Paper Sharing

Product Quantization for Nearest Neighbor Search

杨璇 20200826

1. Motivation
2. Ideas
3. Experiments
4. Conclusion

1. Motivation

Accelerating ANNS, reduce memory usage of indexing structure.

More specifically:

Data Compression: PQ -> reduce memory usage

Exhaustive Search: SDC, ADC -> efficient distance calculation

Non-Exhaustive Search: IVFADC -> avoid exhaustive search

ZILLIZ

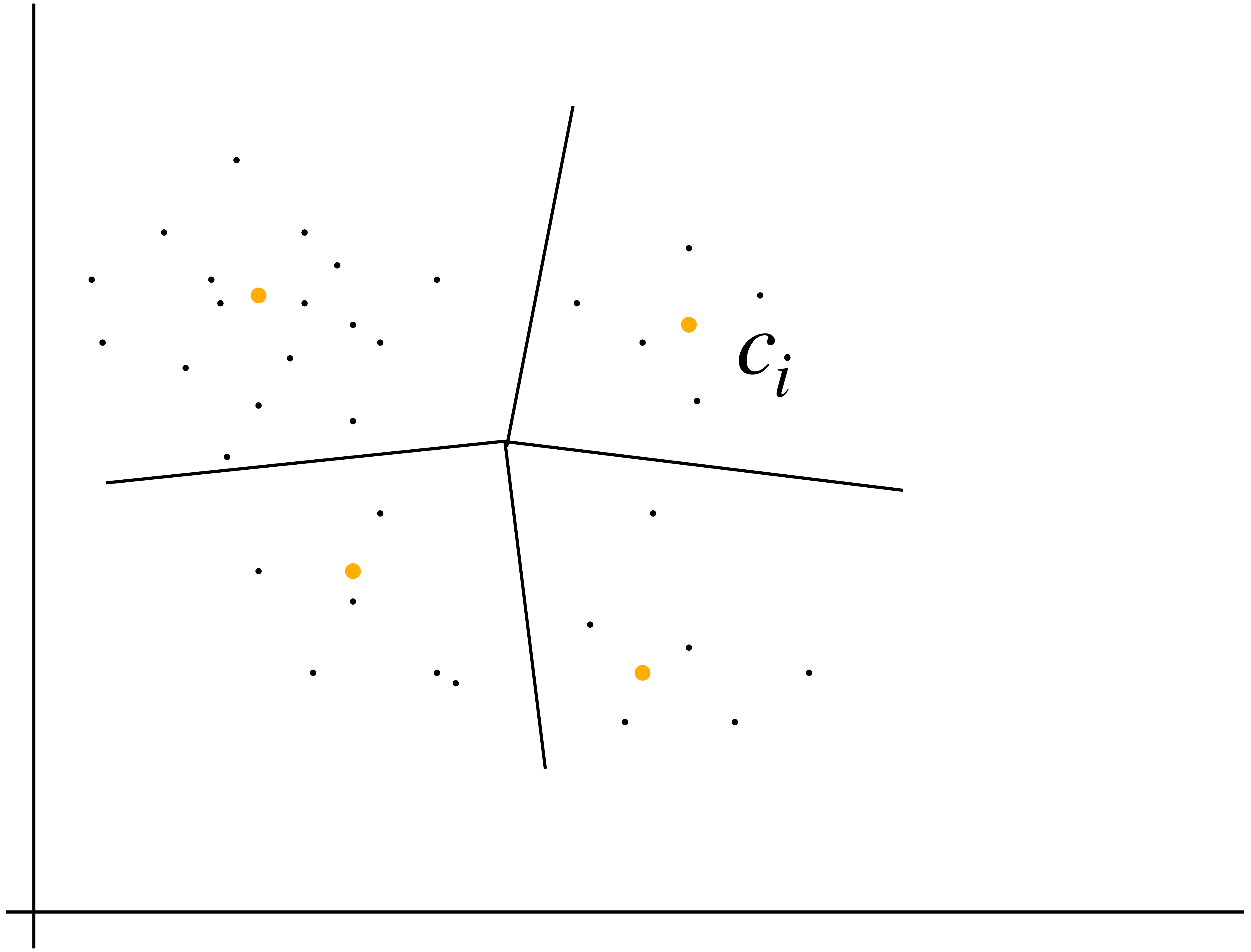
2. Idea

VQ & PQ

ZILLIZ

2. Idea

VQ & PQ



when the input data is real-valued. Formally, a quantizer is a function q mapping a D-dimensional vector $x \in \mathbb{R}^D$ to a vector $q(x) \in \mathcal{C} = \{c_i; i \in \mathcal{I}\}$, where the index set $\mathcal{I}$ is from now on assumed to be finite: $\mathcal{I} = 0 \ldots k-1$. The reproduction values c_i are called *centroids*. The set of reproduction values $\mathcal{C}$ is the *codebook* of size k.

The set $\mathcal{V}_i$ of vectors mapped to a given index i is referred to as a (Voronoi) *cell*, and defined as

$$\mathcal{V}_i \triangleq \{x \in \mathbb{R}^D : q(x) = c_i\}. \tag{2}$$

The k cells of a quantizer form a partition of $\mathbb{R}^D$. By definition, all the vectors lying in the same cell $\mathcal{V}_i$ are reconstructed by the same centroid c_i. The quality of a quantizer

ZILLIZ

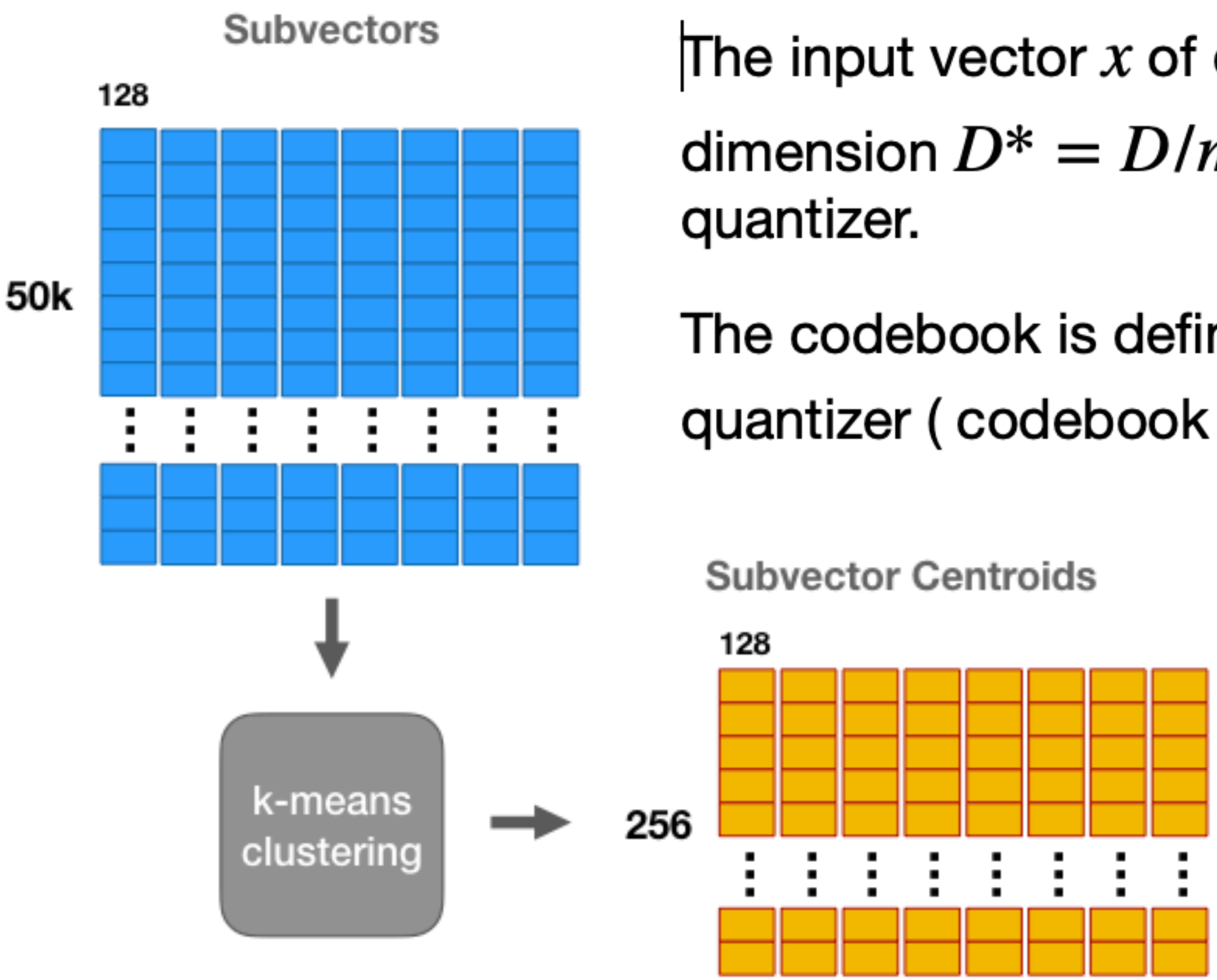
2. Idea

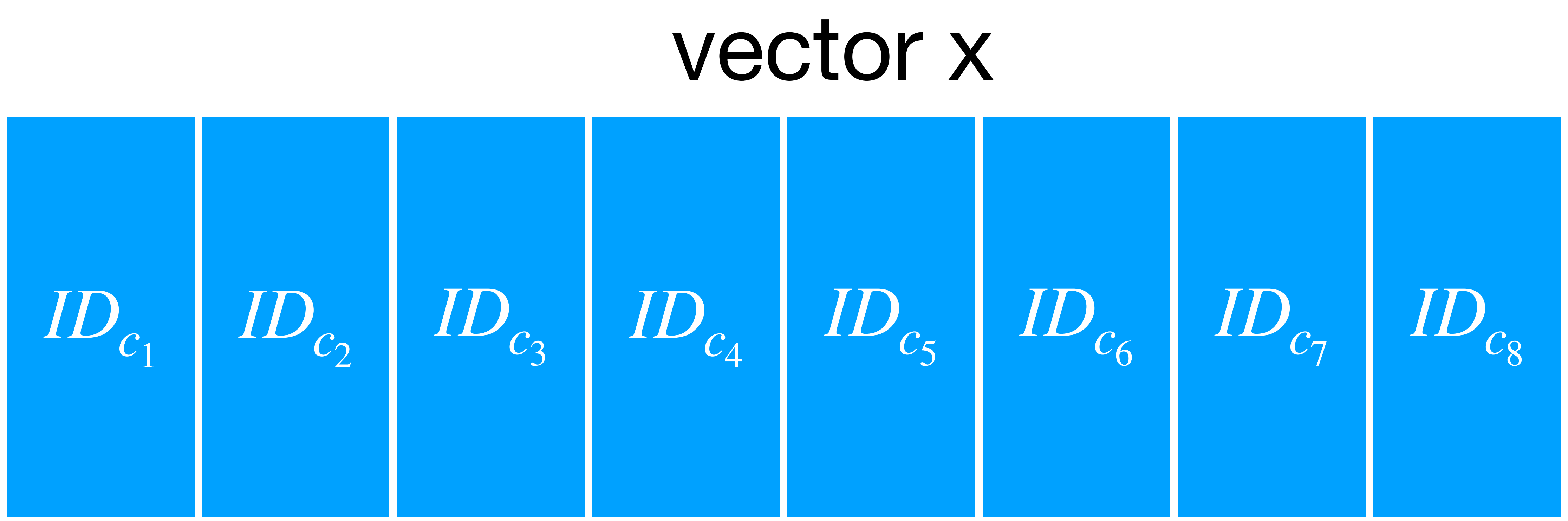
VQ & PQ

PQ can choose the number of components to be quantized.

The input vector x of dimension D is divided into m distinct sub-vectors u_j of dimension $D^* = D/m$. The sub-vectors are quantized separately using m distinct quantizer.

The codebook is defined as the Cartesian product $C = C_1 \times \ldots \times C_m$, each sub-quantizer (codebook) has k^* codes, the total number of codes are $k = (k^*)^m$.





2. Idea
VQ & PQ memory usage comparation

	memory usage	assignment complexity
k-means	$k\, D$	$k\, D$
HKM	$\frac{b_f}{b_f - 1}(k-1)\, D$	$l\, D$
product k-means	$m\, k^*\, D^* = k^{1/m}\, D$	$m\, k^*\, D^* = k^{1/m}\, D$

TABLE I

MEMORY USAGE OF THE CODEBOOK AND ASSIGNMENT COMPLEXITY FOR DIFFERENT QUANTIZERS. HKM IS PARAMETRIZED BY TREE HEIGHT l AND THE BRANCHING FACTOR b_f.

ZILLIZ

2. Idea

Exhaustive search
SDC & ADC

2. Idea

Exhaustive search — SDC & ADC

quantized query vector q

ID_1 ID_2 ID_3 ID_4 ID_5 ID_6 ID_7 ID_8

SDC

vector x

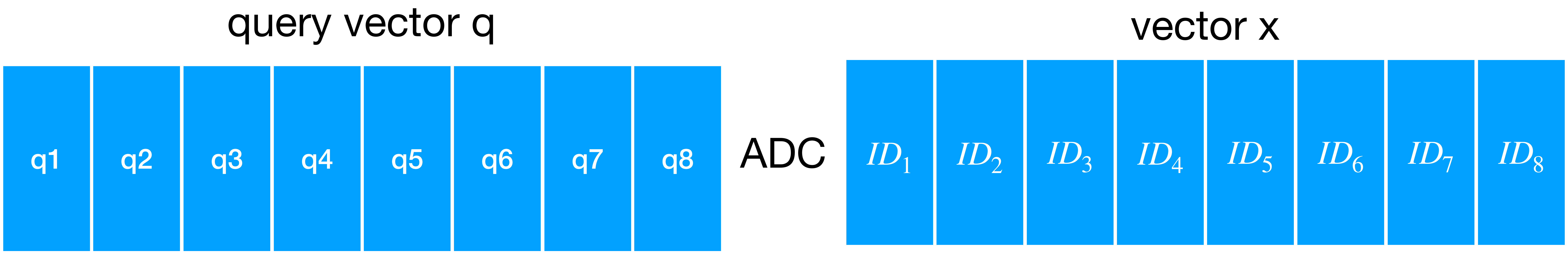
ID_1 ID_2 ID_3 ID_4 ID_5 ID_6 ID_7 ID_8

cm					
…					
c3					
c2					
c1		d1			
	c1	c2	c3	…	cm

ZILLIZ

2. Idea

Exhaustive search — SDC & ADC



	c1	c2	c3	…	cm
q1					

sub-section 1

	c1	c2	c3	…	cm
q2					

sub-section 2

2. Idea

Exhaustive search — SDC & ADC

	SDC	ADC
encoding x	$k^* D$	0
compute $d\big(u_j(x), c_{j,i}\big)$	0	$k^* D$
for $y \in \mathcal{Y}$, compute $\hat{d}(x,y)$ or $\tilde{d}(x,y)$	$n\,m$	$n\,m$
find the k smallest distances	$n + k \log k \log \log n$	

TABLE II

ALGORITHM AND COMPUTATIONAL COSTS ASSOCIATED WITH SEARCHING THE k NEAREST NEIGHBORS USING THE PRODUCT QUANTIZER FOR SYMMETRIC AND ASYMMETRIC DISTANCE COMPUTATIONS (SDC, ADC).

2. Idea

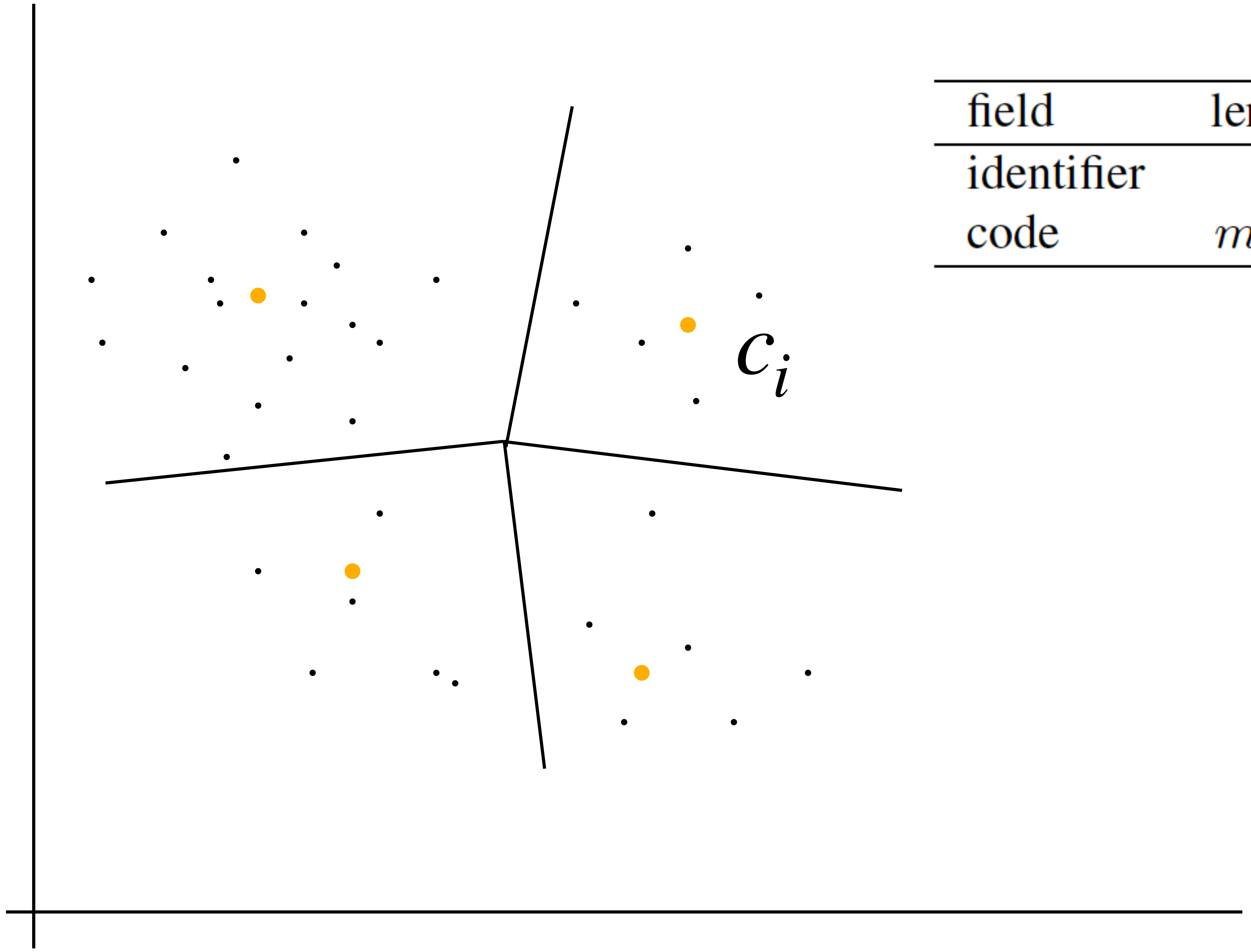
Non Exhaustive search

IVFADC

2. Idea

Non Exhaustive search — IVFADC

- First we apply k-means to learn a codebook of k' centroids (partitions), producing a quantizer q_c. Now each vector belongs to one and only one of the partitions. For every partition, you have a list of all the vectors belong to it (refer to the IVF list)



field	length (bits)
identifier	8–32
code	$m\lceil \log_2 k^* \rceil$

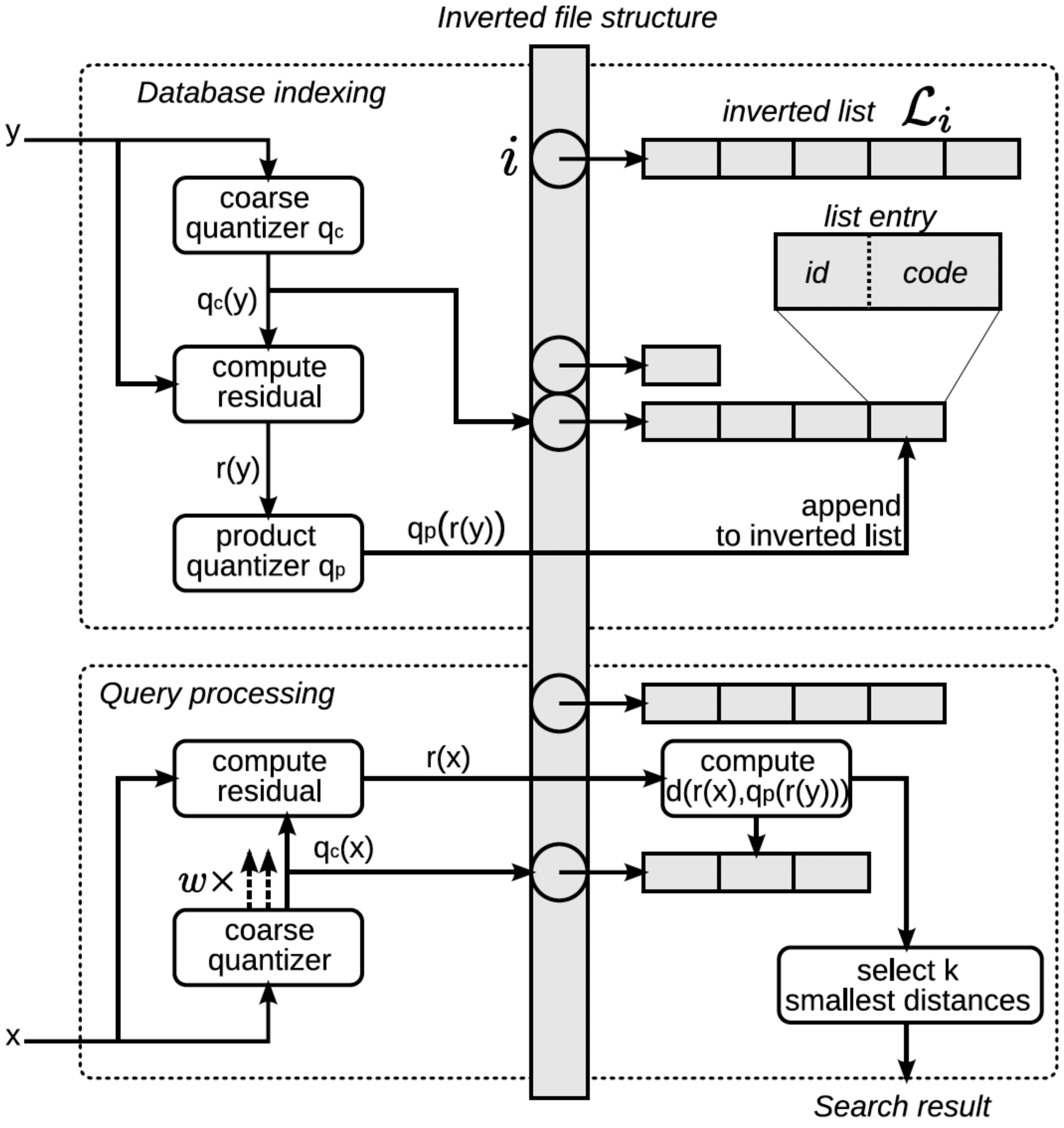
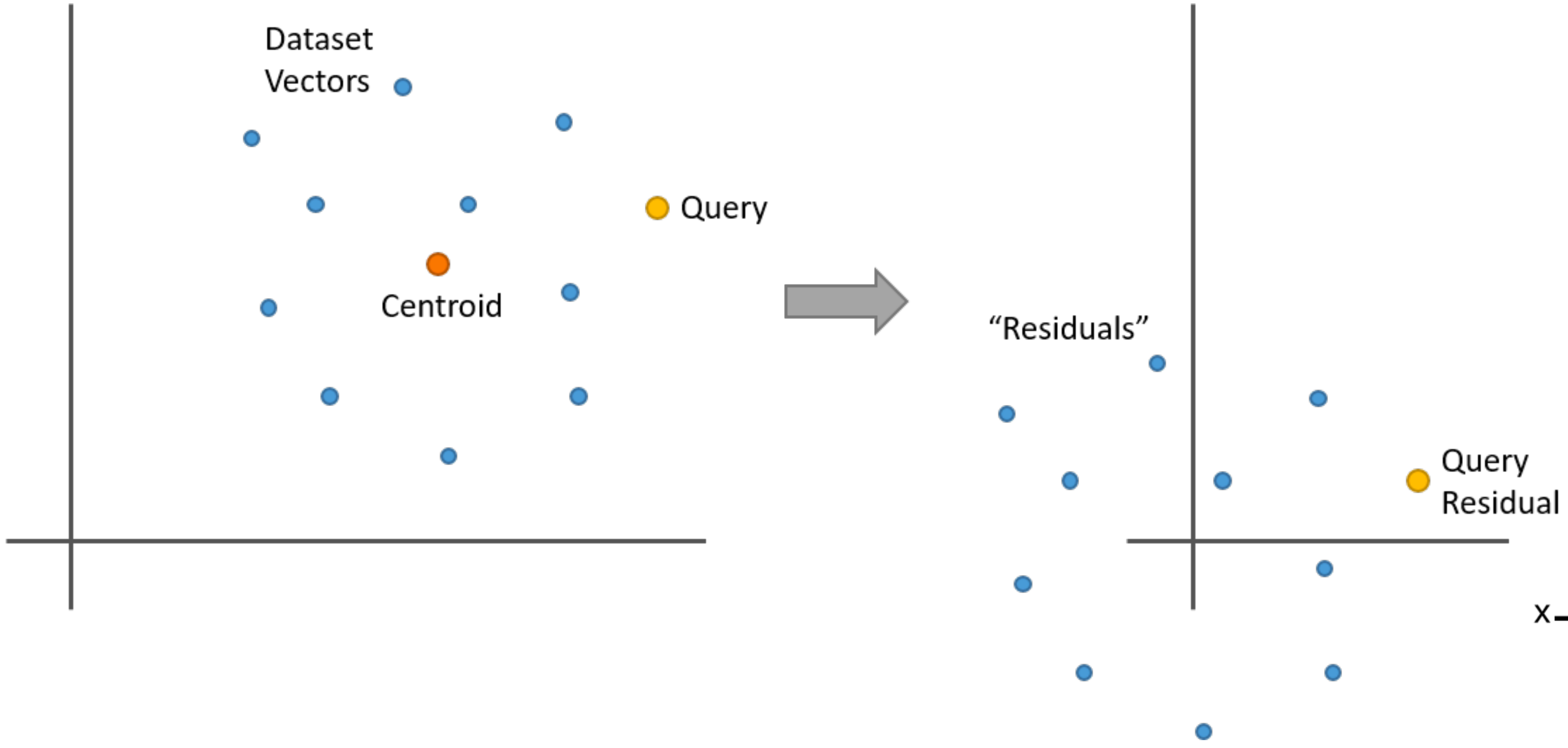


Fig. 5. Overview of the *inverted file with asymmetric distance computation* (IVFADC) indexing system. *Top*: insertion of a vector. *Bottom*: search.

2. Idea

Non Exhaustive search — IVFADC

- For every Voronoi cell (partition), we use PQ to encode the residual of the vector. The residual vector is $r(y) = y - q_c(y)$.



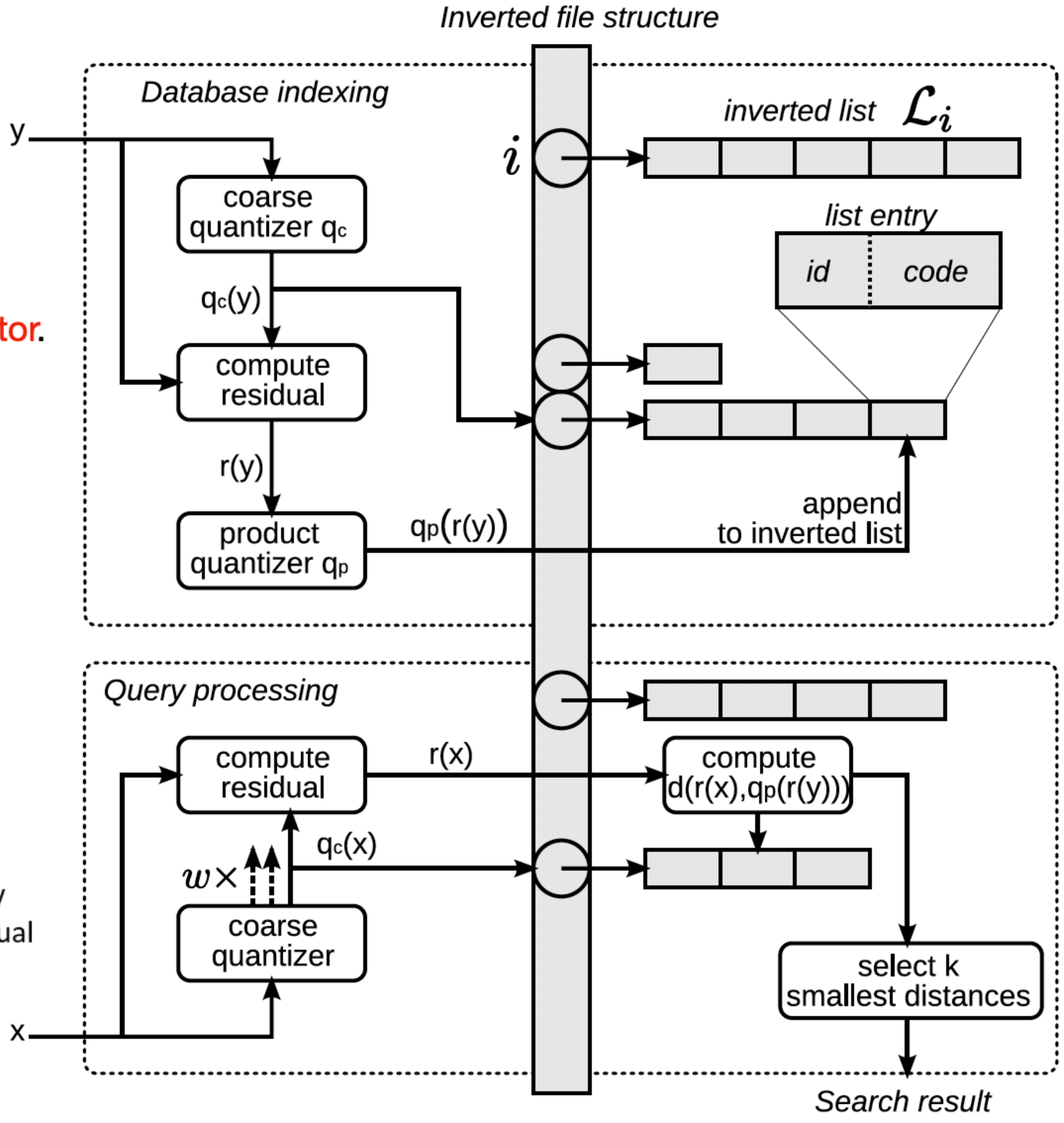


Fig. 5. Overview of the *inverted file with asymmetric distance computation* (IVFADC) indexing system. *Top*: insertion of a vector. *Bottom*: search.

2. Idea

Non Exhaustive search — IVFADC

- For every Voronoi cell (partition), we use PQ to encode the residual of the vector. The residual vector is $r(y) = y - q_c(y)$.

So what's the benefit of it? Since every vector is centered around origin point, the dataset becomes more dense and relatively tightly grouped and reduce the variety of the dataset, and it take fewer "codes" to represent the vectors more efficiently. For another perspective, with a limited number of codes, PQ will be more accurate because the vectors are less distinct than before.

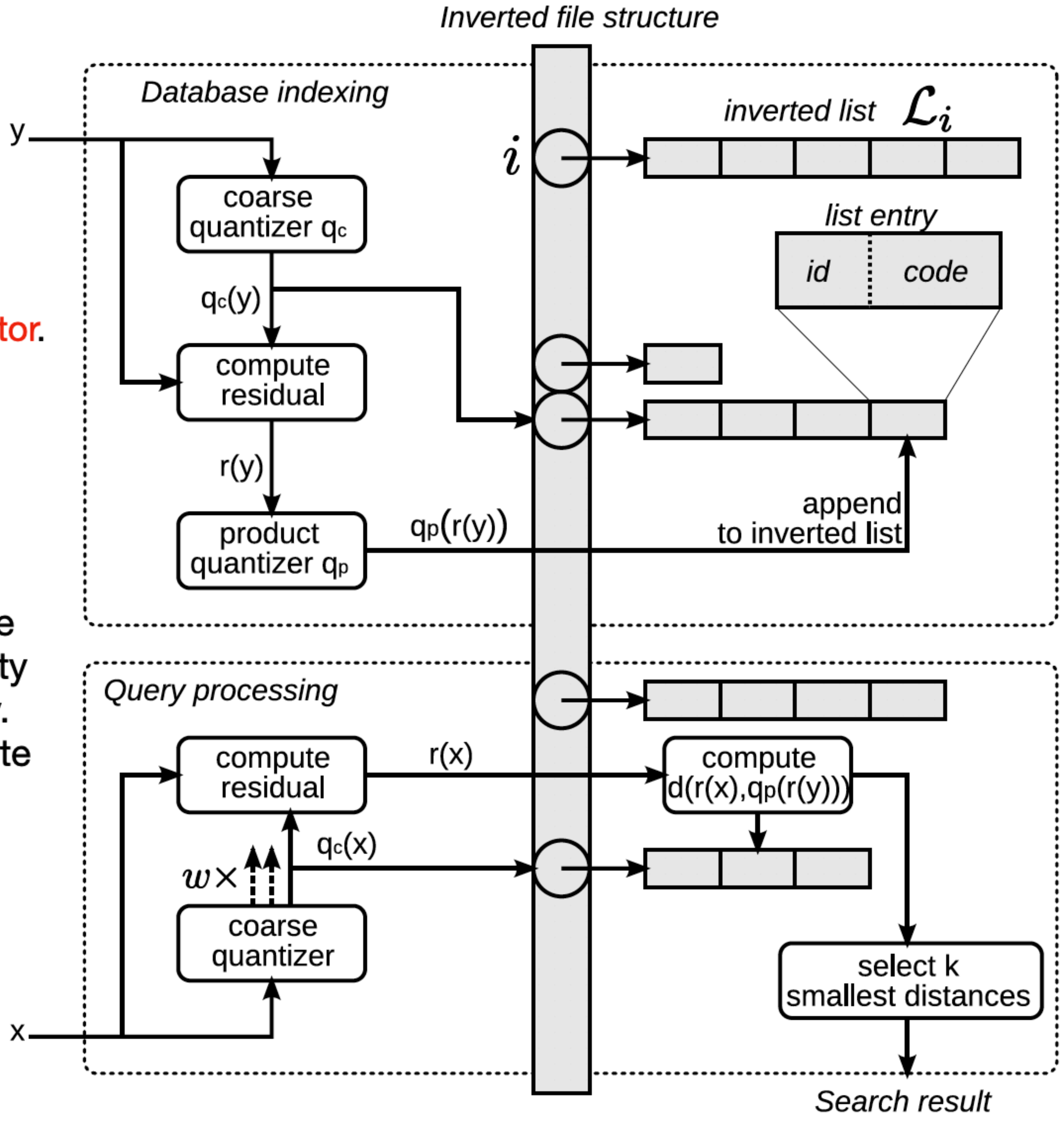


Fig. 5. Overview of the *inverted file with asymmetric distance computation* (IVFADC) indexing system. *Top*: insertion of a vector. *Bottom*: search.

3. Experiments

3. Experiments

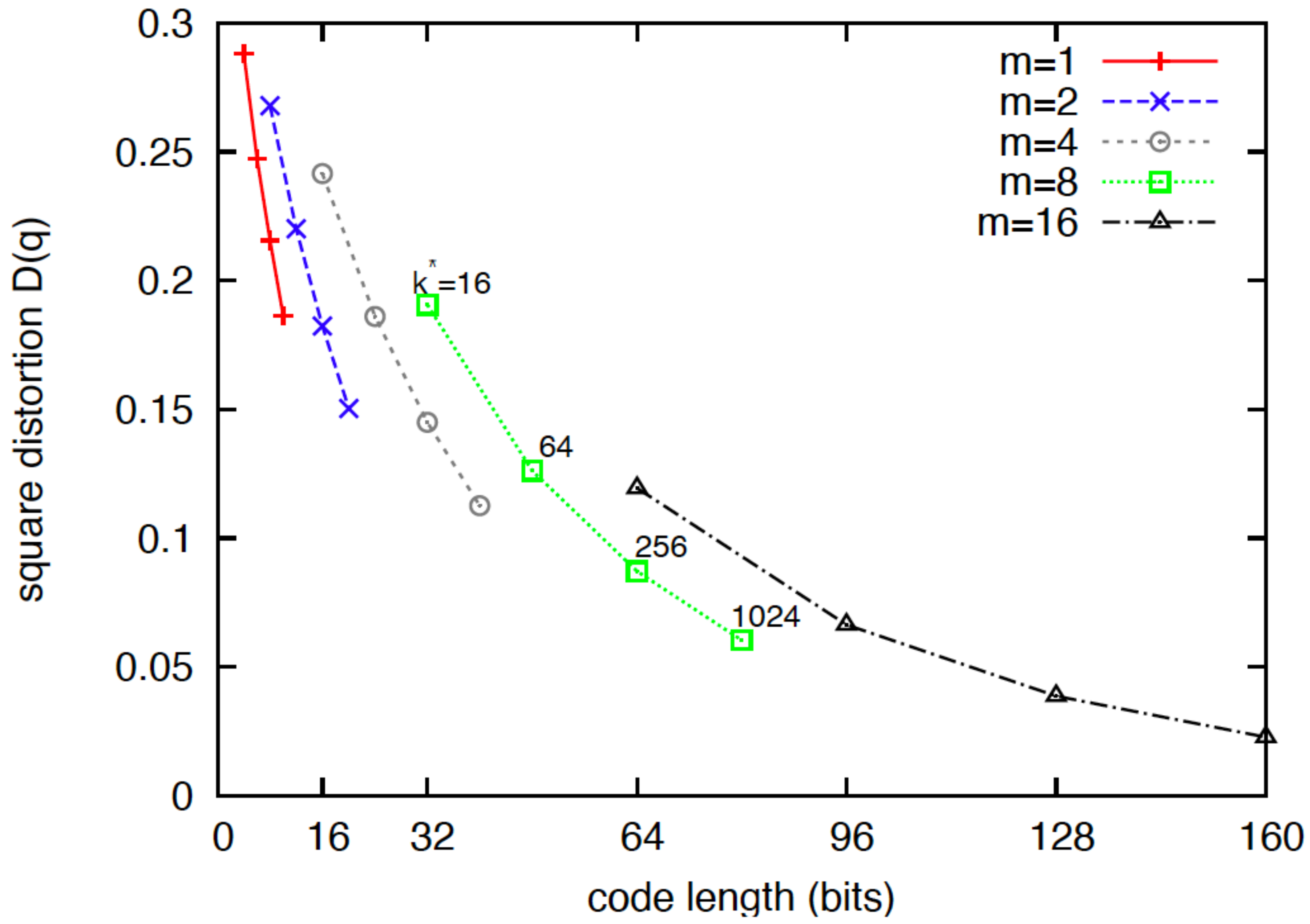


Fig. 1. SIFT: quantization error associated with the parameters m and k^*.

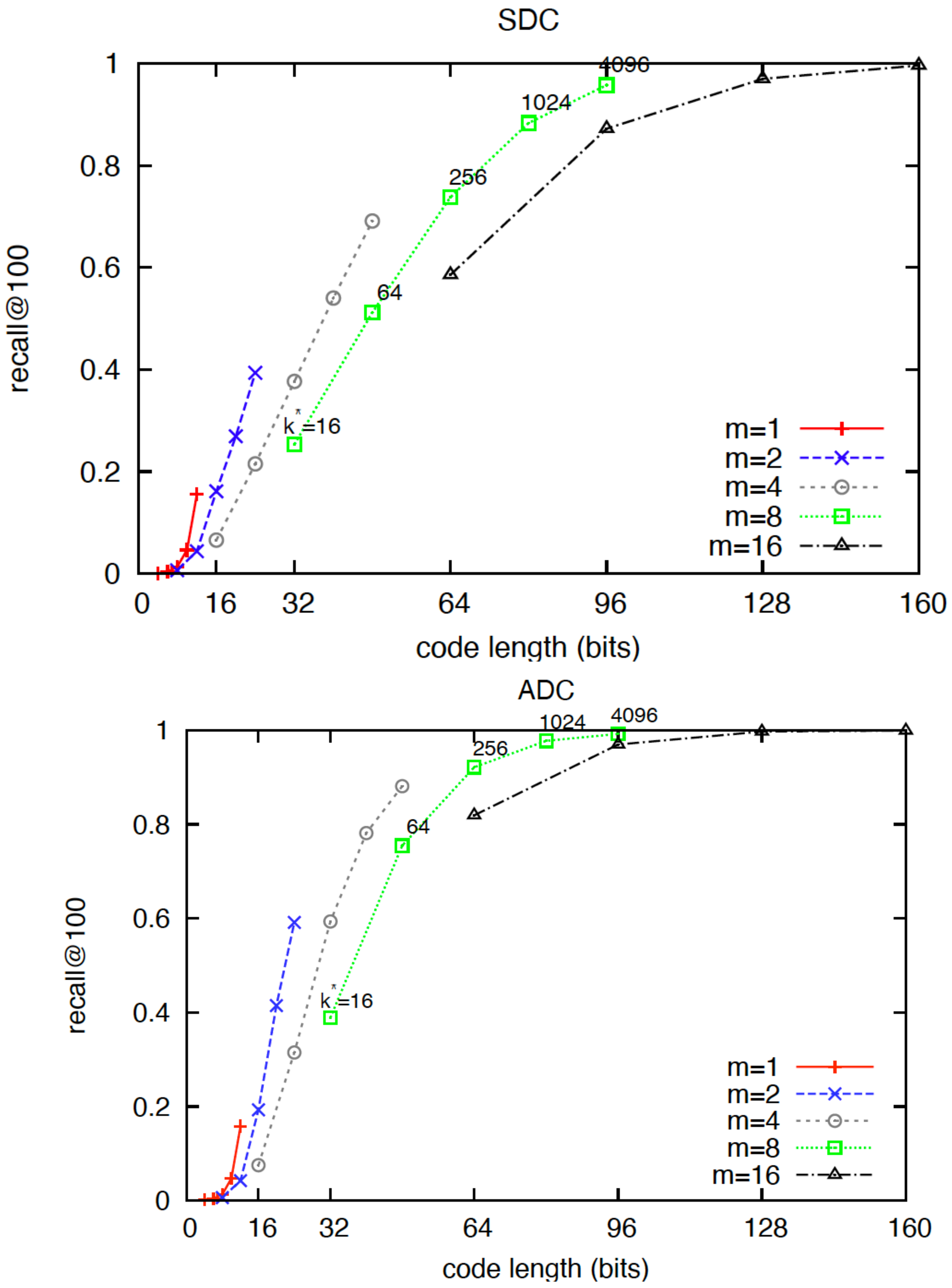


Fig. 6. SDC and ADC estimators evaluated on the SIFT dataset: recall@100 as a function of the memory usage (code length=$m \times \log_2 k^*$) for different parameters (k^*=16,64,256,...,4096 and m=1,2,4,8,16). The missing point (m=16,k^*=4096) gives recall@100=1 for both SDC and ADC.

3. Experiments

Impact of the component grouping

	SIFT		GIST
m	4	8	8
natural	0.593	0.921	0.338
random	0.501	0.859	0.286
structured	0.640	0.905	0.652

TABLE IV

IMPACT OF THE DIMENSION GROUPING ON THE RETRIEVAL PERFORMANCE OF ADC (RECALL@100, k^*=256).

ZILLIZ

4. Conclusion

Introduced PQ for ANNS

Combined with IVF to avoid exhaustive search.

Outperform the state of the art in terms of the trade-off between search quality and memory usage.

[1] Product Quantization for Nearest Neighbor Search
[2] http://mccormickml.com/2017/10/13/product-quantizer-tutorial-part-1/
[3] http://mccormickml.com/2017/10/22/product-quantizer-tutorial-part-2/