

# Approximate Query Processing

1

- Space Partitioning-based
  - ▣ Tree
  - ▣ Encoding
  - ▣ Locality Sensitive Hashing
- Graph-based Methods

## Notes:

- Recent works mainly in the Database area
- Prefer ease of exposition over rigor
- Categorization is not fixed/unique

# Locality Sensitive Hashing (LSH)

2

## □ From the perspective of collision probability

### □ (Ordinary) hash function $h$ :

- $\Pr[ \underline{h(x) = h(y)} ] = \underline{\varepsilon}$ , if  $x \neq y$

### □ LSH

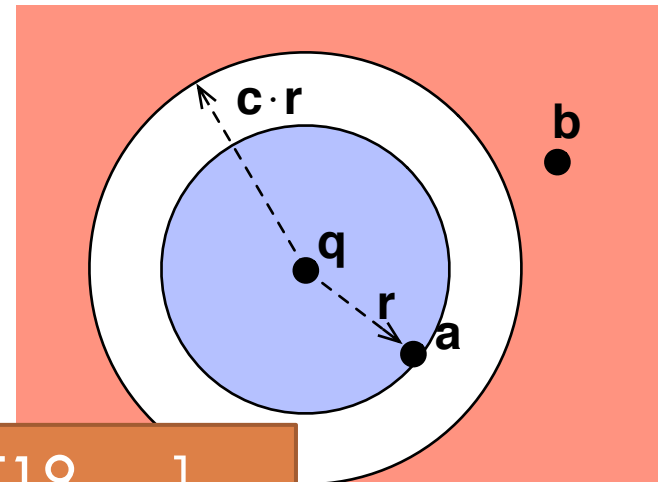
- $\Pr[ \underline{h(x) = h(y)} ]$  increases with **locality**
- Randomness comes from r.v.  $\underline{h} \in \underline{H}$

c.f., Cryptographic hash functions  
 $\Pr [ h(x) = h(y) ] = 2^{-m}$ , if  
 $\text{Hamming}(x, y) = 1$

$(r_1, r_2, p_1, p_2)$ -sensitive [IM98]

- $\Pr[ h(x) = h(y) ] \geq p_1$ , if  $\text{dist}(x, y) \leq r_1$
- $\Pr[ h(x) = h(y) ] \leq p_2$ , if  $\text{dist}(x, y) \geq r_2$

$\Pr[ h(x) = h(y) ] = \underline{\text{sim}}(x, y)$  [C02] **too narrow**



Can be generalized [SWQZ+14, ACPS18, CKPT19, ...]

# Locality Sensitive Hashing (LSH)

3

## □ Equality search

□ Index: store  $o$  into bucket  $h(o)$

$$\Pr[ h(q) = h(o) ] = 1/B$$

□ Query:

■ **retrieve** every  $o_i$  in the bucket  $h(q)$

■ **verify** if  $o_i = q$

## □ LSH

c.f., [PIM12] for the rigorous QP procedure

□  $\forall h \in \text{LSH-family}, \Pr[ Q(h(q)) = Q(h(o)) ] = f( \text{Dist}(q, o) )$

■  $Q(\ )$ : quantization (**not essential**)

■ “Near-by” points have more chance of colliding with  $q$  than “far-away” points

□ Similar index & query procedures, with a **weak** probabilistic guarantee

→ Repeat to boost the guarantee

# LSH Families

4

- Many are known
  - $L_p$  ( $0 < p \leq 2$ ): use  $p$ -stable distribution to generate the projection vector
    - For  $L_2$ , just use random Gaussian vector
    - Other families exist, e.g., sparse random projection
  - Angular distance (arccos): SimHash
  - Jaccard: minhash (based on random permutation)
  - Hamming:
    - random projection
    - covering LSH

# Comments

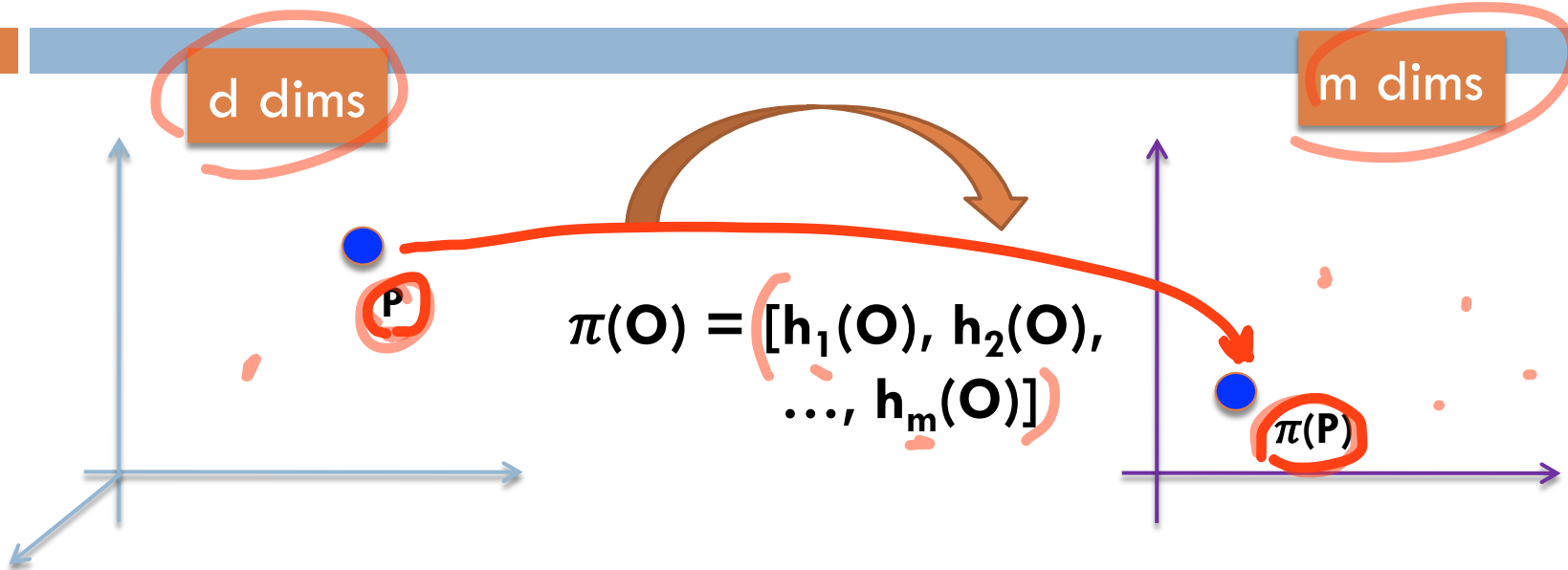
5

- New queries can be reduced to known LSH cases
  - Maximum inner product search (MIPS)
  - Set containment
  - Group aggregated query
- Related to various distortion-bounded embedding
  - Edit distance: CGK-embedding to Hamming with  $O(K)$  distortion

# Probabilistic Mapping

$d \gg m$

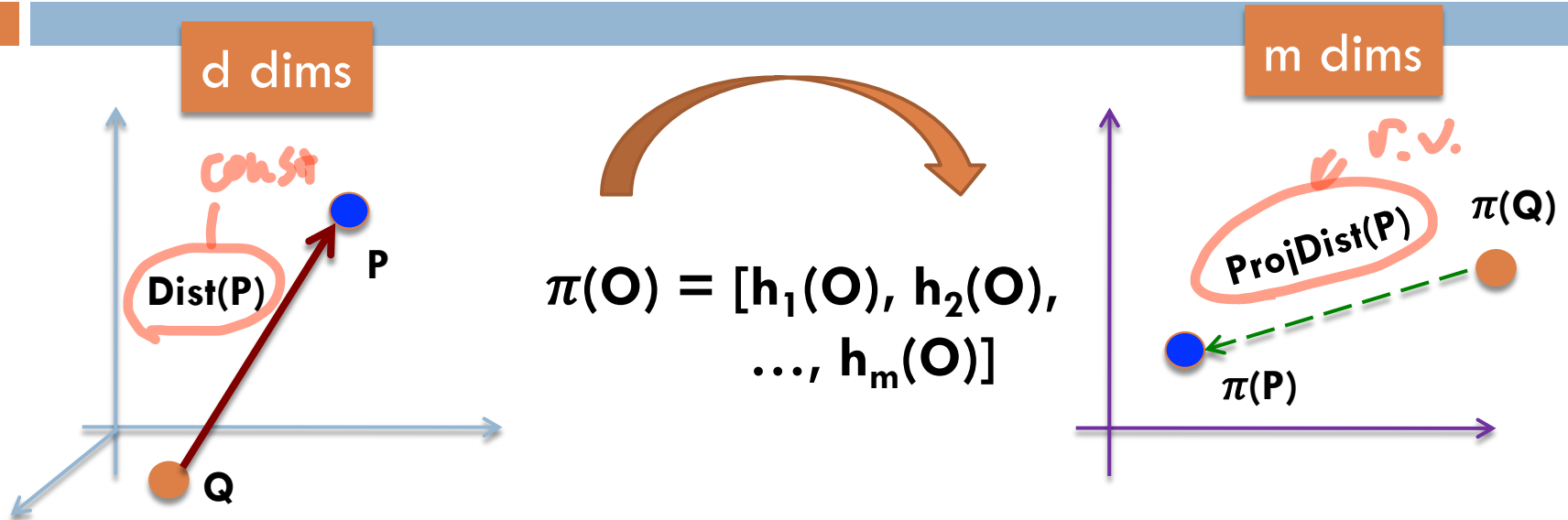
6



- Probabilistic, linear mapping from the original space to the projected space

# Probabilistic Mapping

7

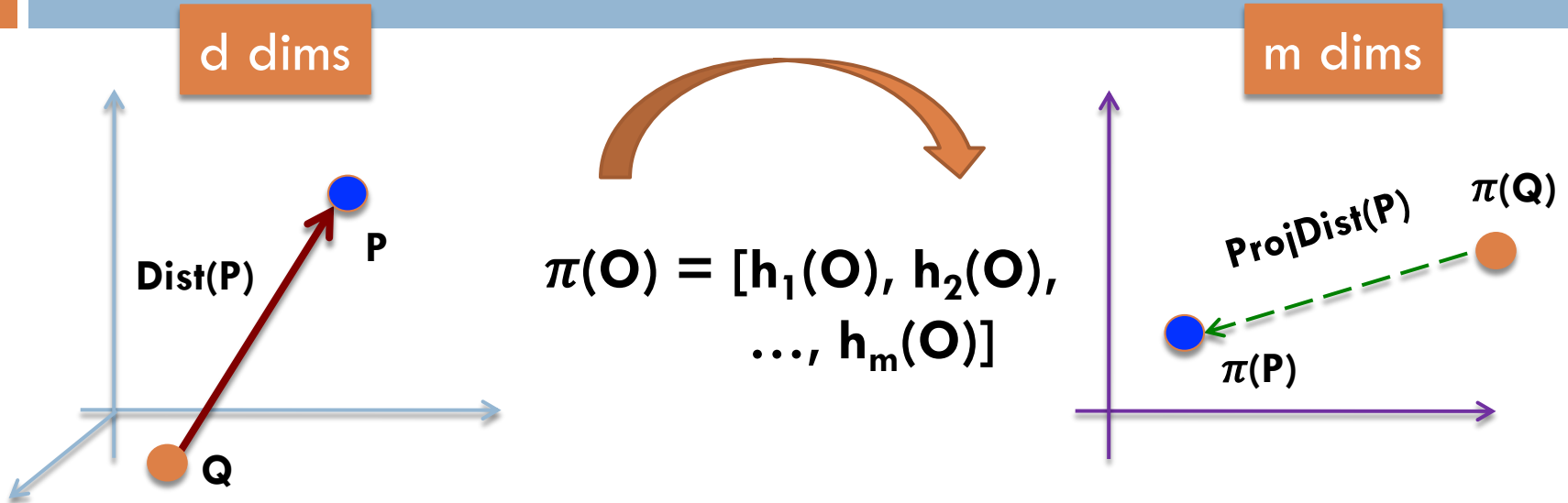


- Probabilistic, linear mapping from the **original space** to the **projected space**
- What about the **distances** (wrt  $Q$  or  $\pi(Q)$ ) in these two spaces?

# Probabilistic Distance Tracking

## Property of the Mapping

8



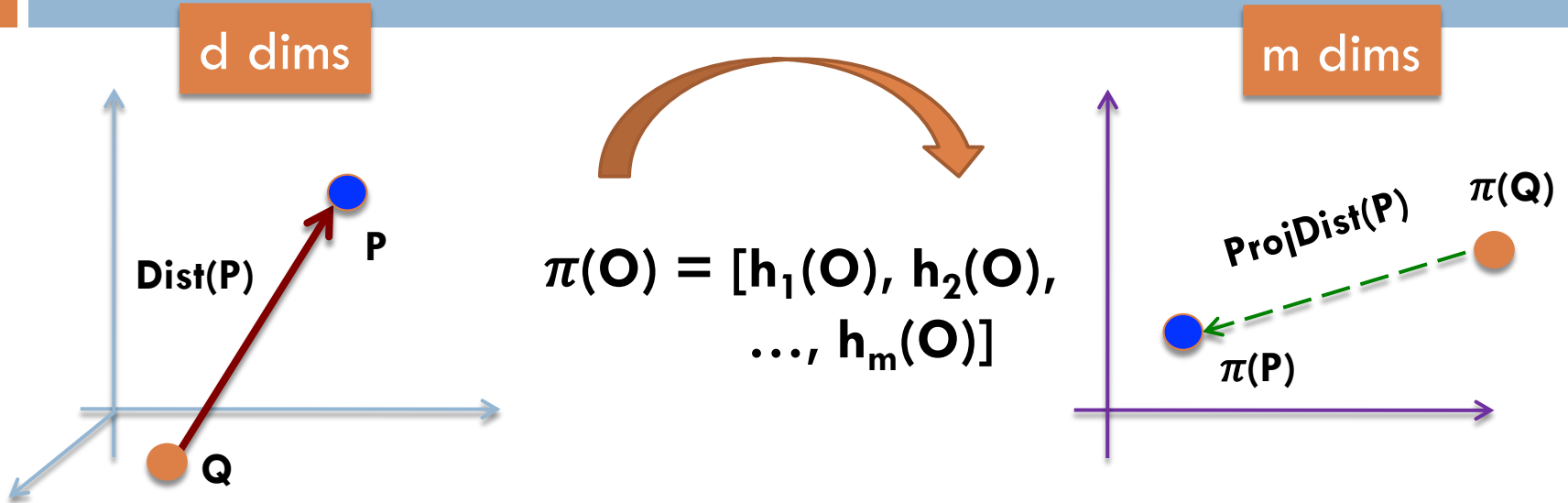
- $\text{ProjDist}(P)^2 \sim \text{Dist}(P)^2 * \chi_m^2$  [SWQZ+14]
  - $\text{ProjDist}(P)^2$  can be computed (incrementally) from  $h_i(P)$  and  $h_i(Q)$  due to the linearity of the hash function
  - Can be generalized to other p-stable LSH functions



# Probabilistic Distance Tracking

## Property of the Mapping

9



LSH provides a probabilistic distance-preserving mapping between the two spaces

Johnson & Lindenstrauss Lemma [JL84] only works for L2 and induces a method that requires more space than LSH [AIR18]

# Roadmap

10

## □ Roadmap

$O(n^{k+p})$

- **Practical** LSH methods (i.e., linear index complexity)
- Data-dependent LSH methods

# New Perspectives

11

- Inference Method
- Access method
- Stopping condition

# Inference

12

## □ Problem 1:

$$\text{ProjDist}(P)^2 \sim \text{Dist}(P)^2 * \chi^2_m$$

- Given that  $\text{ProjDist}(P) \leq r$ , what can we infer about  $\text{Dist}(P)$ ?

## □ SRS:

Similar to the usual  $(r_1, r_2, p_1, p_2)$  definition of LSH

- If  $\text{Dist}(P) \leq R$ , then  $\Pr[\text{ProjDist}(P) \leq r] \geq \Psi_m((r/R)^2)$
- If  $\text{Dist}(P) > cR$ , then  $\Pr[\text{ProjDist}(P) \leq r] \leq \Psi_m((r/cR)^2) = t$
- (some probability) at most  $O(tn)$  points with  $\text{ProjDist} \leq R$
- (constant probability) one of the  $O(tn)$  points has  $\text{Dist} \leq R$

$E_1$

$E_2$

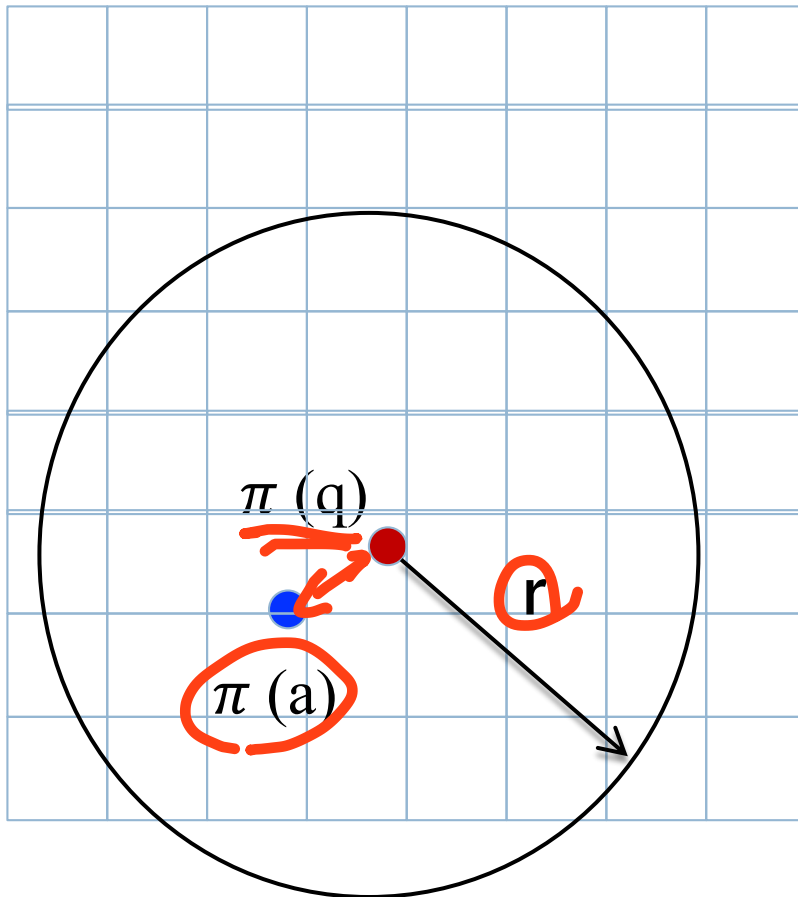
- This solves the so-called  $(R, c)$ -NN queries  $\rightarrow$  returns a  $c^2$  ANN
- Using another algorithm & proof  $\rightarrow$  returns a  $c$ -ANN

$\rightarrow$  Inference requires precise & complete information of the projections

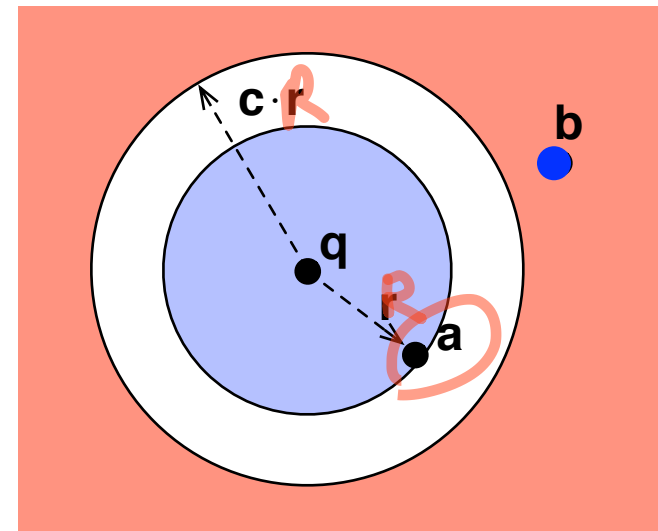
# Near Points

13

Likely ( $\geq p_1$ )



$\text{Dist}(a) \leq R$

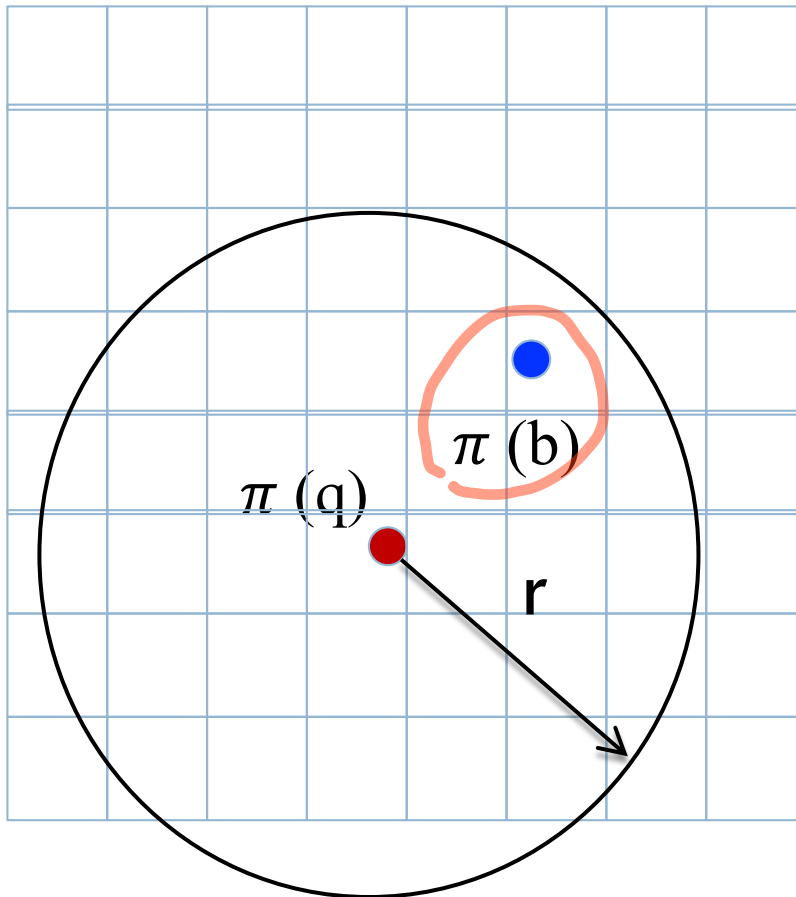


$d$ -dimensional space

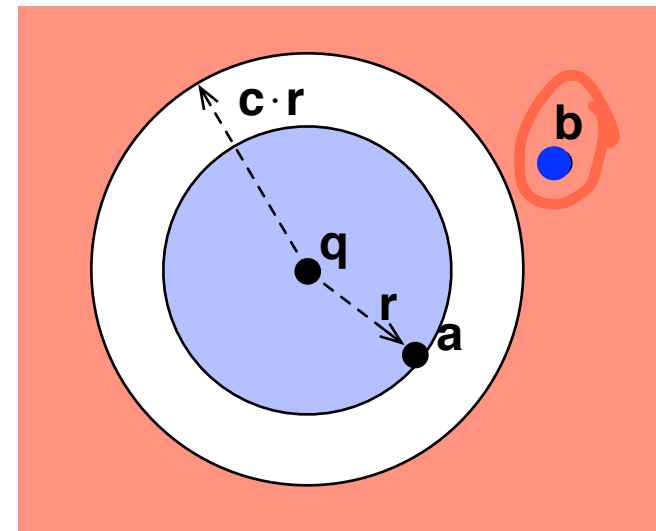
# Faraway Points

14

Unlikely ( $\leq p_2$ )



Dist(b)  $\geq cR$

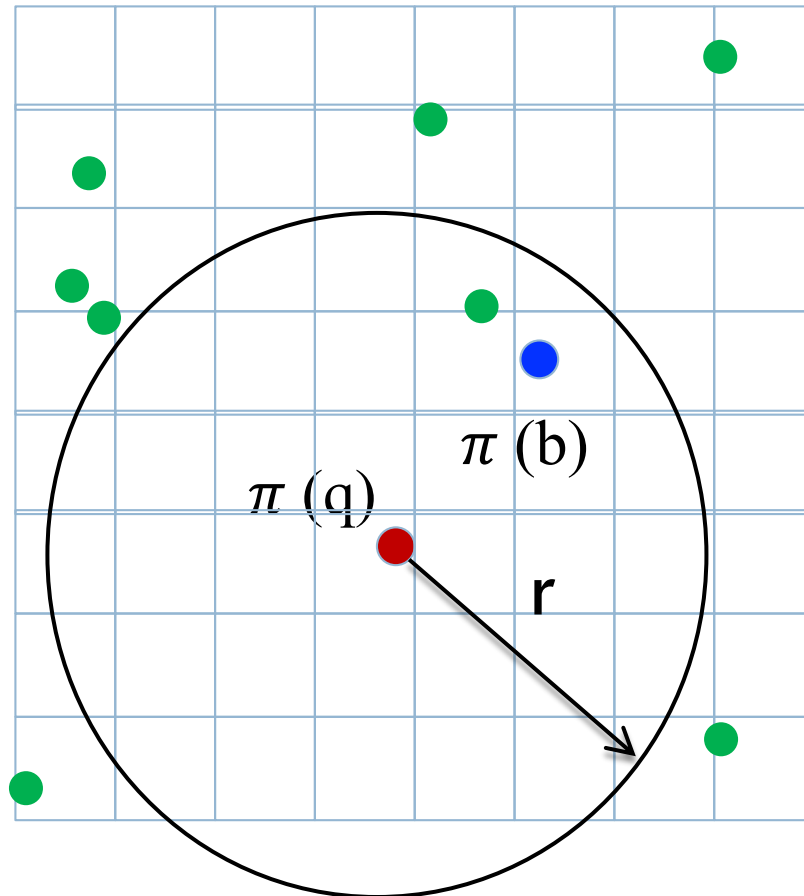


**d-dimensional space**

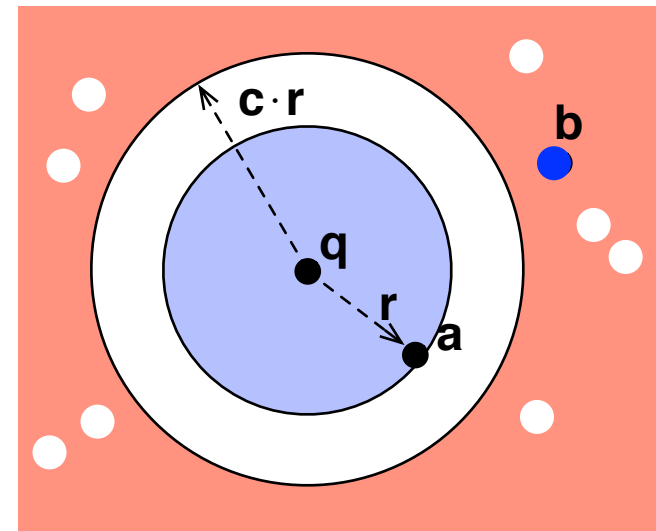
# Consider all faraway Points

15

Expected to see few of them



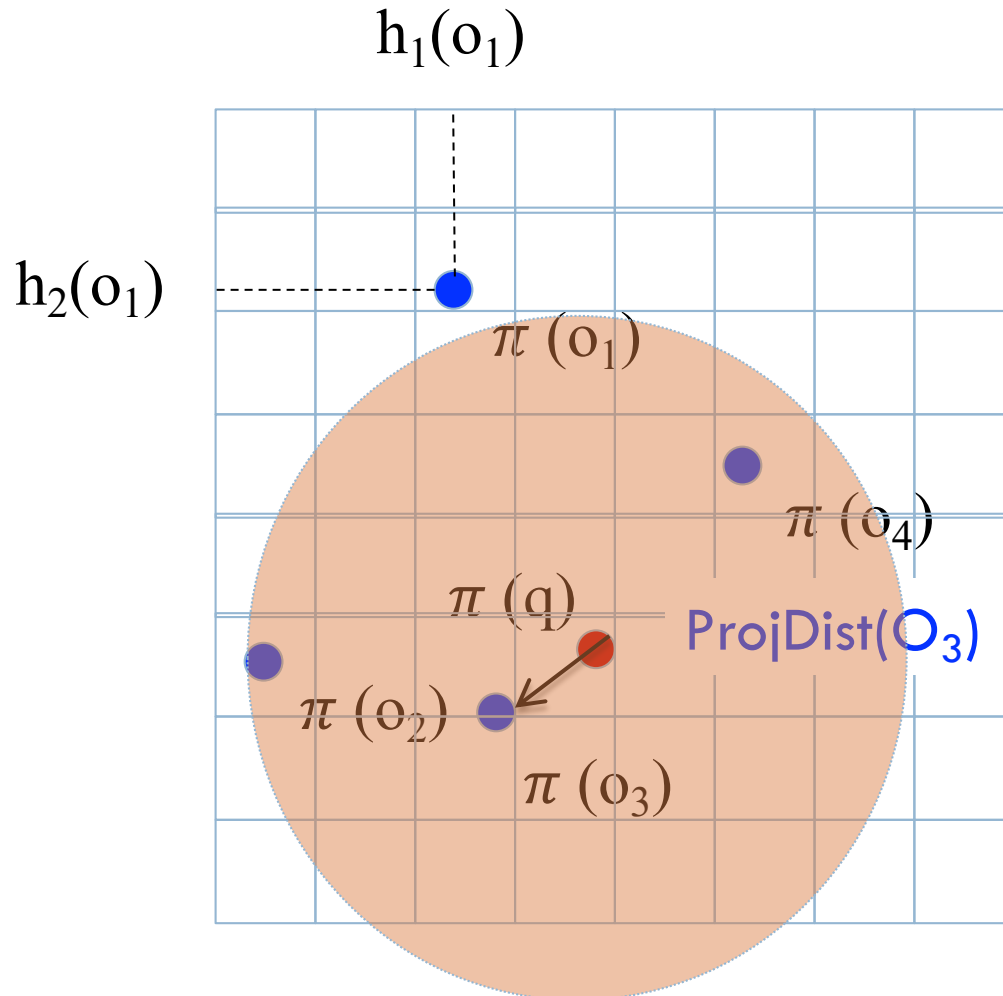
$\text{Dist}(b) \geq cR$



**d-dimensional space**

# Exact $t^*n$ -NN Query in $m$ -dim Space

16



if  $t_n = 2$ , then one of the **top-3 NNs** in the **projected space** around  $\pi(Q)$  is a c-ANN with constant probability

**ProjDist( $o_3$ )** is the **minimum among the 4 points**



# Inference

17

## □ Problem 2:

- Given that  $z(\pi(P))$  is similar to  $z(\pi(Q))$ , what can we infer about  $\text{Dist}(P)$ ?

Measured by  $\text{LLCP}(z(\pi(P)), z(\pi(Q)))$

## □ LSB:

- If  $\text{Dist}(P) \leq R$ , then  $\Pr[\text{LLCP}(P, Q) \geq \delta] \geq p_1^m$
- If  $\text{Dist}(P) > 2R$ , then  $\Pr[\text{LLCP}(P, Q) \geq \delta] \leq p_2^m$
- (some probability) at most  $O(p_2^m n)$  points with  $\text{ProjDist} \leq R$
- (constant probability) one of the  $O(p_2^m n)$  points has  $\text{Dist} \leq R$

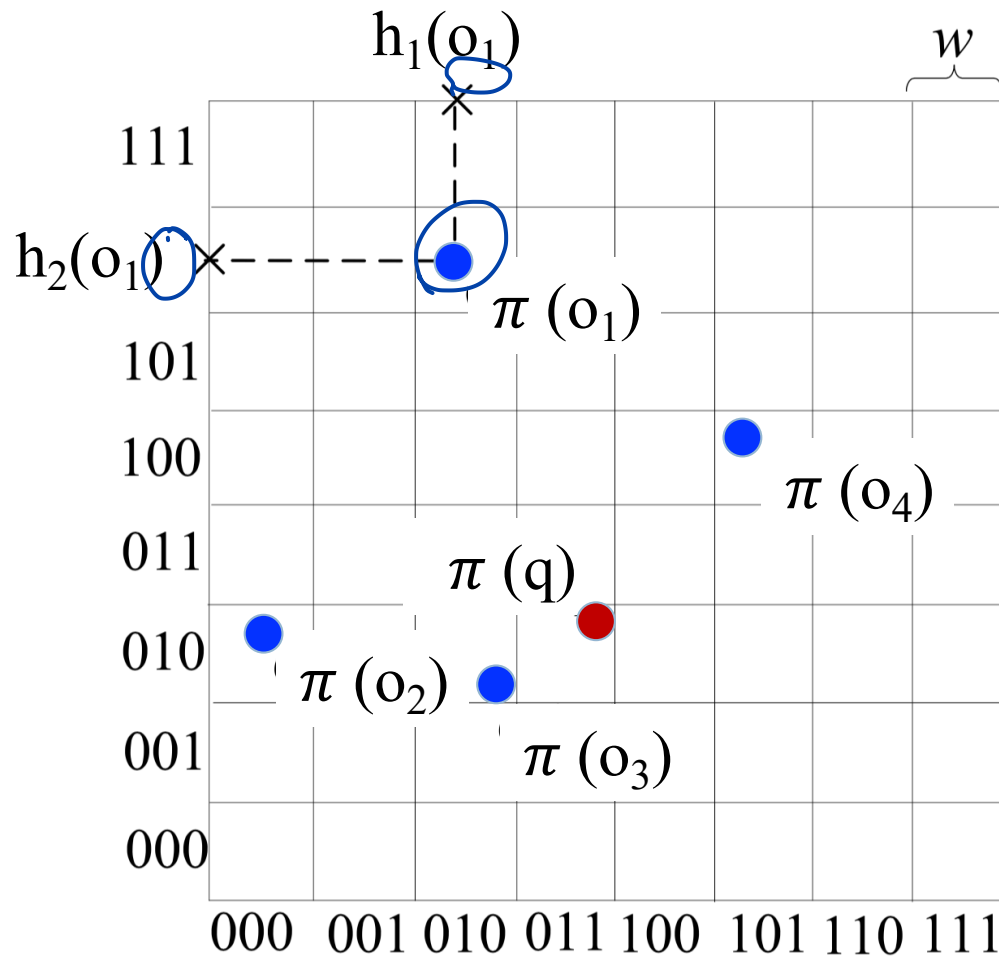
$E_1$

$E_2$

Inference requires precise & complete information of about  $z()$  values

# z-order

18



$$z(\pi(o_1)) = 011100$$

$$z(\pi(Q)) = 001110$$

$$z(\pi(o_3)) = 001100$$

$$\text{LLCP}(o_1, Q) = 1$$

$$\text{LLCP}(o_3, Q) = 4$$

# Inference

19

## □ Problem 3:

Collision wrt  $w$ : if  $|h_i(P) - h_i(Q)| \leq w$

- Given that P's #collision  $\geq \alpha m$ , what can we infer about  $\text{Dist}(P)$ ?

## □ C2LSH/QALSH:

- If  $\text{Dist}(P) \leq R$ , then  $\text{Pr}[\text{\#collision} \geq \alpha m] \geq \gamma_1$
- If  $\text{Dist}(P) > cR$ , then  $\text{Pr}[\text{\#collision} \geq \alpha m] \leq \gamma_2$
- (some probability) at most  $O(\gamma_2^* n)$  points with  $\text{\#collision} \geq \alpha m$
- (constant probability) one of the  $O(\gamma_2^* n)$  points has  $\text{\#collision} \geq \alpha m$

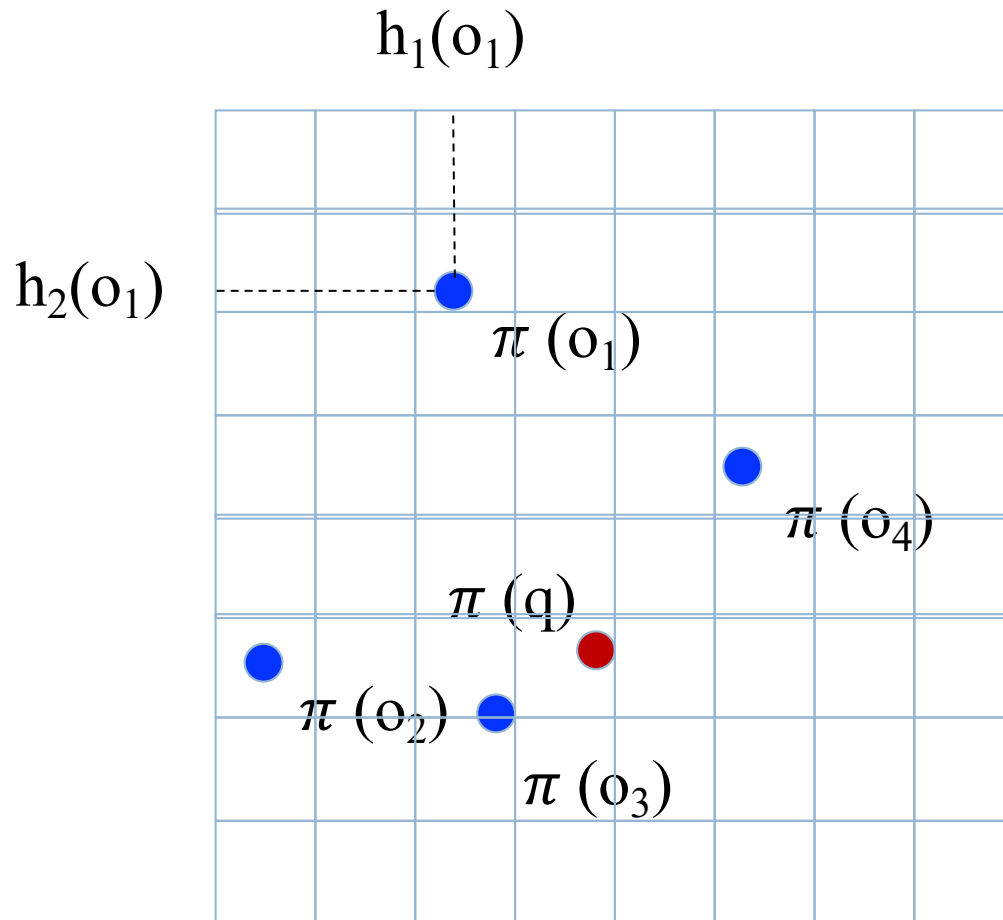
$E_1$

$E_2$

Inference requires rough & incomplete information of the projections

# Collision count

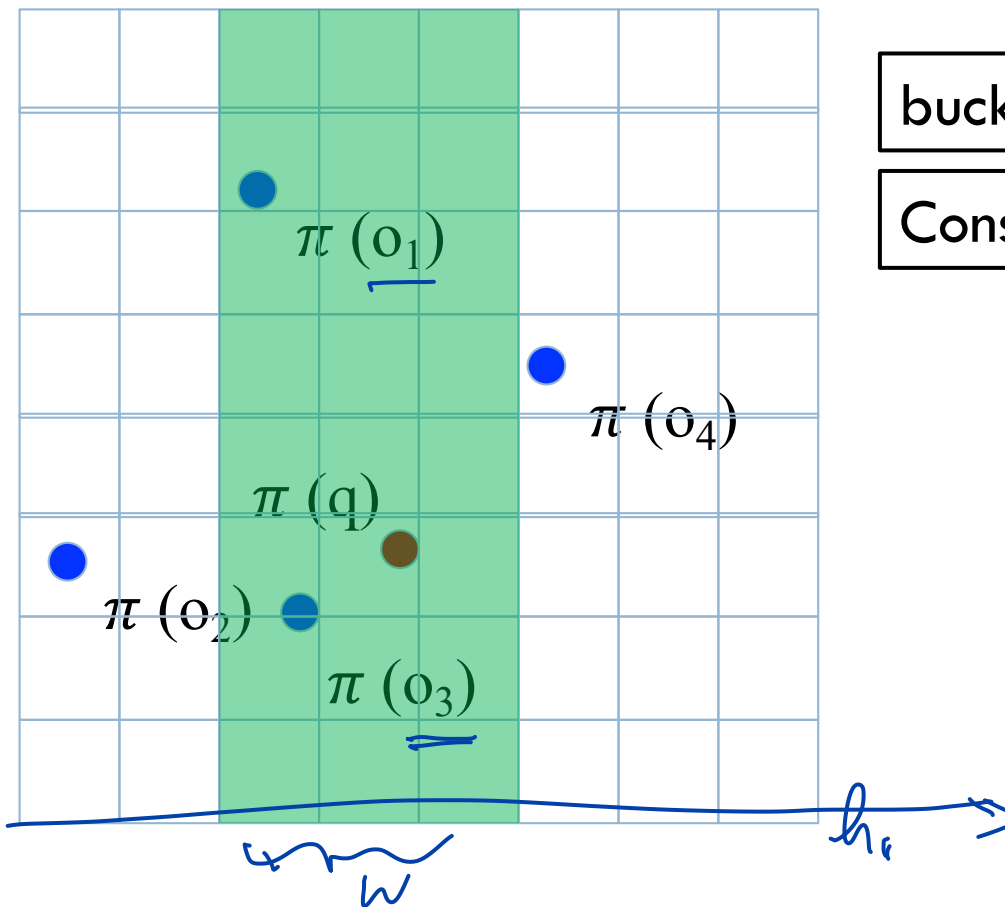
20



bucket width = 4

# Collision count

21

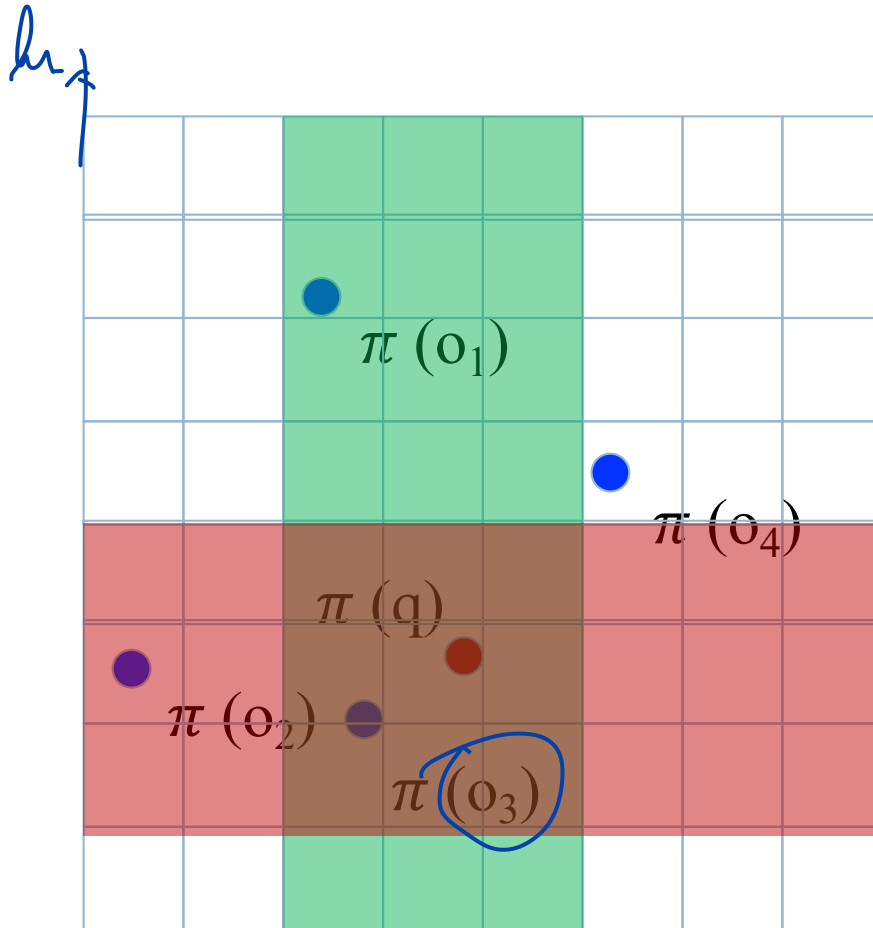


bucket width = 3

Consider  $h_1()$  axis

# Collision count

22



bucket width = 3

Consider  $h_1()$  axis

Consider  $h_2()$  axis

#Collision( $o_1$ ) = 1

#Collision( $o_2$ ) = 1

#Collision( $o_3$ ) = 2

#Collision( $o_4$ ) = 0

# Some Variants

23

- Looseness in C2LSH
  - $\gamma_1$  and  $\gamma_2$  computed using tail bounds
  - Constant probability obtained using the union bound
  - Contrast between  $\gamma_1$  and  $\gamma_2$  is low by using  $QZ(h_i(P)) = QZ(h_i(Q))$  as the collision
- QALSH
  - Use the right collision definition (virtual bucketing)
- PDA-LSH [YDSS20]
  - Computes  $\gamma_1$  and  $\gamma_2$  using Gaussian as an approximation
  - Approximately compute  $\Pr[E_1 \wedge E_2]$

# Inference

24

## □ Problem 4:

Collision wrt  $w$ : if  $|h_i(P) - h_i(Q)| \leq w$

- E
- Given that P's #collision  $\geq \alpha m$ , what can we infer about  $x$   
 $\triangleq \text{Dist}(P)$  ?

Requires assumption or tolerance of a prior

## □ Bayesian LSH:

- $\text{Pr}[x | E] = \text{Pr}[E | x] * \text{Pr}[x] / \text{Pr}[E]$

Posterior distribution

- Then, one can calculate many things

- $\text{Pr}[x \geq R | E]$
- MAP estimate  $x^\wedge = \text{argmax}_x \text{Pr}[x | E]$
- Bayesian Tail probabilities:  $\text{Pr}[|x^* - x^\wedge| > \epsilon | E]$

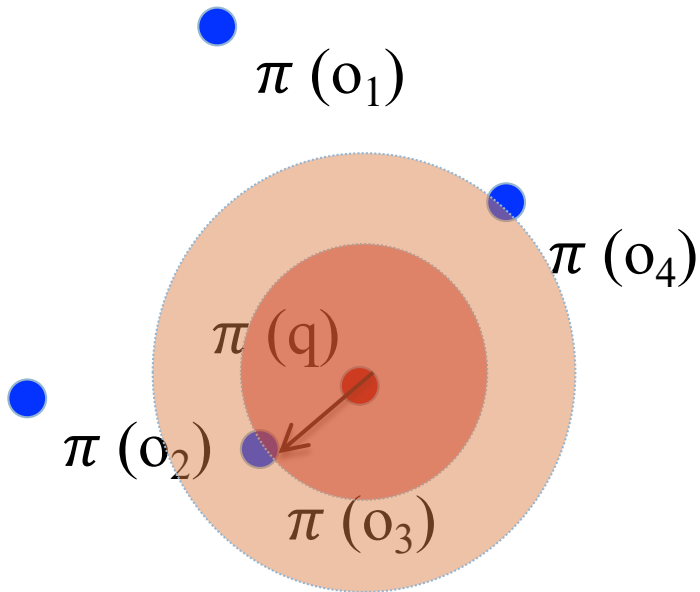
Inference requires rough & incomplete information of the projections



# Access Method

25

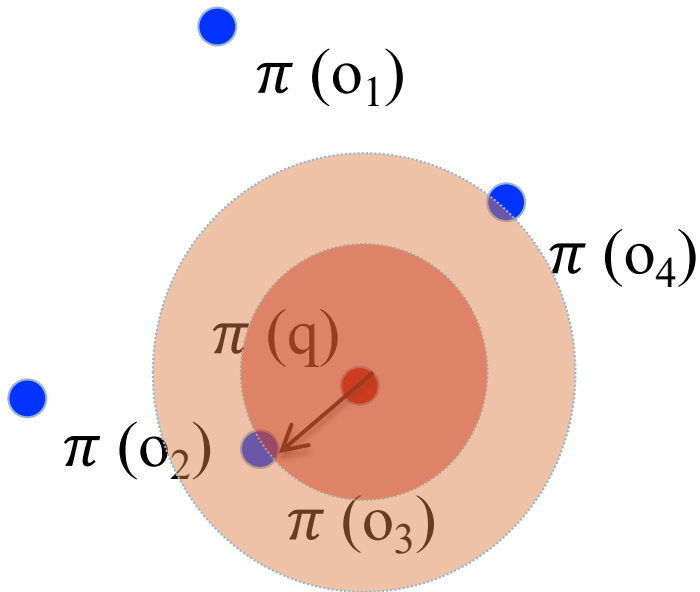
- Inference method → Access method
- SRS requires accessing projected points according to increasing ProjDist → R-tree (on disk) or Cover Tree (in memory)
  - m cannot be too large (e.g., m in [6, 8]) for R-tree



# Access Method

26

- Inference method → Access method
- Replace ~~R-tree~~ in SRS by a variant of the M-tree →  
PM-LSH
  - Allow m to be reasonably large (e.g., 15)
  - Uses distortion-based inference (partially)



# Access Method

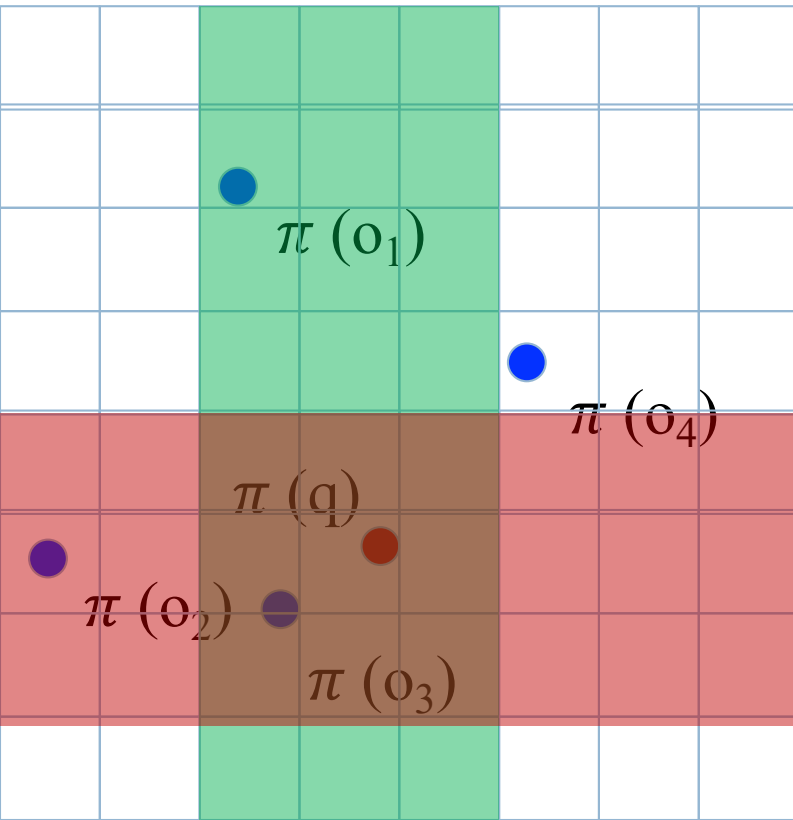
27

- Inference method ← Access method
- Use a new definition of bucket and collision →  
R2LSH
  - Bucket in 2d subspace  $\triangleq$  Ball of radius  $w$  centered at  $\pi_i(Q)$ ,  $i = 1, \dots, m/2$
  - Use the SRS-2 style stopping condition
  - Use a new index based on polar coordinates

# Access Method

28

- Inference method ← Access method
- C2LSH requires accessing each projection with a bucket width constraint → B-tree



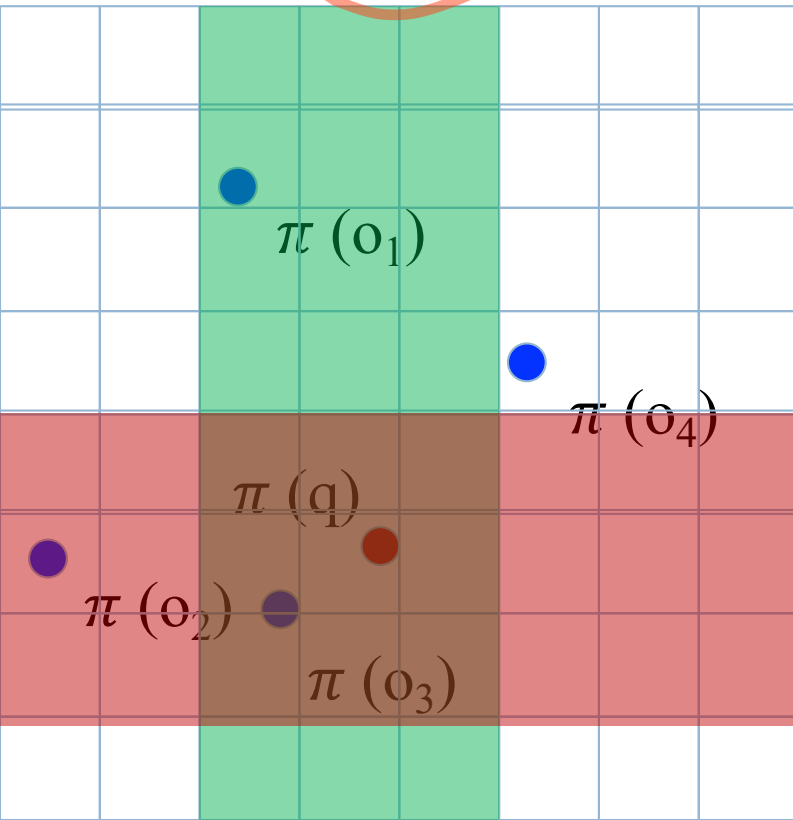
bucket width = 3  
#Collision threshold = 2  
Only  $o_3$  is a candidate

However, we do have partial information about  $o_1$  and  $o_2$

# Access Method

29

- Inference method ← Access method
- Make use of almost ALL accessed points in QALSH  
→ VHP

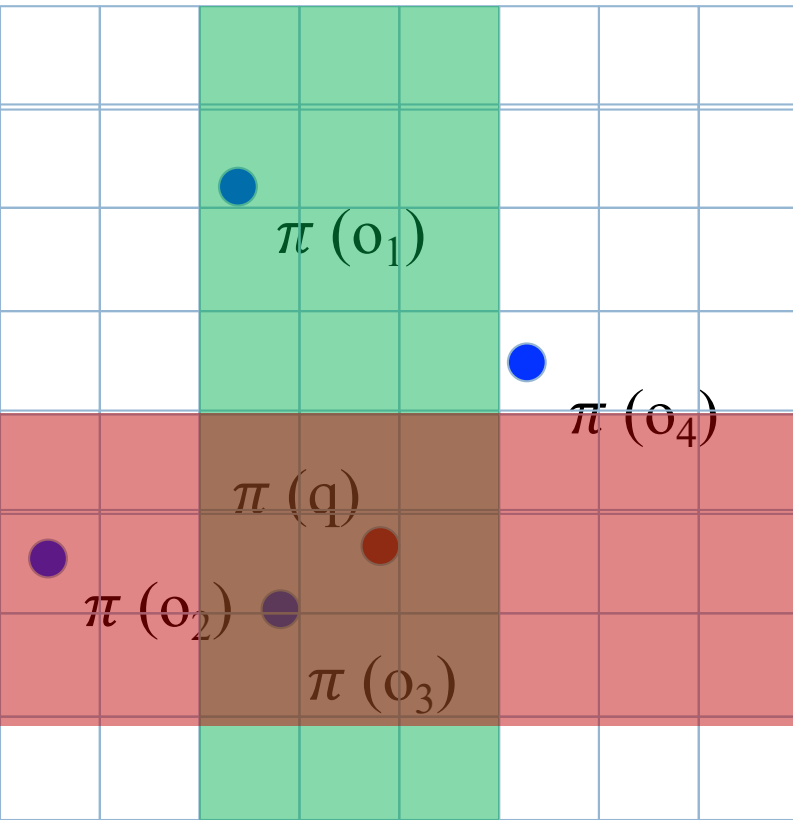


Object	#Collisions	Partial ProjDist	
$o_1$	1	3.2	
$o_2$	1	3.5	
$o_3$	2	1.3	

# Access Method

30

- Inference method ← Access method
- Make use of almost ALL accessed points in QALSH  
→ VHP



Object	#Collisions	Partial ProjDist	PPDist Threshold
<u>o<sub>1</sub></u>	<u>1</u>	<u>3.2</u>	<u>3.4</u>
o <sub>2</sub>	1	3.5	3.4
<u>o<sub>3</sub></u>	<u>2</u>	<u>1.3</u>	<u>1.9</u>

Assuming the Partial ProjDist thresholds for different #Collisions, both o<sub>1</sub> and o<sub>3</sub> are candidates

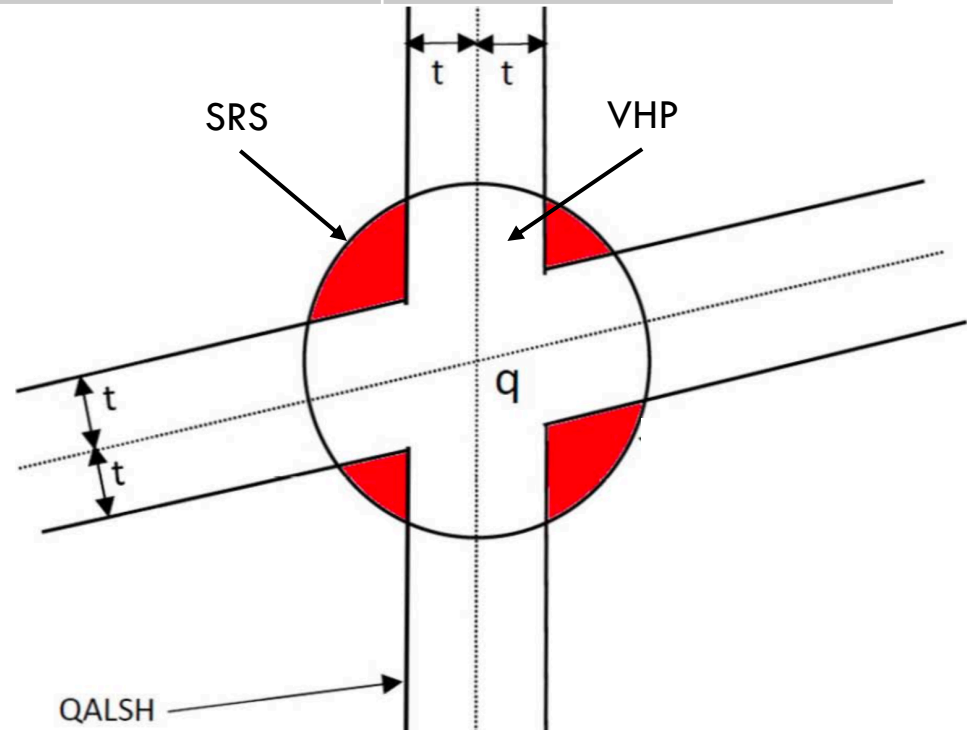
# Some Comparisons

- Candidate Conditions

Method	Collision Count	(Observed) Distance	Max Candidates
SRS	$= m$	$\leq r$	$T$
QALSH	$\geq \alpha m$	$n/a$	$\beta n$
VHP	$\geq i$ ( $i = 1, 2, \dots, m$ )	$\leq l_i$	$\beta n$

- Candidate Regions

$$\text{VHP} = \text{SRS} \cap \text{QALSH}$$



# Stopping Condition

32

- Traditional LSH, C2LSH, SRS-1
  - ▣ Solve  $(c^k r, c)$ -NN queries,  $k = 0, 1, \dots$ 
    - limitations:
      - only  $c^2$  approximate ratio
      - cannot support  $c = 1$
    - ▣ Stopping on either condition:
      - a candidate has distance  $\leq c^k R$
      - there are more than “enough” candidates found

$E_1$

$E_2$



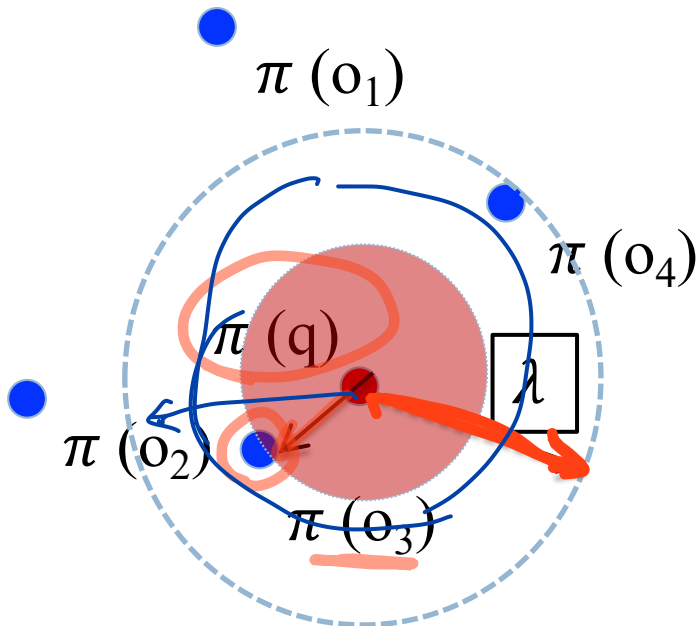
# Stopping Condition

33

- SRS-2 and R2LSH
  - ▣ Accessing objects by increasing order of their ProjDist
  - ▣ Keep the  $o_{\min}$  which has the smallest Dist so far
  - ▣ Stop when ProjDist  $\geq \lambda$  Dist( $O_{\min}$ )

$$\lambda = \frac{1}{c} \sqrt{\Psi_m^{-1}(p_\tau)}$$

Works even for  $c = 1$  !!



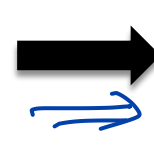
Assume  $\text{Dist}(o_3) = 1$ , then SRS-2 at most scans a hypersphere of radius  $\lambda$

This hypersphere is monotonically shrinking (as we found better  $o_{\min}$ )

# Stopping Condition

Distance Distortion

LSH



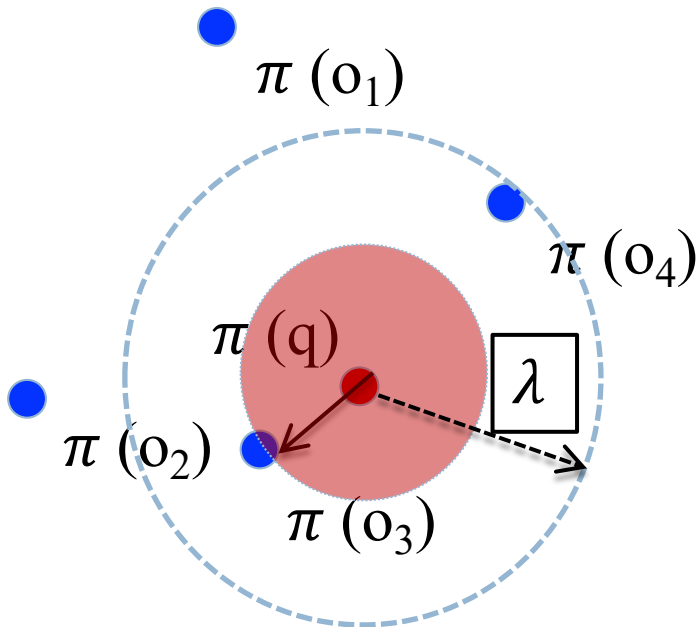
## □ SRS-2 and R2LSH

⇒ Probabilistic Mapping

- Accessing objects by increasing order of their ProjDist
- Keep the  $o_{\min}$  which has the smallest Dist so far
- Stop when  $\text{ProjDist} \geq \lambda \text{Dist}(O_{\min})$

$$\lambda = \frac{1}{c} \sqrt{\Psi_m^{-1}(p_\tau)}$$

Works even for  $c = 1$  !!



Assume  $\text{Dist}(o_3) = 1$ , then SRS-2 at most scans a hypersphere of radius  $\lambda$

This hypersphere is monotonically shrinking (as we found better  $o_{\min}$ )

# Stopping Condition

35

- I-LSH (upon QALSH)
  - ▣ Solve  $(r_k, c)$ -NN queries, where the  $r_k$  sequence is obtained according to the data near  $\pi(Q)$ 
    - obtains c-ANN
  - ▣ Stopping on either condition:
    - $\text{Dist}(o_{\min}) \leq \lambda r$ , where  $2r$  is the "current" virtual bucket width
    - there are more than "enough" candidates found

# Comment

36

- Easy to relax the LSH method in practice at the cost of no worst-case guarantees
  - **E2LSH**: use fewer number of random projections
  - **Multiprobe LSH** (entropyLSH and other variants): space-time tradeoff
  - **LSH in practice**: use empirically tuned parameters ( $k, l$ )
  - **HD-index**: space filling curves as pseudo-LSH functions
  - **SK-LSH**: Replace LSB-tree/forest by a dimension-wise linear mapping
  - ...

# Data-sensitive Hashing

37

- LSH is data-insensitive
  - ▣ Indexing hyper-parameters determined by the shape of the data only
  - ▣ Indexing parameters are randomly generated
- Efforts to make data-sensitive, LSH-like methods
  - ▣ [AR15]
    - Aim: break the lower bounds of  $\rho$
  - ▣ DSH
  - ▣ OPFA / NeOPFA
- Learning-to-hash methods
  - ▣ NSH [PCM15]
  - ▣ [LYZX+18] and many in the ML/CV communities

c.f., [AIR18]

c.f., <https://learning2hash.github.io> and <https://cs.nju.edu.cn/lwj/L2H.html>

# DSH [GJLO14]

38

- Learn a family of (hash) functions,  $H$ , that preserves  $k$ NN of queries

1. Training data:  $\rightarrow \mathbf{W}_{ij} = \begin{cases} 1 & , \text{ if } o_j \in kNN(q_i) \\ -1 & , \text{ if } o_j \notin kNN(q_i) \wedge o_j \text{ is sampled} \\ 0 & \text{ otherwise.} \end{cases}$

- sampled queries

- their  $k$ -NN objects (+ve)

- samples non- $c*k$ -NN objects (-ve)

2. Function family:

- Thresholded linear functions  $h(\mathbf{x}; \mathbf{a}) = \text{sgn}(\mathbf{a}^\top \mathbf{x})$

3. Learn one hash function

$$\arg \min_h \sum_i \sum_j \ell(q_i, o_j) \mathbf{W}_{ij} \quad , \text{ where } \ell(q, o) = (h(q) - h(o))^2$$

# DSH [GJLO14]

39

- Learn a family of (hash) functions,  $H$ , that preserves  $k$ NN of queries
- 4. Learn multiple hash functions
  - Multiplicative updates on  $W_{ij}$ 
    - Increase  $W_{ij}$  if incorrectly classified
    - Decrease  $W_{ij}$  if correctly classified
  - (Under some assumptions) obtain  $H$  that satisfies the  $(k, ck, p_1, p_2)$ -sensitive property for the training data
    - if  $o \in \text{NN}(q, k)$ , then collision probability from  $H \geq p_1$
    - if  $o \notin \text{NN}(q, ck)$ , then collision probability from  $H \leq p_2$

Key difference with AdaBoost: High recall and low precision

# Learned ANN Index [LZSW+20]

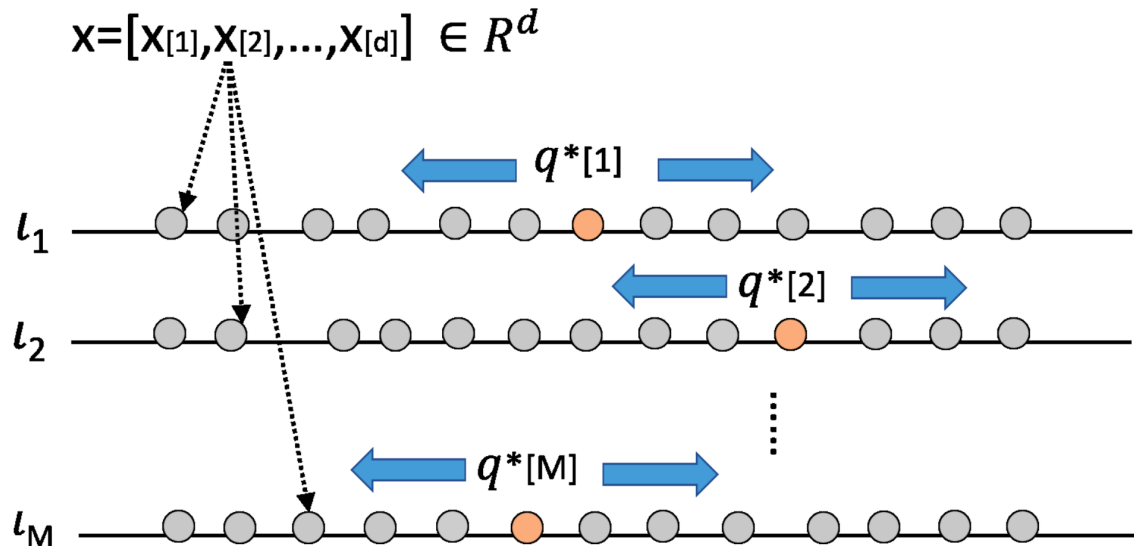
40

- Focus on **external I/O**
  - ▣ Use B-trees and maximize the use of sequential I/Os
- Scheme:
  - ▣  $H: \mathbb{R}^d \rightarrow \mathbb{R}^M$
  - ▣ Index each dimension of  $H(X)$  in a **clustered** B-tree
- Query processing

$\approx$  MedRank  
(minfreq = 1)  
[FKS03]

must c.f., [AFKPS08]

- ▣ Collect candidates on each of the  $M$  projected dimensions
- ▣ When  $T$  candidates are seen on **all**  $M$  lists, rerank them and return top- $k$





# Function family

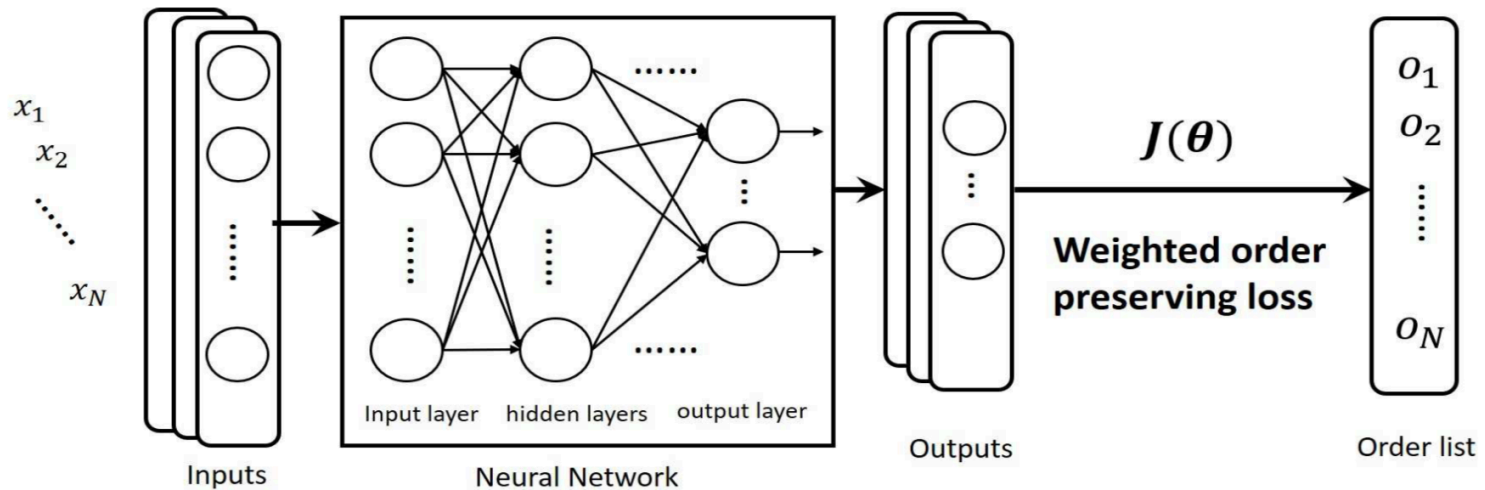
41

## □ Consider

### ▣ linear functions

■  $H(x)[m] = \mathbf{w}_m^T \mathbf{x}$

### ▣ non-linear functions



# How to learn the parameters?

42

Consider the linear functions:  
 $H(x)[m] = \mathbf{w}_m^T \mathbf{x}$

- Goal:
  - ▣ Encourage **segment**-order preserving mappings

Part of the Loss function

$$J^*(\mathbf{w}_m) = \sum_{i=1}^L \sum_{\tilde{x} \in l_i^o} \mathbf{1}_{r(\tilde{x}) \in [t \cdot (i-1), t \cdot i]}$$

Continuous Relaxation

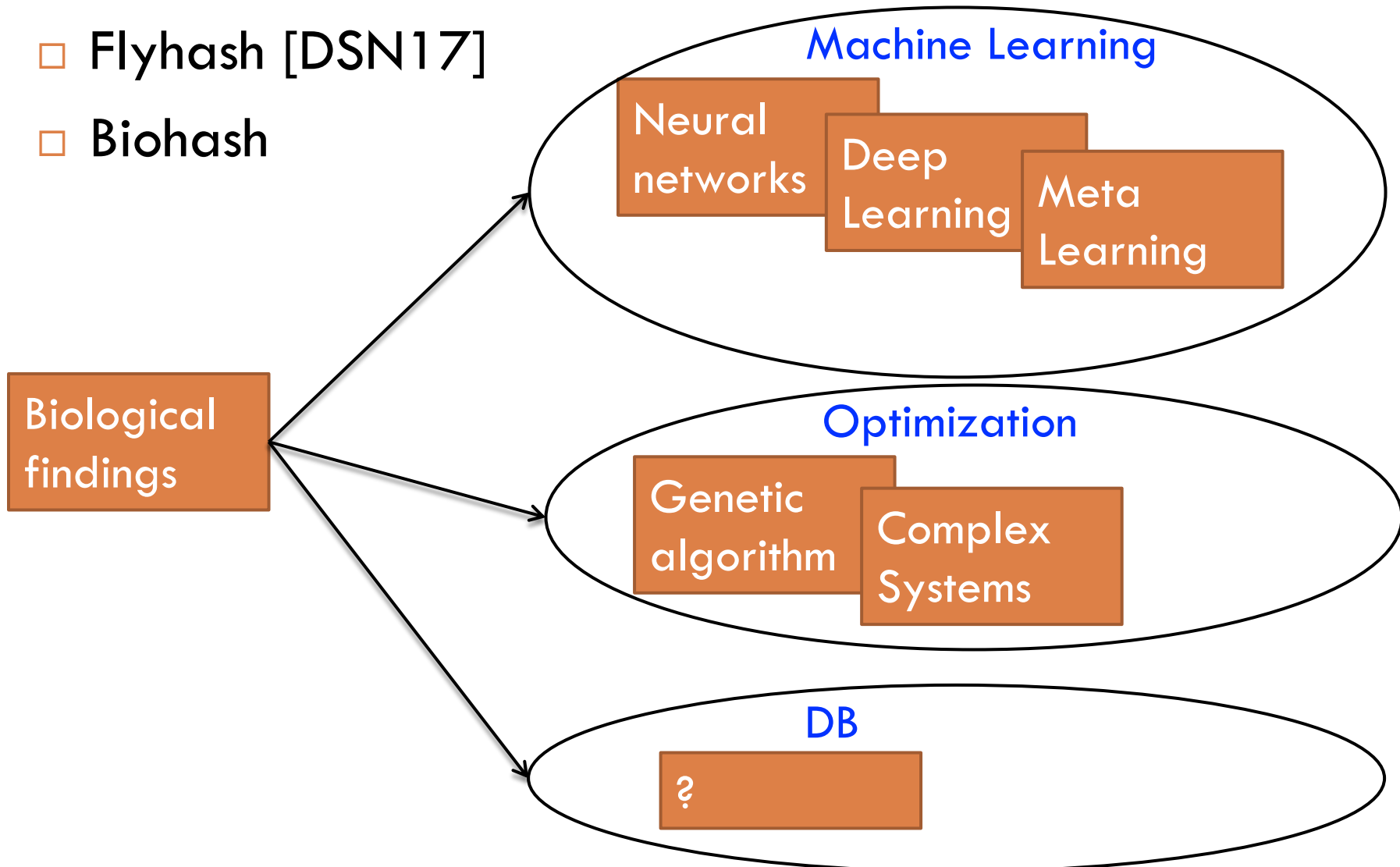
mapped  $x$  in the  $i$ -th segment

$x$  in the  $i$ -th segment

# Biology Inspired Hashing

43

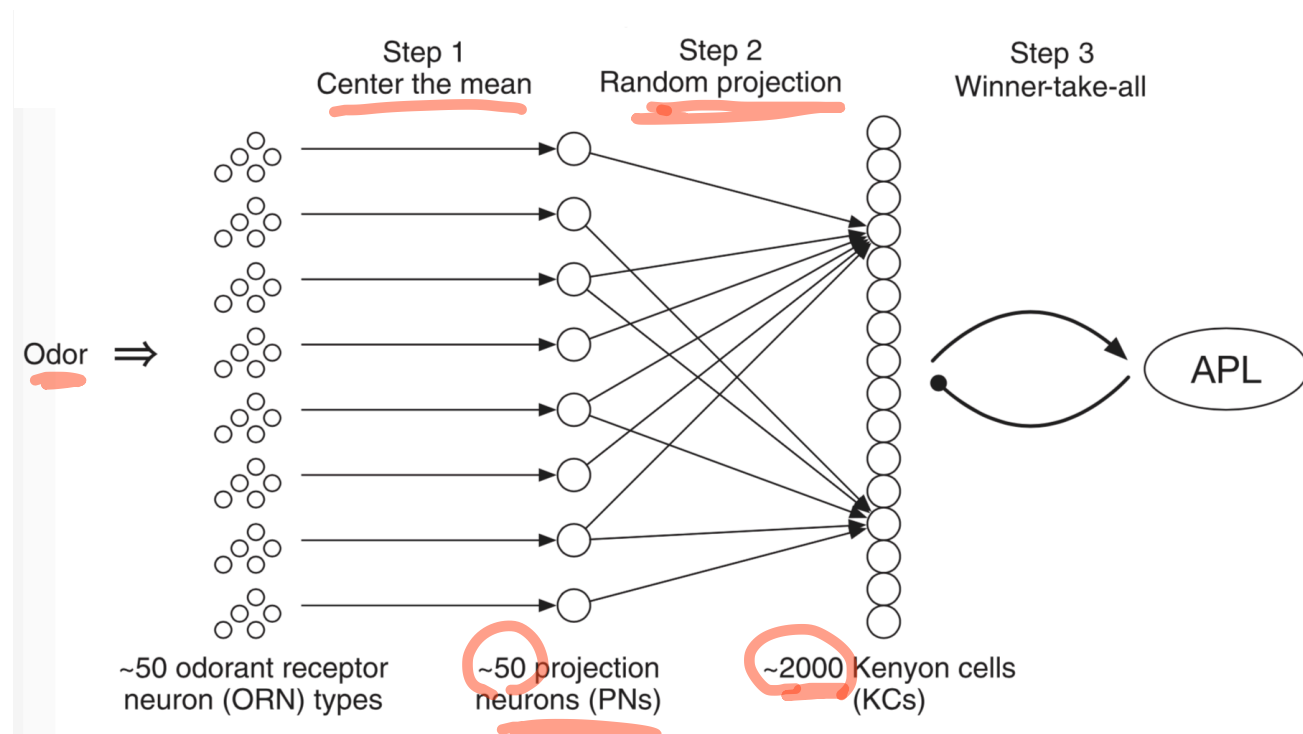
- Flyhash [DSN17]
- Biohash



# FlyHash

44

- The fly olfactory circuit generates a **low-overlapping, sparse** neuron activation pattern when an odor is presented



# FlyHash

45

## □ Difference with LSH

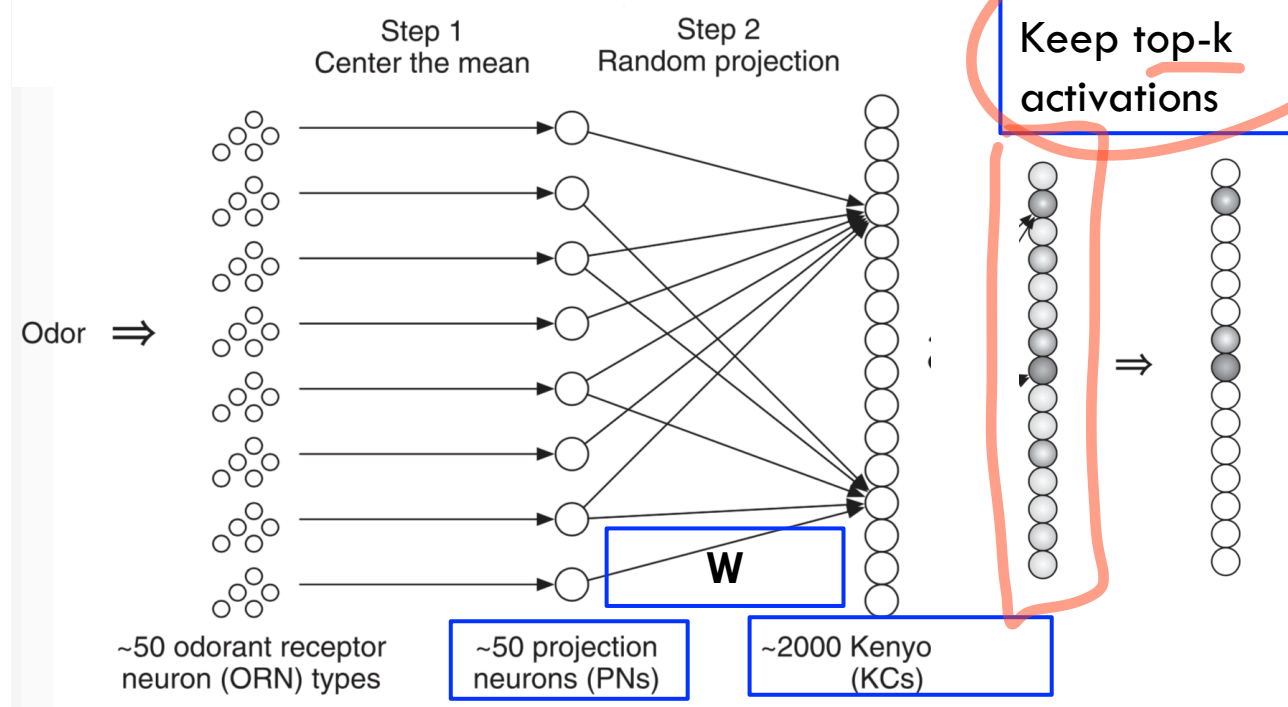
- W is a sparse binary random matrix
- Dimensionality expansion !!
- Sparsification

## □ L2 distance approximately preserved in expectation

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x})$$

*Handwritten notes:  $z_4$ ,  $z_5$  are circled in red.*

Enable



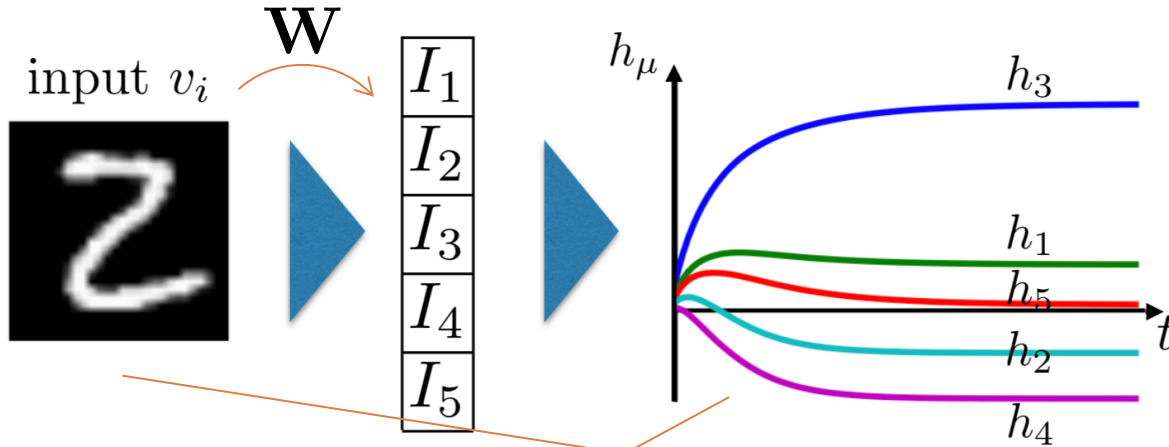
# [KH19]

46

- Unsupervised learning inspired by biological synaptic plasticity rules
- Overview
  - ▣ (Given  $W$ ) Stabilizing the hidden **competing** neurons
  - ▣ Learning the projection matrix  $W$

# Learning h

47



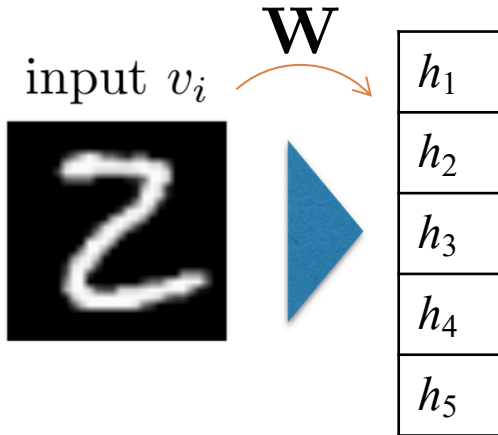
$$\tau \frac{d}{dt} (h_\mu) = I_\mu - \gamma \sum_{\nu \neq \mu} \text{RELU}(h_\nu) - h_\mu, \quad \text{where } I_\mu = \langle \mathbf{W}_\mu, \mathbf{v} \rangle$$

Competing for activation

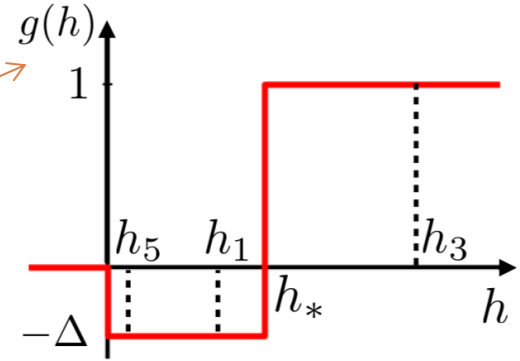
- Fixing the  $\mathbf{W}$ , the dynamical equation will converge to a stable hidden vector  $h$

# Learning $W$

48



$h_*$  is the activation threshold



$$\tau_L \frac{d}{dt} (\mathbf{W}_i) = g(Q) (\mathbf{v}_i - \langle \mathbf{W}, \mathbf{v} \rangle \mathbf{W}_i),$$

where  $Q = \frac{\langle \mathbf{W}, \mathbf{v} \rangle}{\langle \mathbf{W}, \mathbf{W} \rangle^{\frac{1}{2}}}$

$p = 2$ , for any one hidden neuron  
 $\mathbf{W}$  is its corresponding weight vector  
 $R = 1$

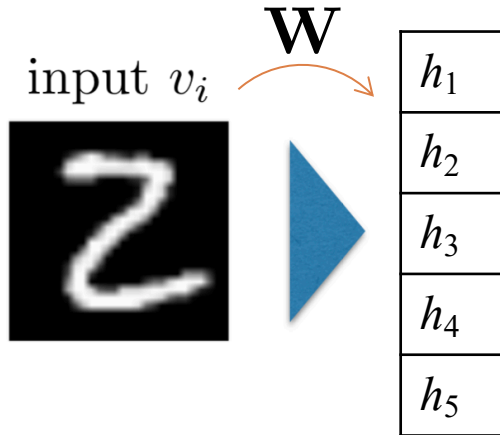
Force  $|\mathbf{W}|_p$  to converge to  $R^p$

- Fixing the  $h$ , the dynamical equation will converge to a final weight matrix  $W$



# BioHash – Learning $\mathbf{W}$

49



sort the projections and rank them

$$g[z] = \begin{cases} 1, & \text{rank}(z) = 1 \\ -\Delta, & \text{rank}(z) = r \\ 0, & \text{otherwise} \end{cases}$$

$$\tau_L \frac{d}{dt} (\mathbf{W}_i) = g(\{Q\}) (\mathbf{v}_i - \langle \mathbf{W}, \mathbf{v} \rangle \mathbf{W}_i), \quad \text{where } Q = \frac{\langle \mathbf{W}, \mathbf{v} \rangle}{\langle \mathbf{W}, \mathbf{W} \rangle^{\frac{1}{2}}}$$

sorting among different  
normalized inner products (each  
produced by a different  $\mathbf{W}_i$ )

- The rest is the same as FlyHash (i.e., WTA sparsification)