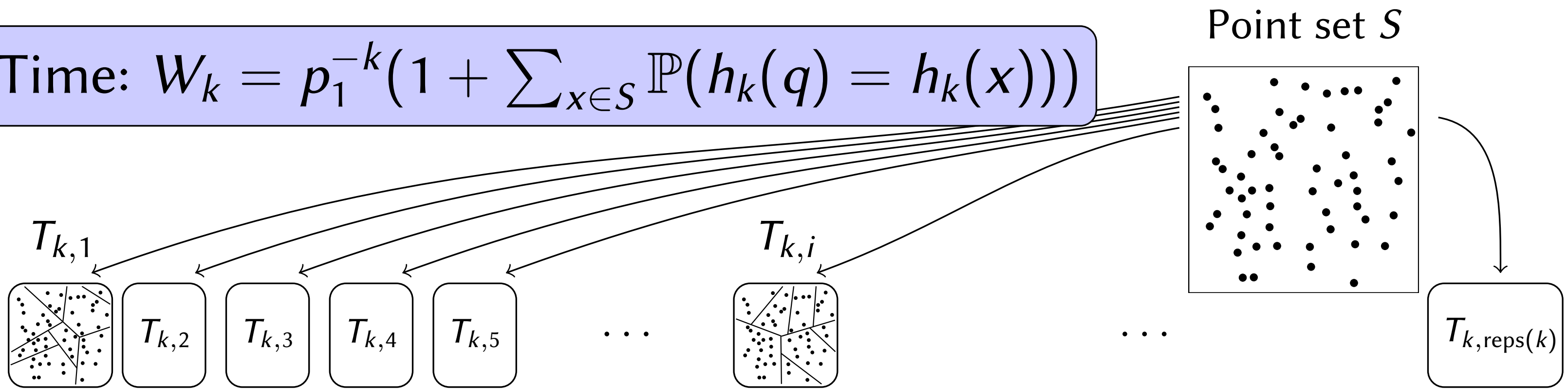
Standard LSH Approach: LSH Data Structure

1. concatenate $k \geq 1$ hash functions
2. repeat $\text{reps}(k) := p_1^{-k}$ many times

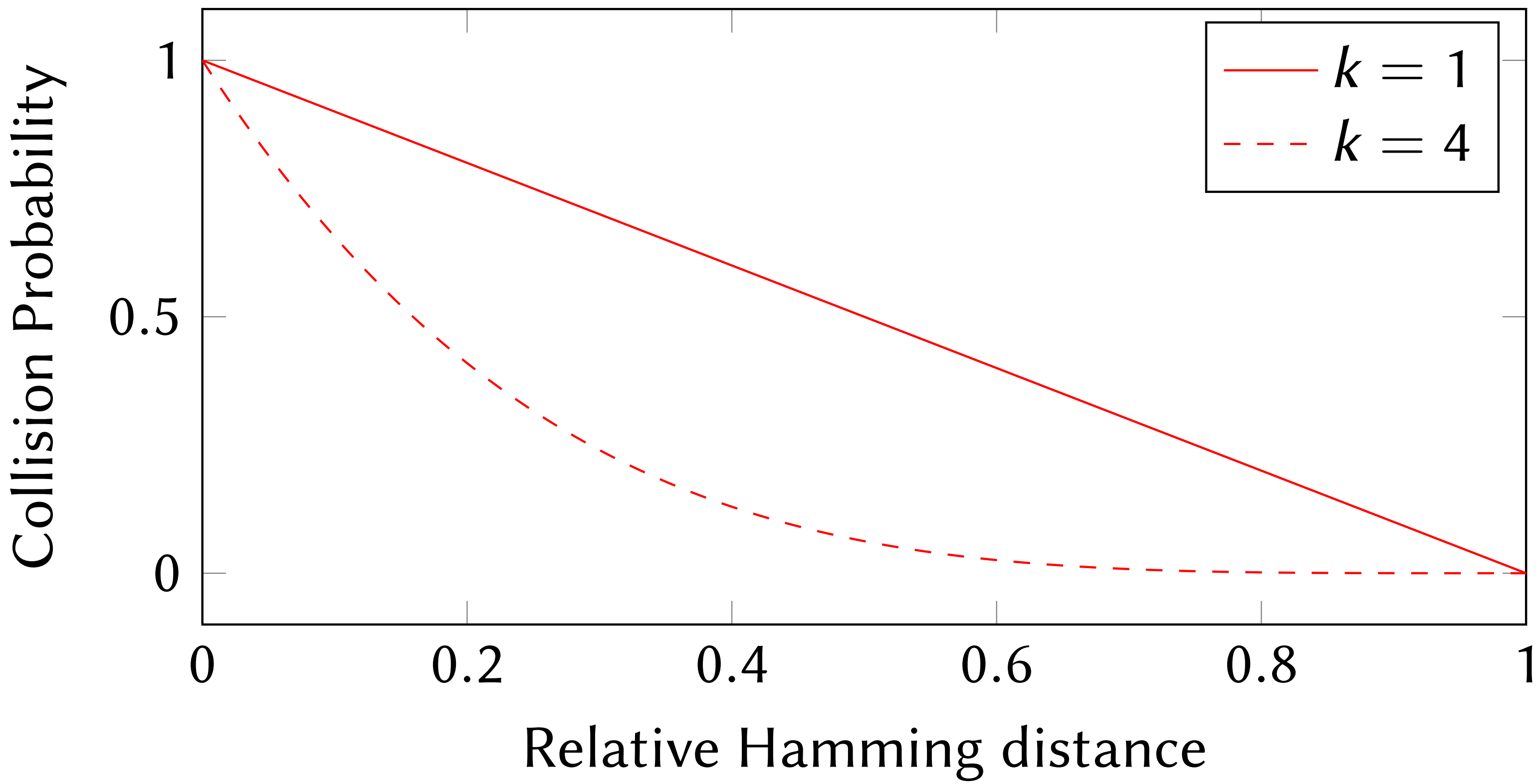


- Query algorithm: Collect all points that collide with the query over all tables, report the ones at distance at most r.
- Expected running time?

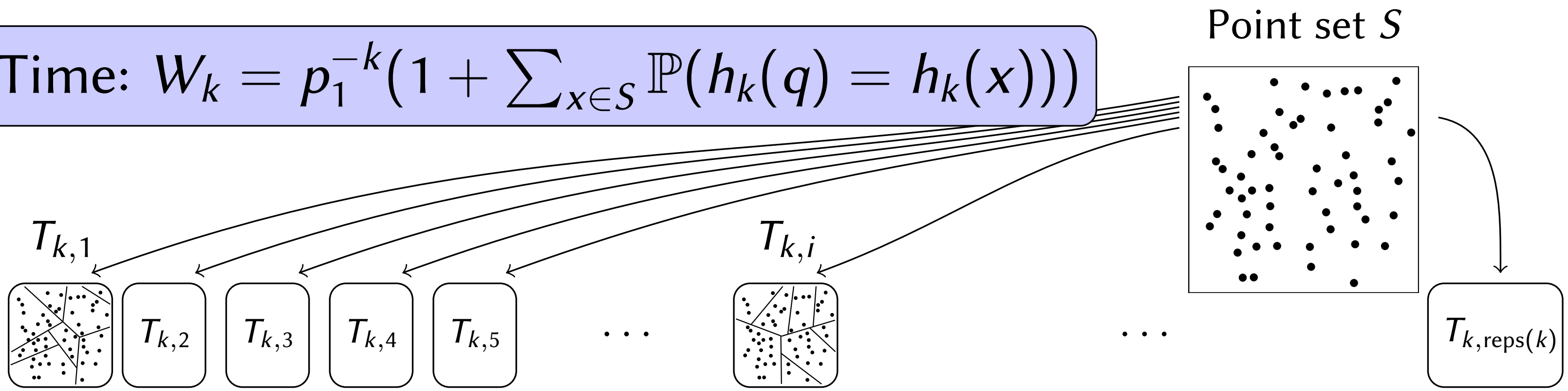
Expected Time: $W_k = p_1^{-k}(1 + \sum_{x \in S} \mathbb{P}(h_k(q) = h_k(x)))$

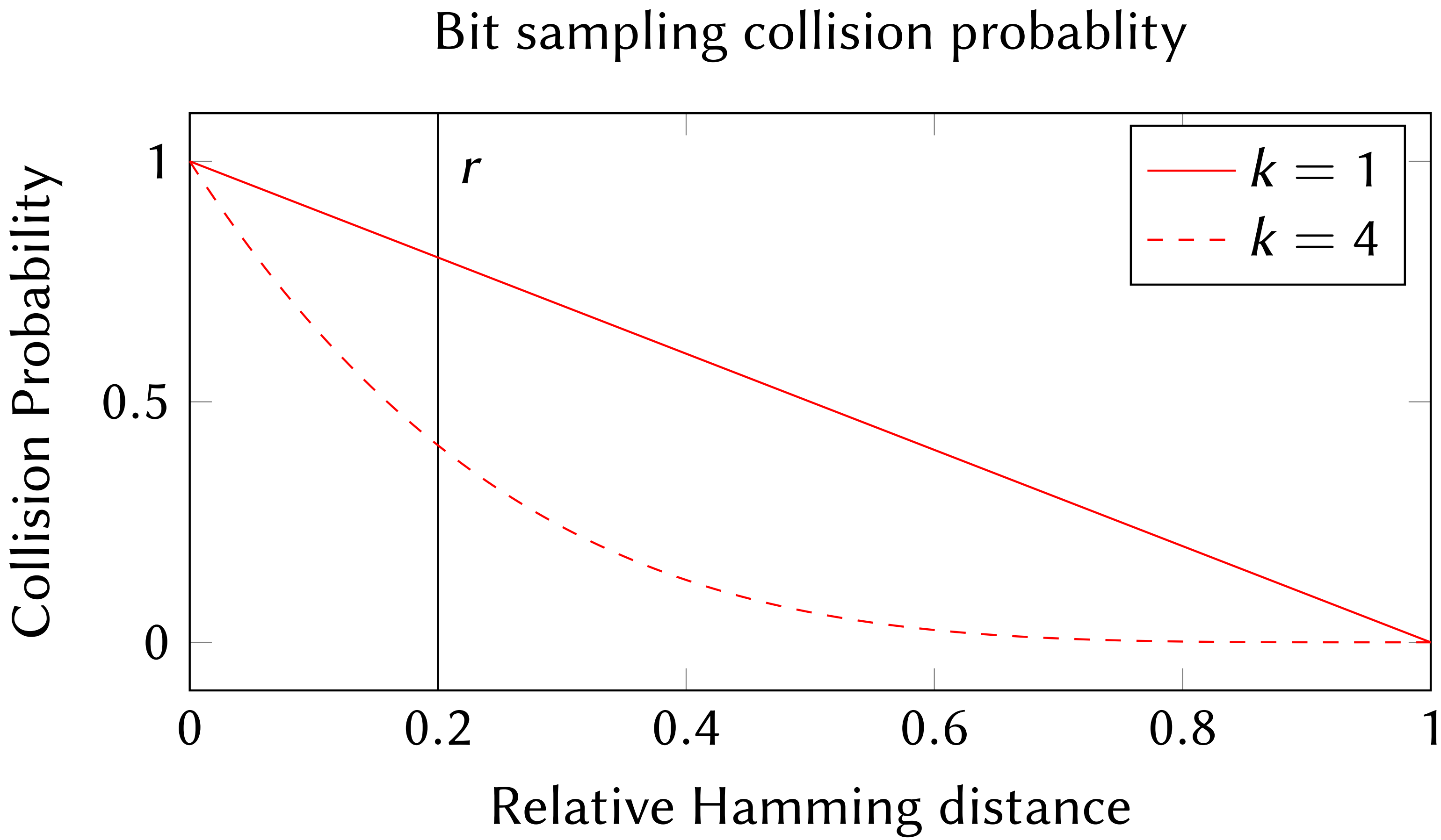


Bit sampling collision probablity

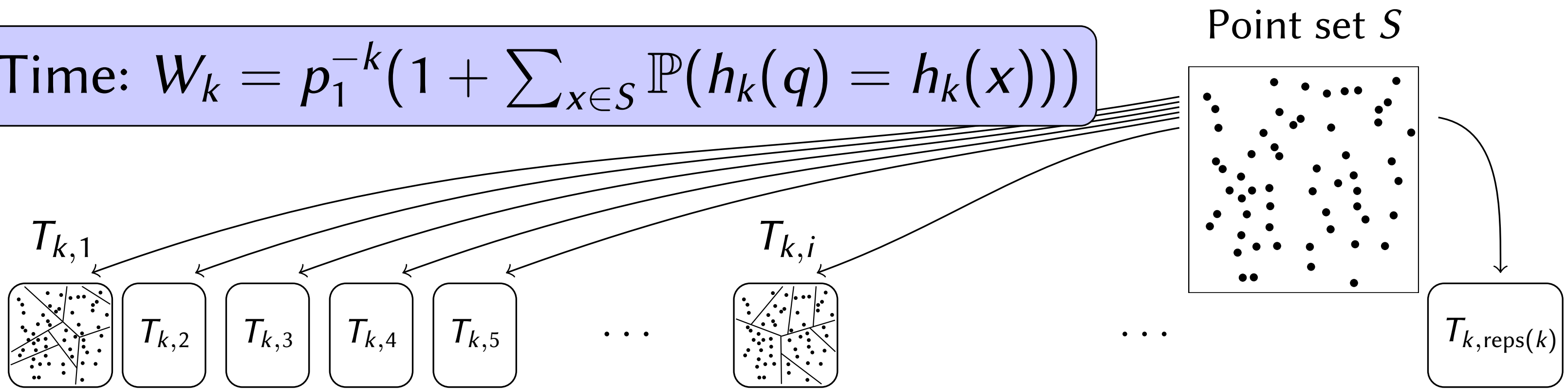


Expected Time: $W_k = p_1^{-k}(1 + \sum_{x \in S} \mathbb{P}(h_k(q) = h_k(x)))$

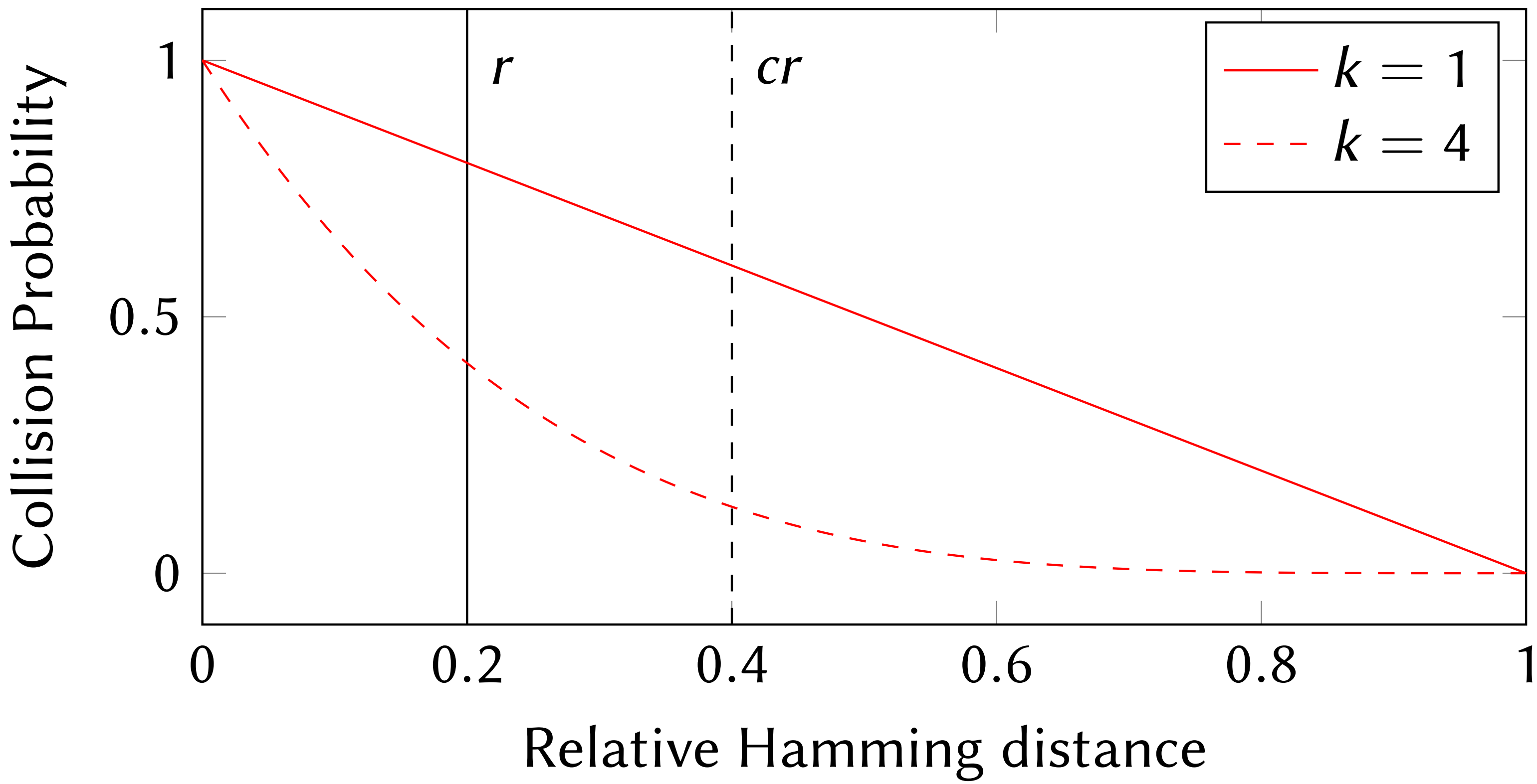




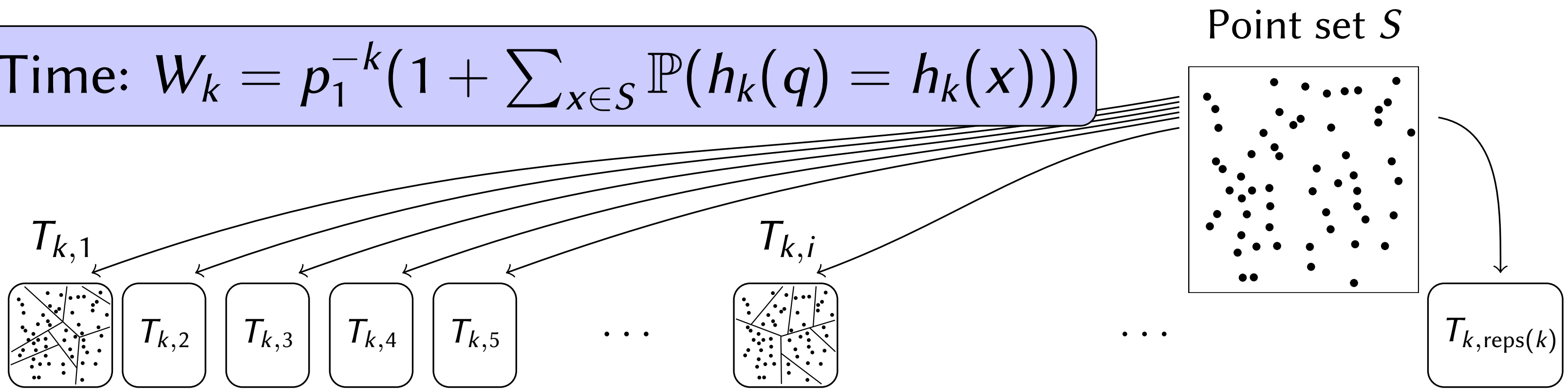
Expected Time: $W_k = p_1^{-k}(1 + \sum_{x \in S} \mathbb{P}(h_k(q) = h_k(x)))$



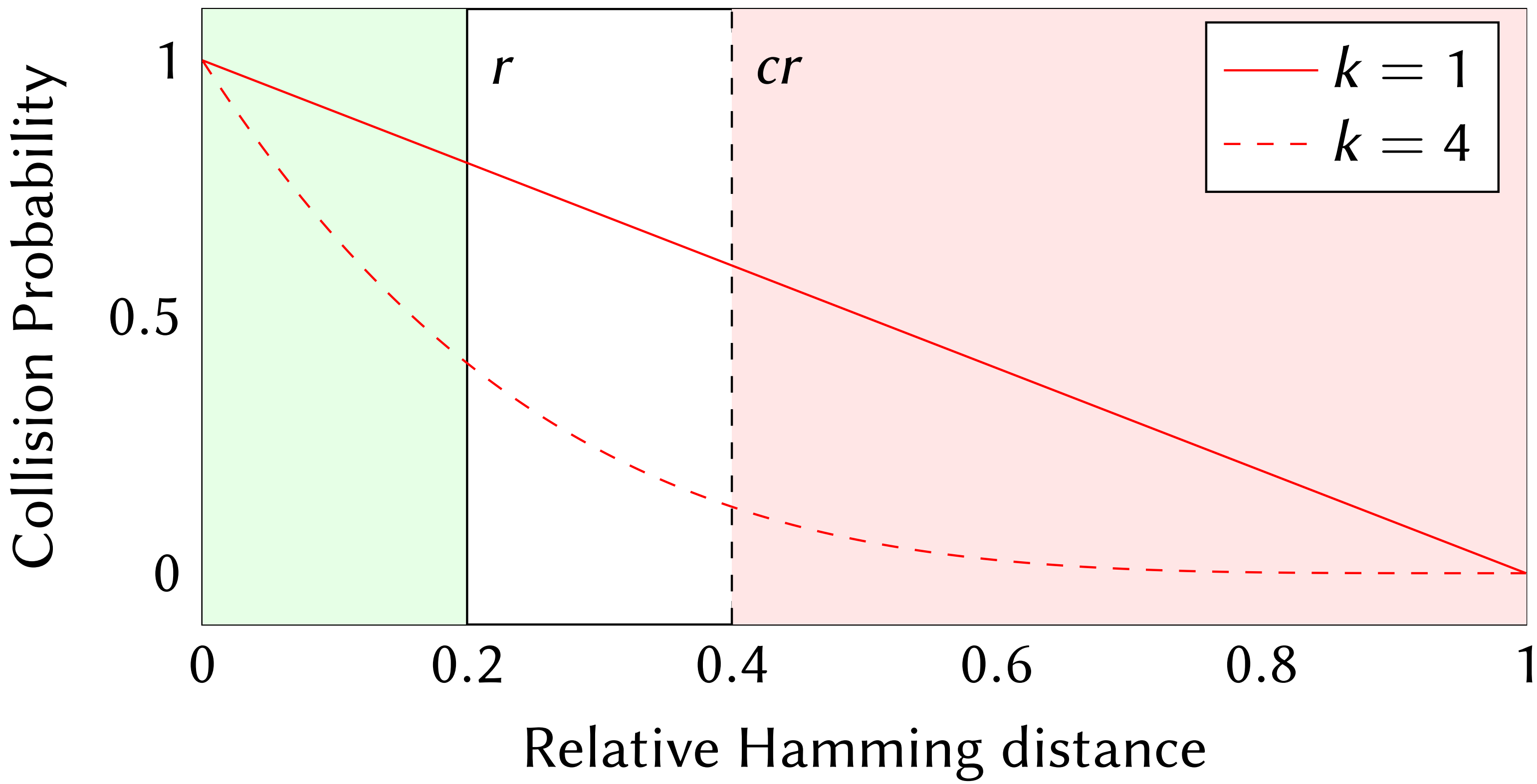
Bit sampling collision probablity



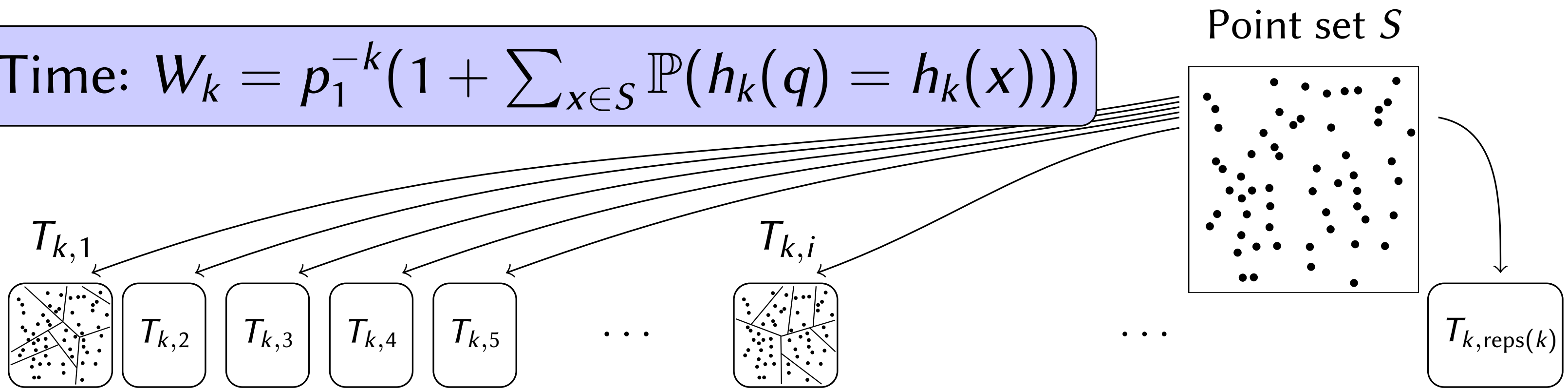
$$\text{Expected Time: } W_k = p_1^{-k}\left(1 + \sum_{x \in S} \mathbb{P}(h_k(q) = h_k(x))\right)$$



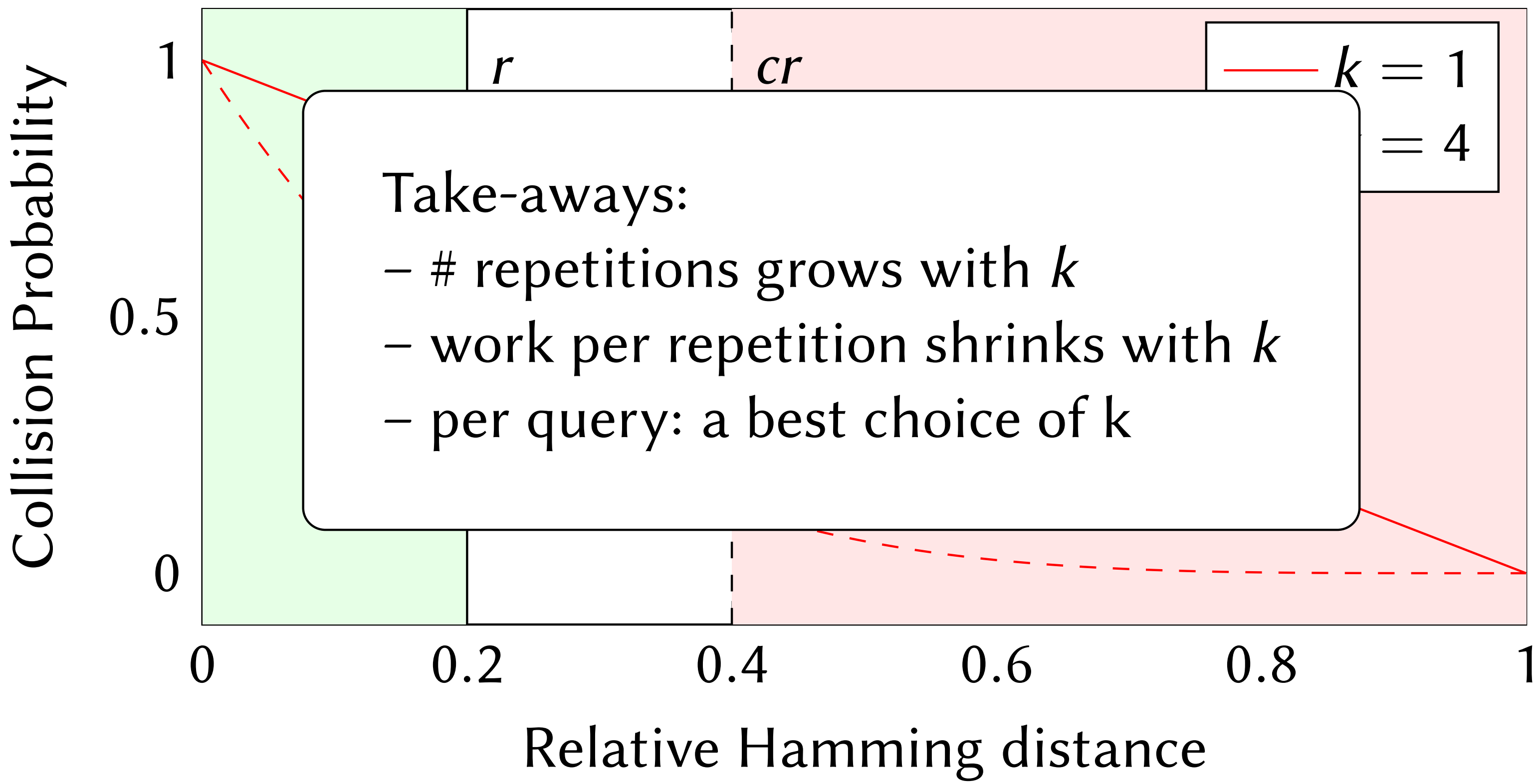
Bit sampling collision probablity



Expected Time: $W_k = p_1^{-k}(1 + \sum_{x \in S} \mathbb{P}(h_k(q) = h_k(x)))$



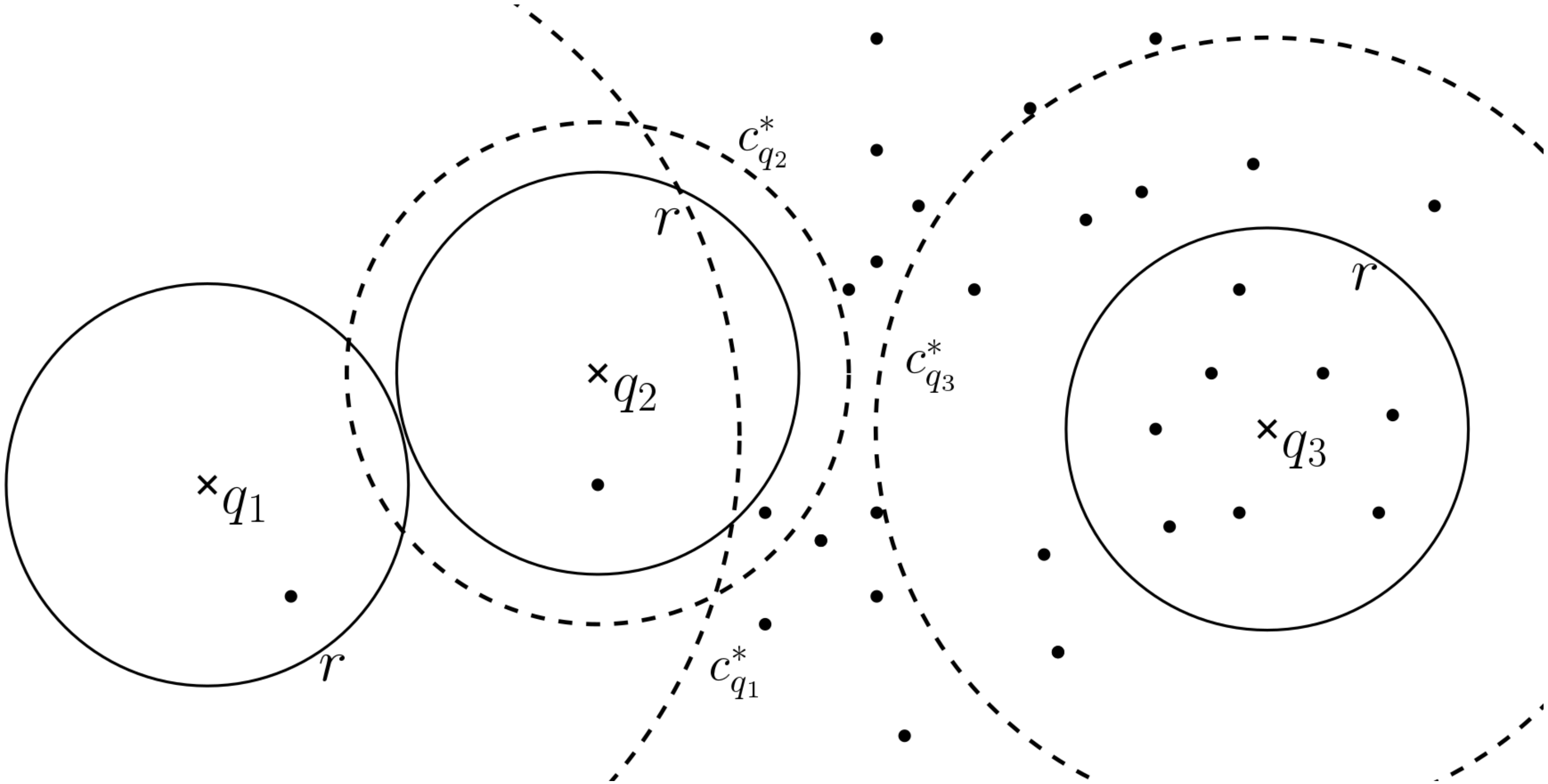
Bit sampling collision probablity



Take-aways:

- # repetitions grows with k
- work per repetition shrinks with k
- per query: a best choice of k

Expansion at the Query



Expansion c^*_q at query q: Largest c such that

#points at distance $cr \leq 2$ #points at distance r

Probing the Right Level

Lemma

Output size t & expansion $c_q^* \Rightarrow$ *there exists k such that*

$$W_k = O\left(t(n/t)^{\rho^*}\right),$$

where $\rho^* = \frac{\log(1/p_1)}{\log(1/p(c_q^* r))}$ *with* $p(c_q^* r)$ *being prob. of collision at distance* $c_q^* r$.

Probing the Right Level

Lemma

Output size t & expansion c_q^* $\Rightarrow$ there exists k such that

$$W_k = O\left(t(n/t)^{\rho^*}\right),$$

where $\rho^* = \frac{\log(1/p_1)}{\log(1/p(c_q^* r))}$ with $p(c_q^* r)$ being prob. of collision at distance $c_q^* r$.

Idea:

- expected work:

$$p_1^{-k}\Bigg(1 + \underbrace{\sum_{\substack{x\in S\\ \mathrm{dist}(x,q)\le c_q^* r}} \mathbb{P}(h_k(q) = h_k(x)) +}_{\le 2t \text{ by def. of expansion}} \sum_{\substack{x\in S\\ \mathrm{dist}(x,q)> c_q^* r}} \mathbb{P}(h_k(q) = h_k(x))\Bigg)$$

Probing the Right Level

Lemma

Output size t & expansion c_q^* $\Rightarrow$ there exists k such that

$$W_k = O\left(t(n/t)^{\rho^*}\right),$$

where $\rho^* = \frac{\log(1/p_1)}{\log(1/p(c_q^* r))}$ *with* $p(c_q^* r)$ *being prob. of collision at distance* $c_q^* r$.

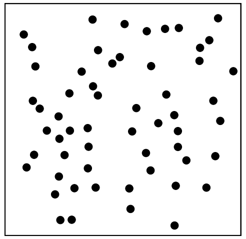
Idea:

- expected work:

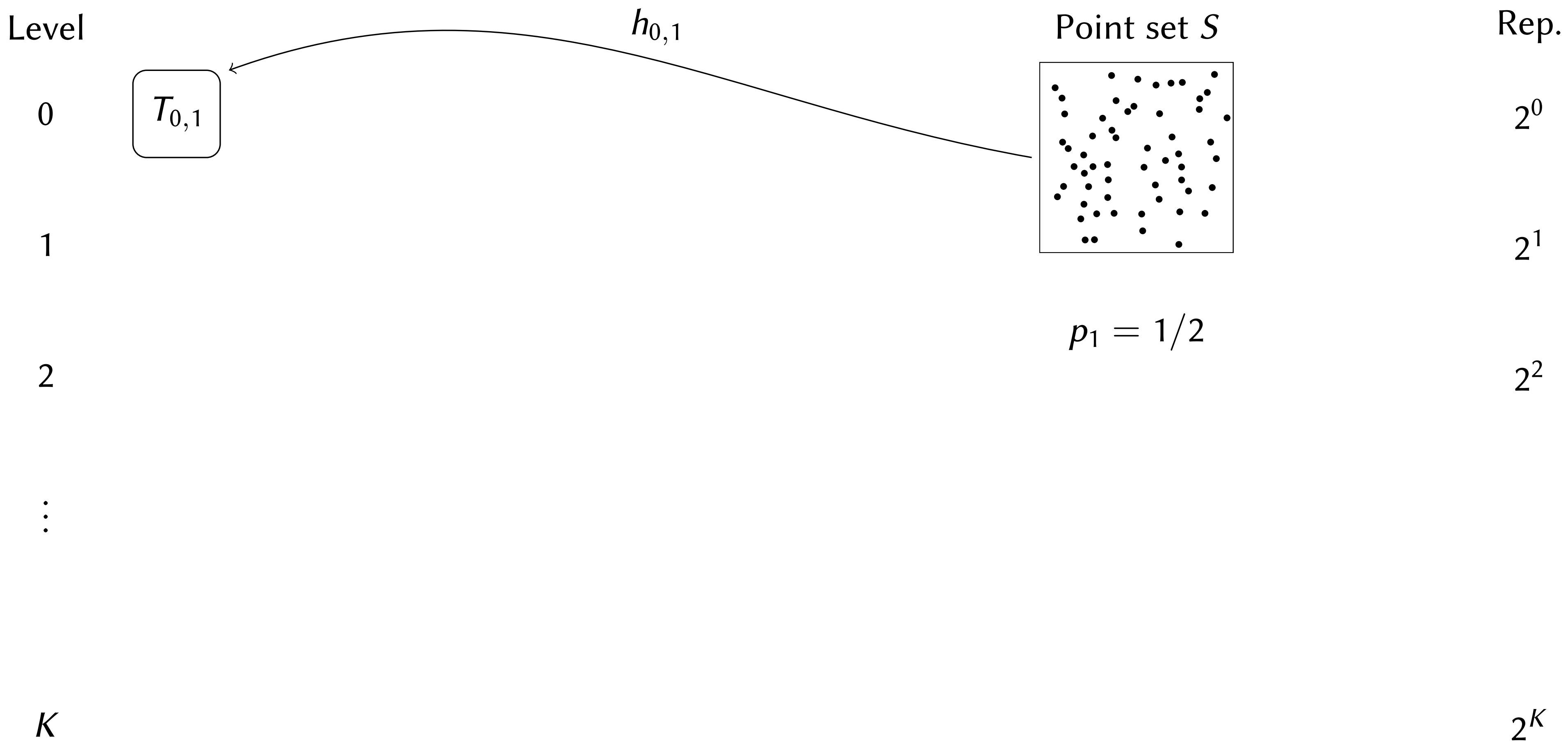
$$p_1^{-k}\Bigg(1 + \underbrace{\sum_{\substack{x \in S \\ \text{dist}(x,q) \le c_q^* r}} \mathbb{P}(h_k(q) = h_k(x)) + \sum_{\substack{x \in S \\ \text{dist}(x,q) > c_q^* r}}}_{\le 2t \text{ by def. of expansion}} \mathbb{P}(h_k(q) = h_k(x))\Bigg)$$

- set k such that $\le t$ expected collisions with far points

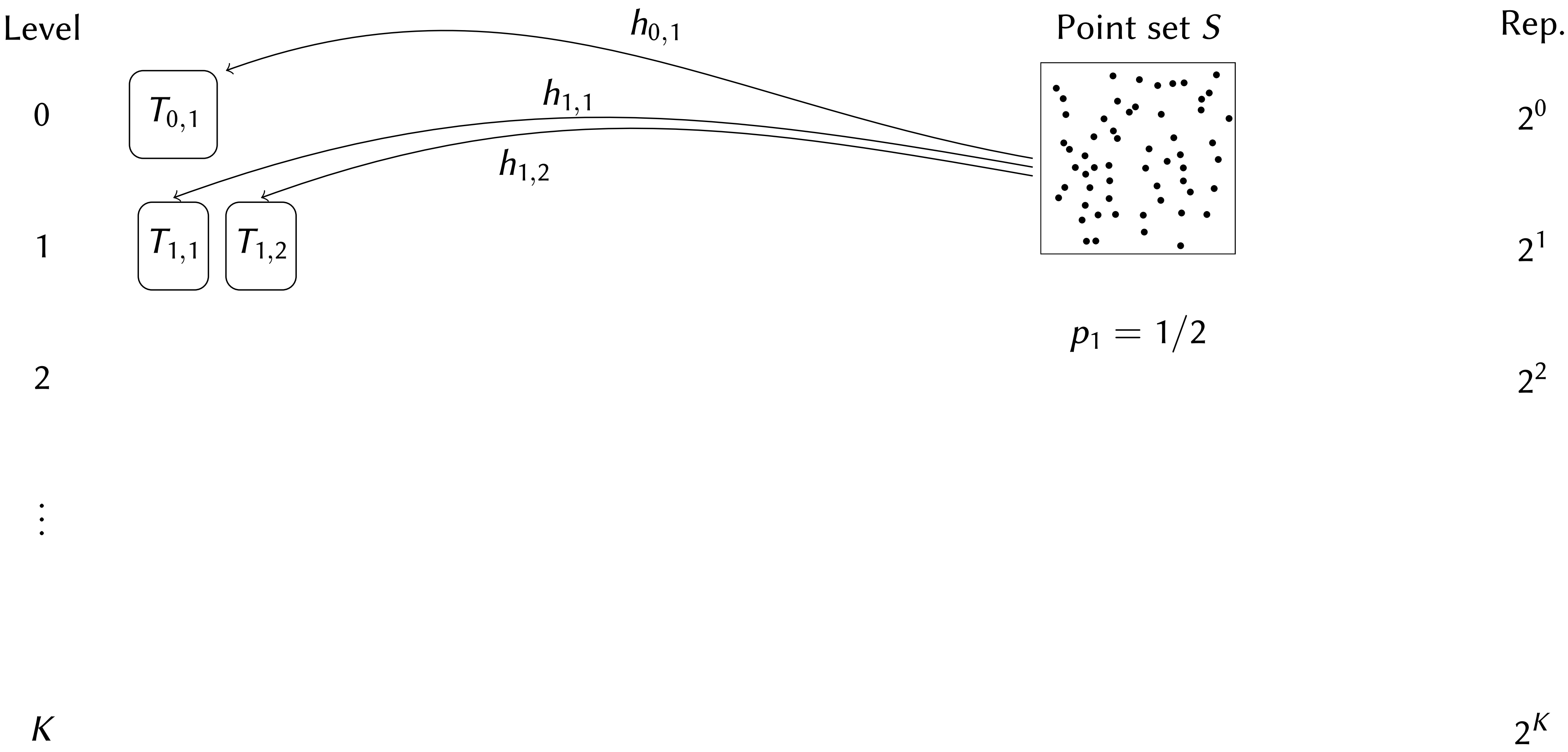
Adaptiveness: Multi-Level LSH + Adaptive Query

Level	Point set S	Rep.
0		2^0
1		2^1
2	$p_1 = 1/2$	2^2
$\vdots$		
K		2^K

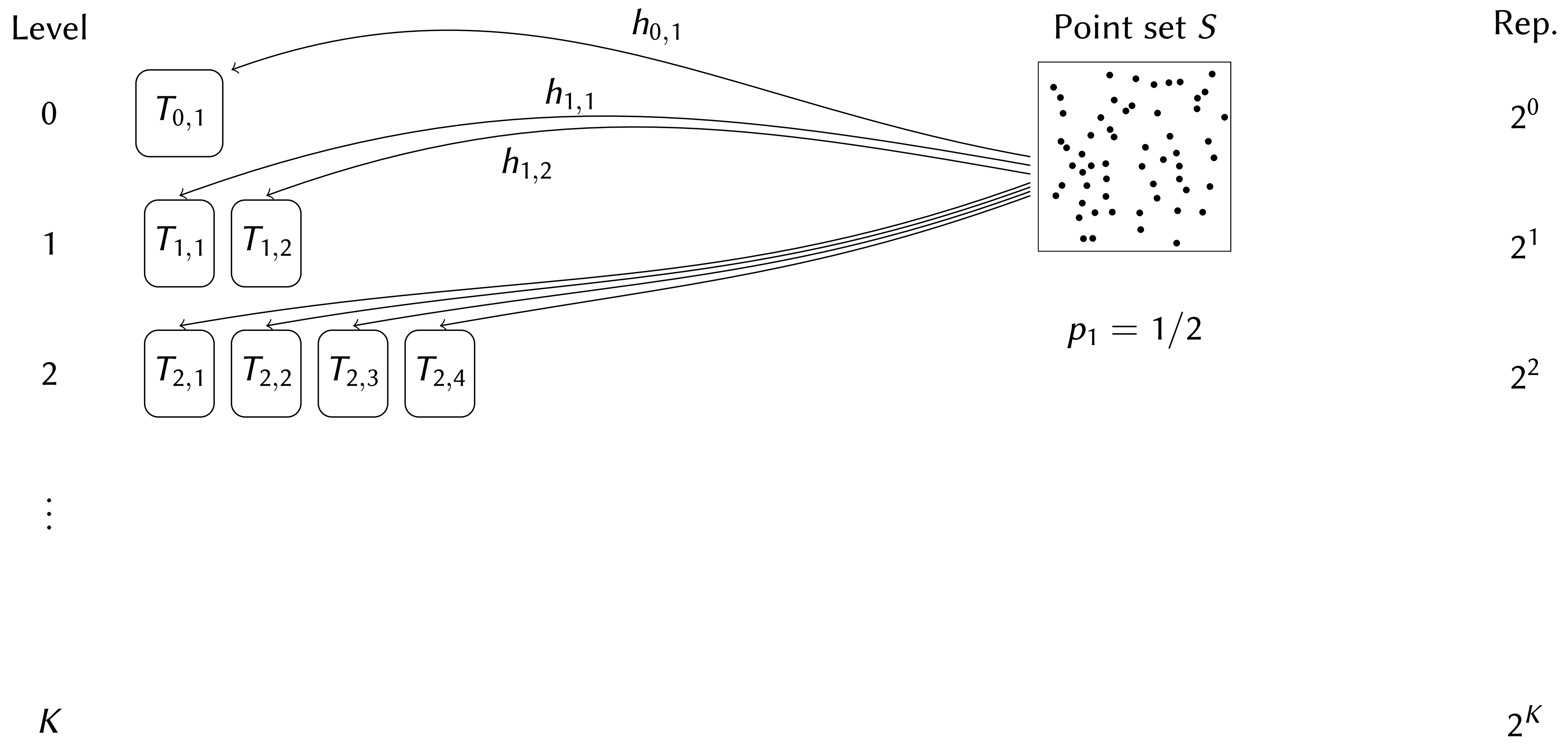
Adaptiveness: Multi-Level LSH + Adaptive Query



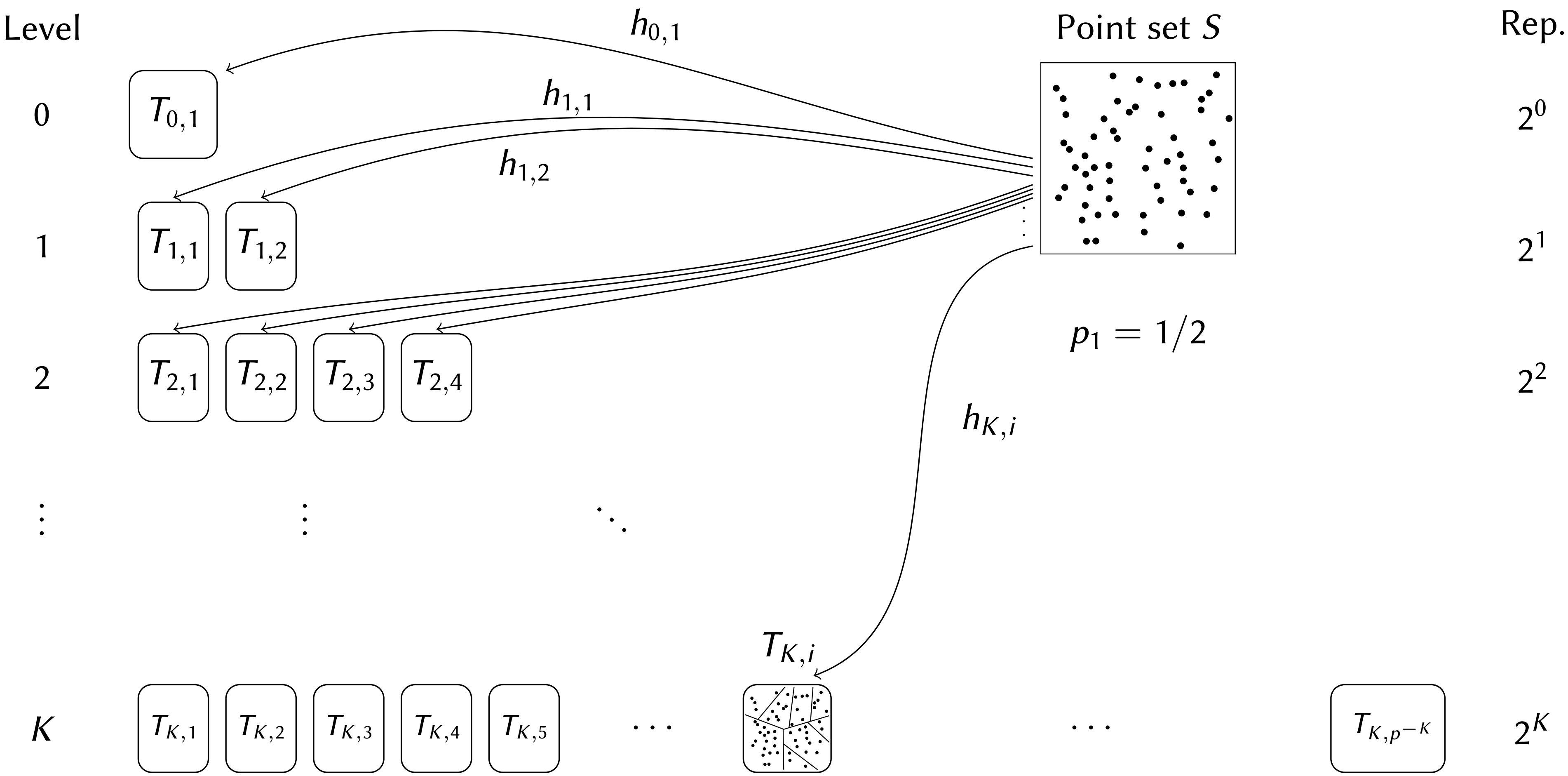
Adaptiveness: Multi-Level LSH + Adaptive Query



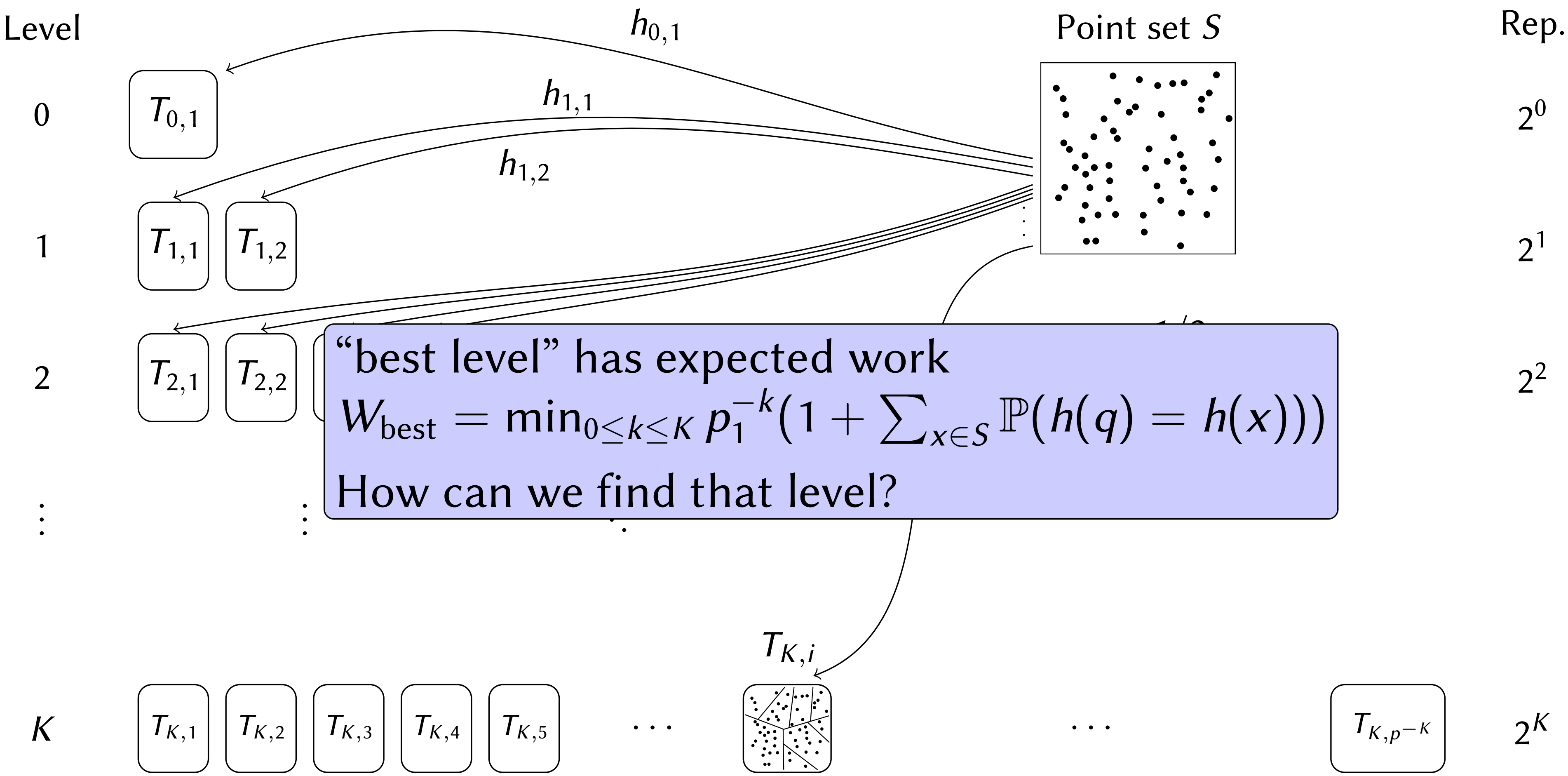
Adaptiveness: Multi-Level LSH + Adaptive Query



Adaptiveness: Multi-Level LSH + Adaptive Query



Adaptiveness: Multi-Level LSH + Adaptive Query

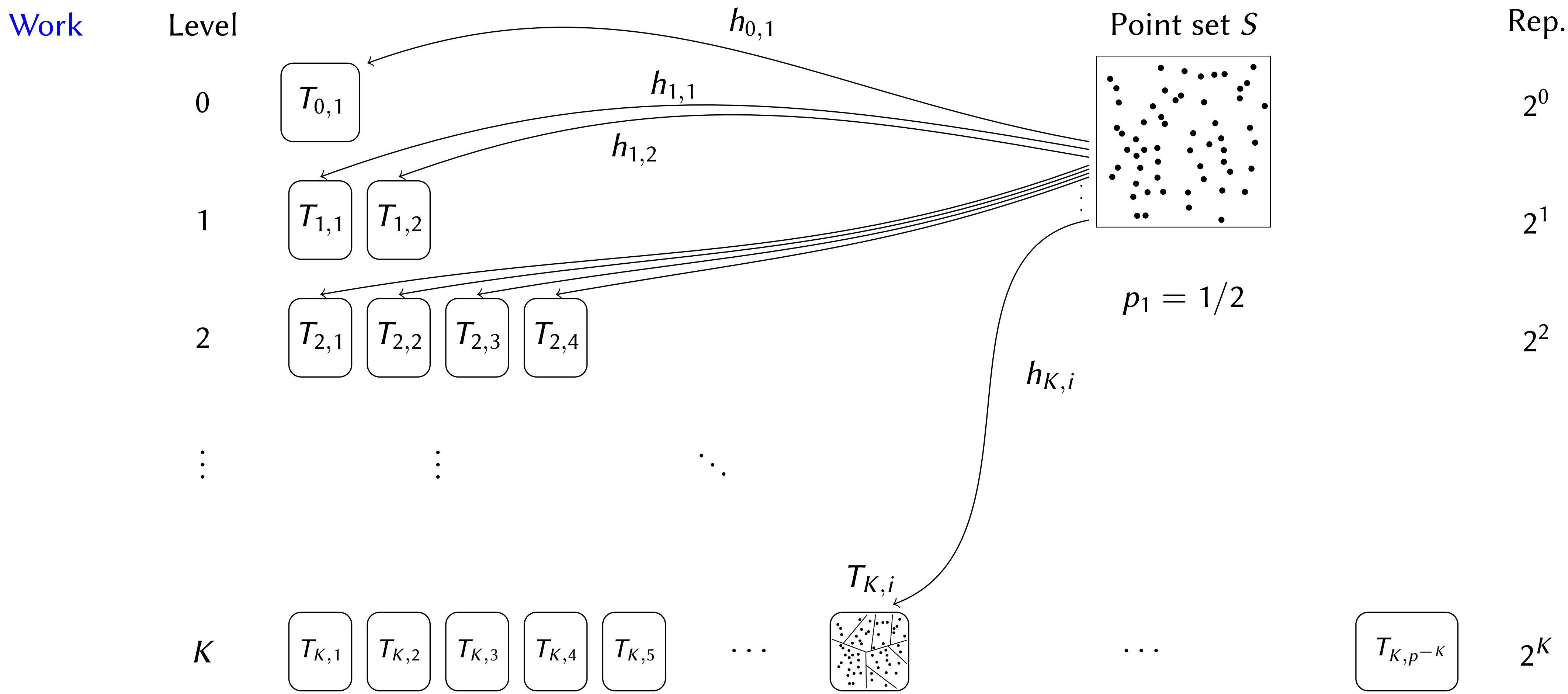


"best level" has expected work

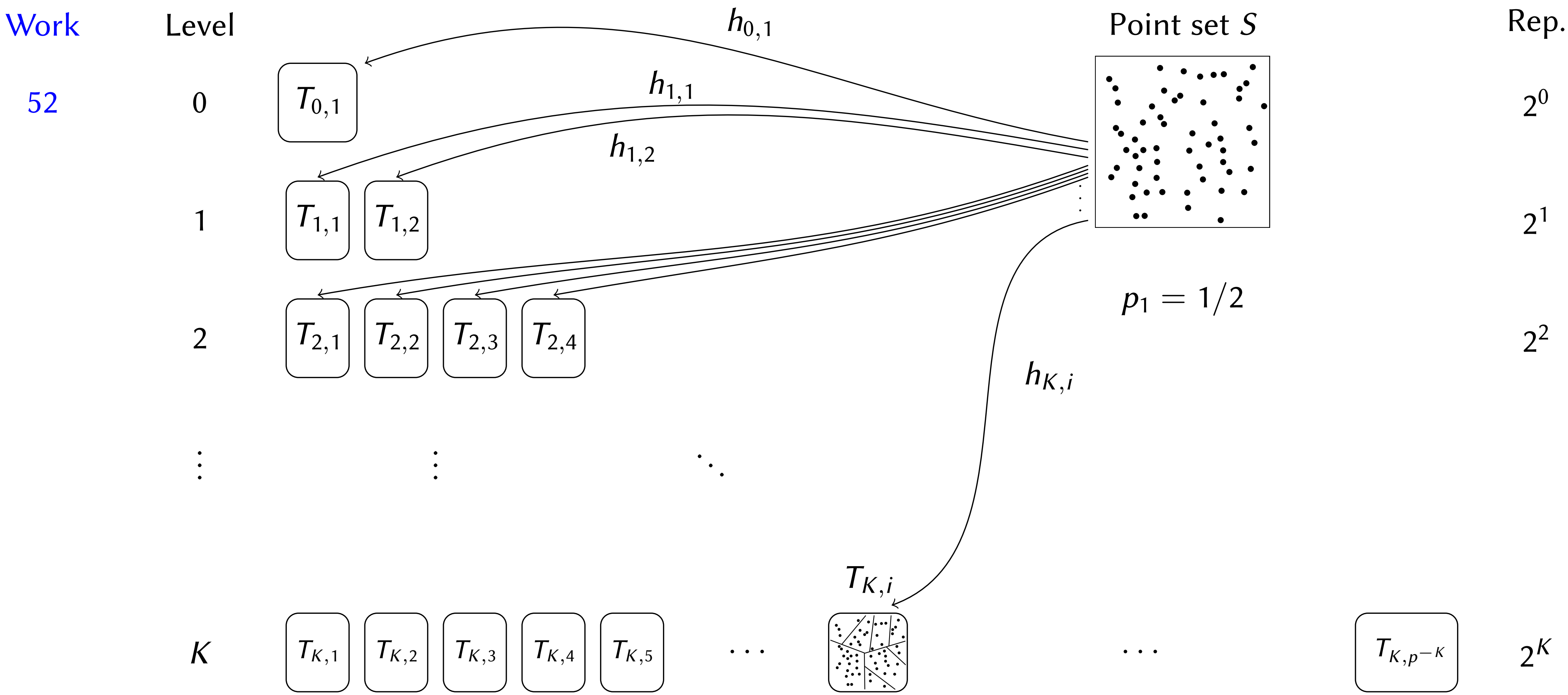
$W_{\text{best}} = \min_{0 \le k \le K} p_1^{-k}(1 + \sum_{x \in S} \mathbb{P}(h(q) = h(x)))$

How can we find that level?

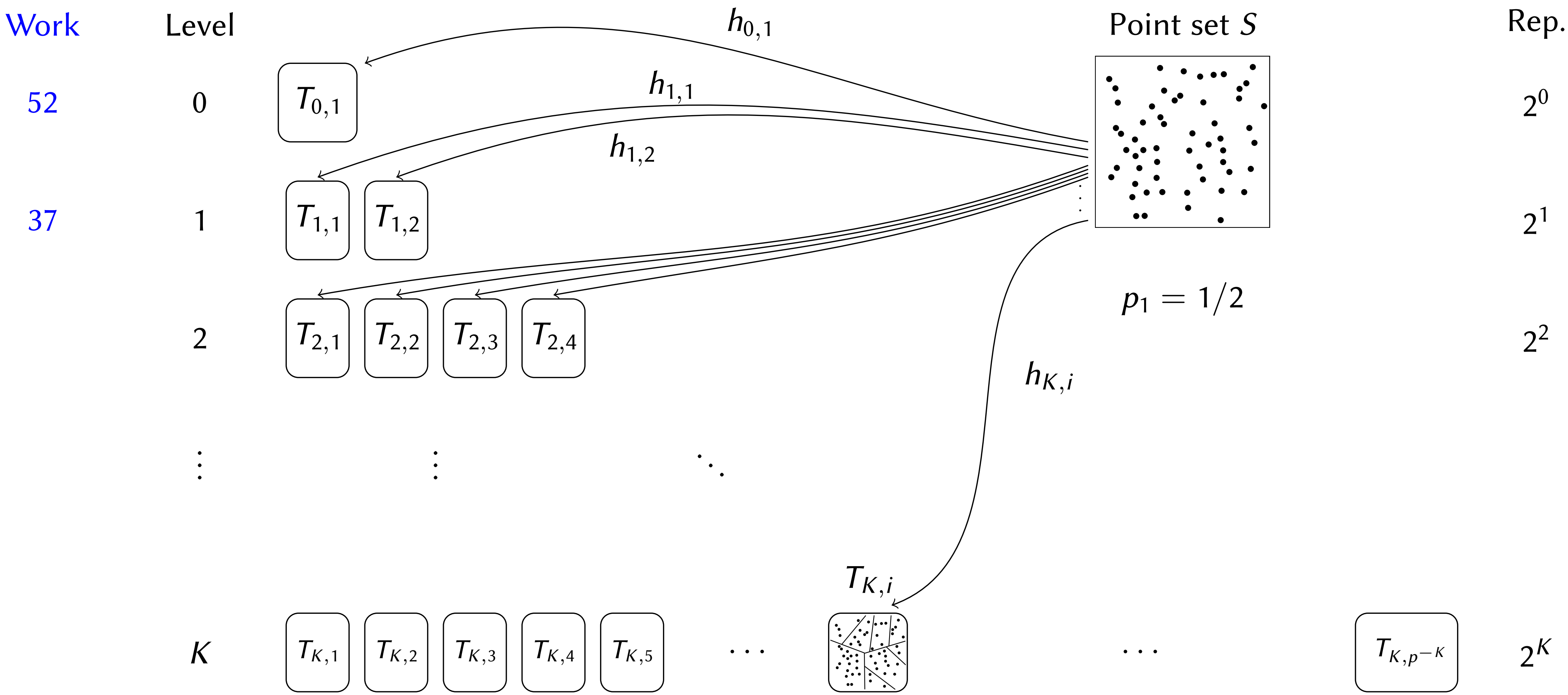
Adaptiveness: Multi-Level LSH + Adaptive Query



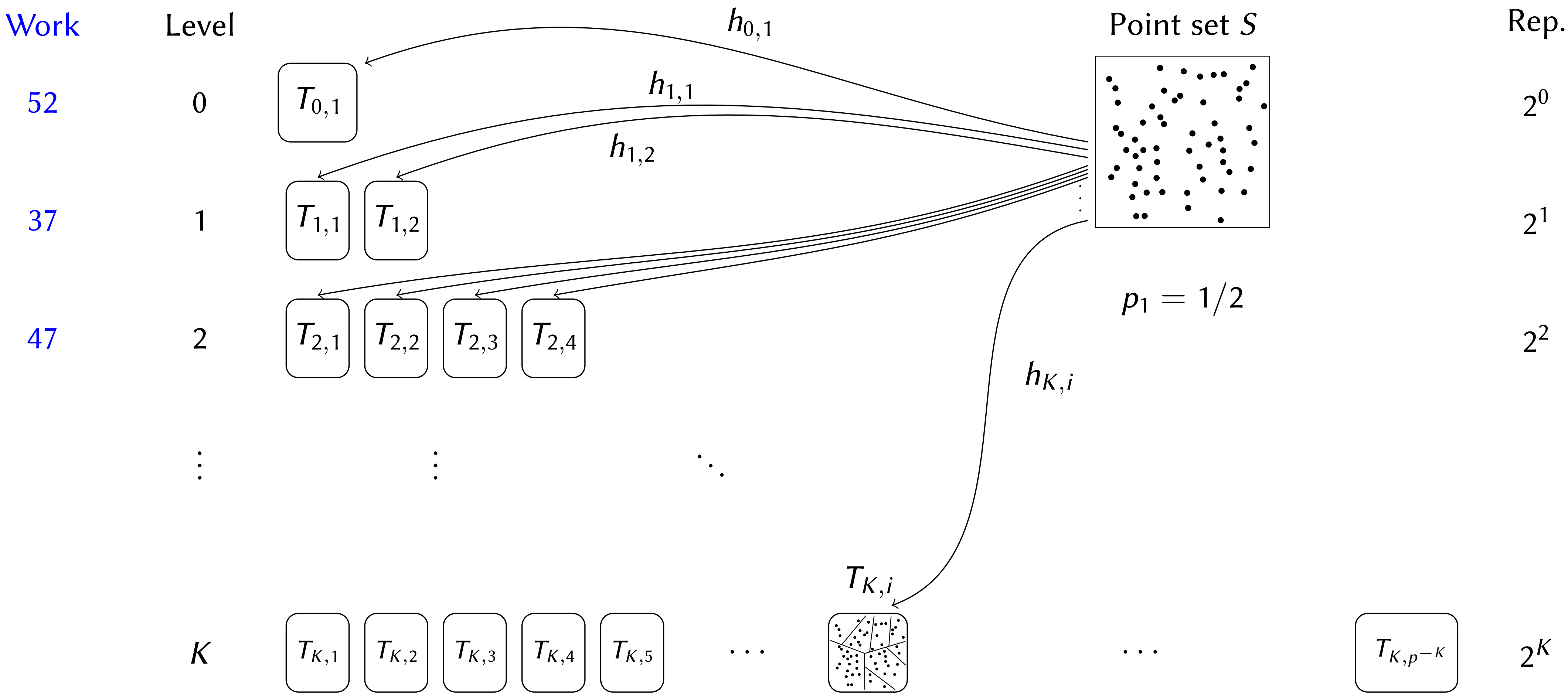
Adaptiveness: Multi-Level LSH + Adaptive Query



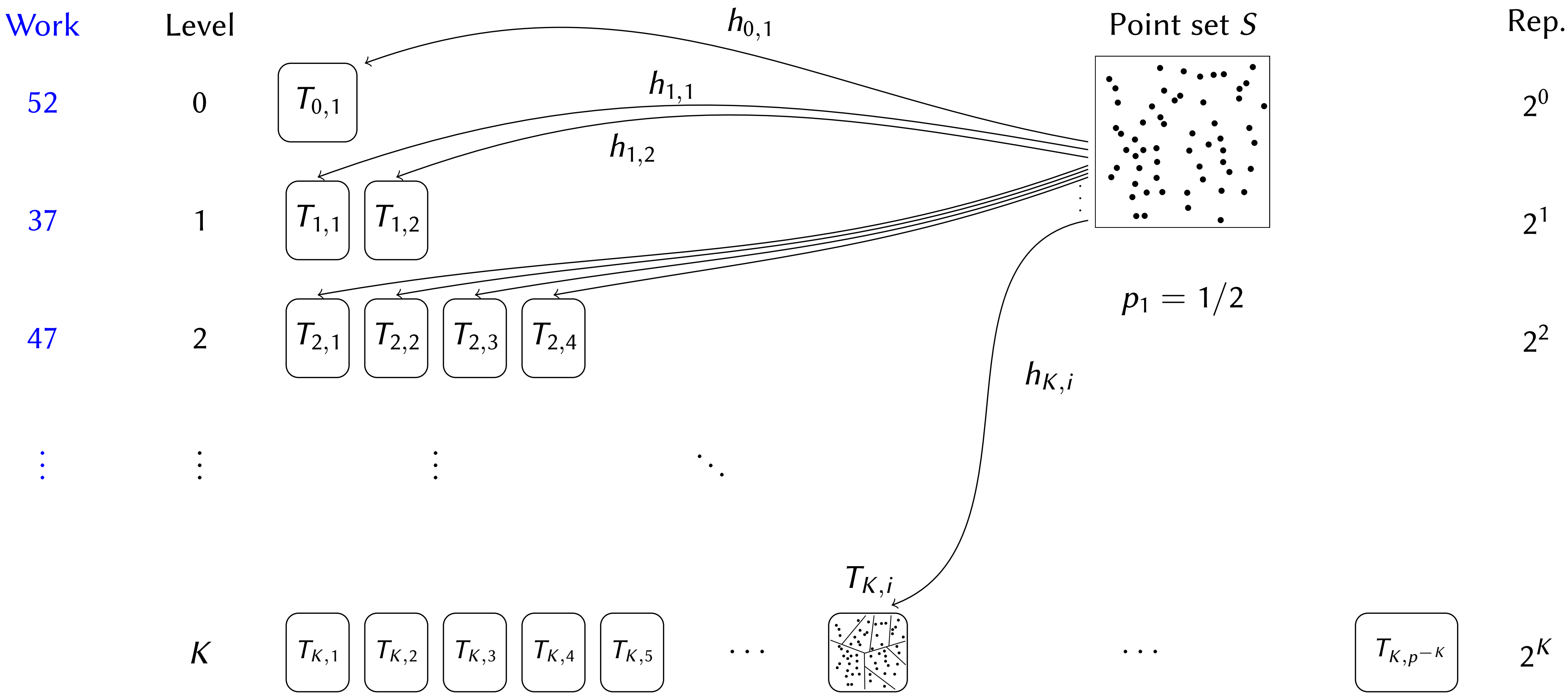
Adaptiveness: Multi-Level LSH + Adaptive Query



Adaptiveness: Multi-Level LSH + Adaptive Query



Adaptiveness: Multi-Level LSH + Adaptive Query



Running Time of Adaptive Algorithm

Theorem

Best level has expected work $W_{best} \Rightarrow$ algorithm has expected running time $O(W_{best} \log\log W_{best})$.

- algorithm is asymptotically only log log factor away from the **best possible work for the query**
- log log factor comes from needing $\approx \log k$ more repetitions on level k then standard LSH approach

Conclusion & Open Problems

- query algorithm on a multi-level LSH data structure that adapts to **best possible work**
- spherical range reporting in time $O(t(n/t)^{\rho^*})$, where ρ^* depends on expansion around the query
- in paper: multi-probing-aware variant of algorithm, slight improvement in running time

Open questions:

- spherical range reporting in time $O(n^\rho + t)$, adaptive to query?
- usefulness of algorithm for finding k-nearest neighbors?
- data-dependent methods?
- can we make use of space/time-tradeoff LSH data structures?

Conclusion & Open Problems

- query algorithm on a multi-level LSH data structure that adapts to **best possible work**
- spherical range reporting in time $O(t(n/t)^{\rho^*})$, where ρ^* depends on expansion around the query
- in paper: multi-probing-aware variant of algorithm, slight improvement in running time

Open questions:

- spherical range reporting in time $O(n^\rho + t)$, adaptive to query?
- usefulness of algorithm for finding k-nearest neighbors?
- data-dependent methods?
- can we make use of space/time-tradeoff LSH data structures?

Thank you!