Xi'an, Shaanxi, China
SIGMOD/PODS 2021

Learned Cardinality Estimation for Similarity Queries

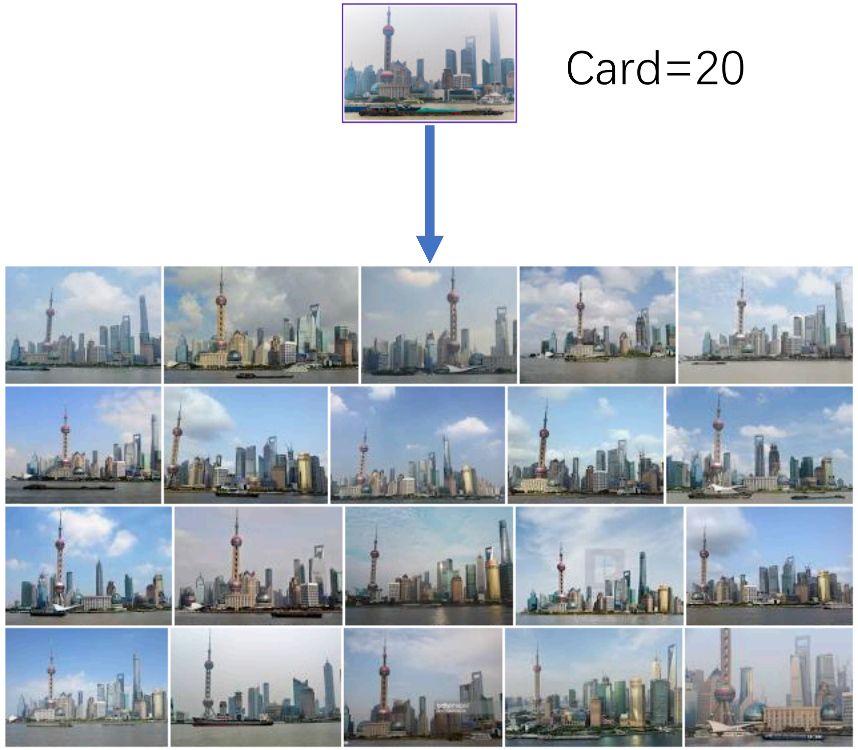
Ji Sun[+], Guoliang Li[+], Nan Tang[*]

[+]Tsinghua University, [*]QCRI

Problem Statement

- **Cardinality for Similarity Search.** Number of objects in D whose distances to a query q are not greater than a distance threshold τ .
- **Cardinality for Similarity Join.** Total number of pairs (q, p) whose distance between $q \in Q$ and $p \in D$ is not greater than τ .

Problem Statement



Card=20

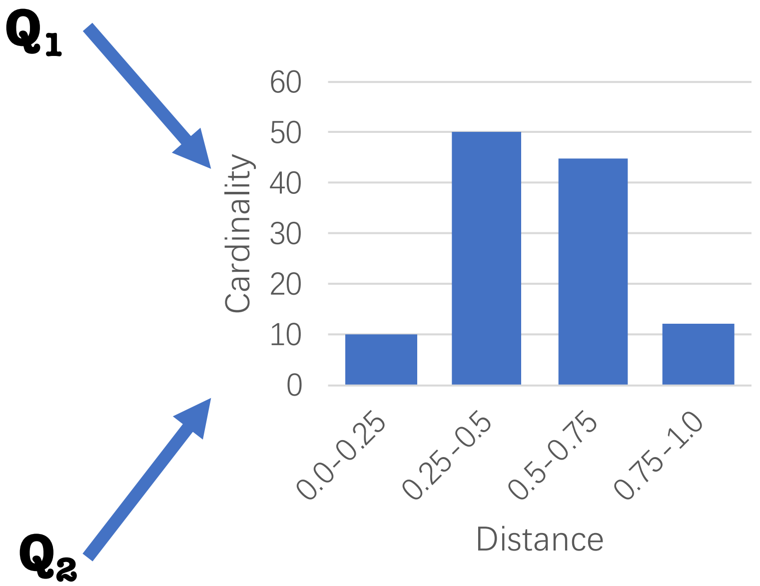
Basketball

Card=9

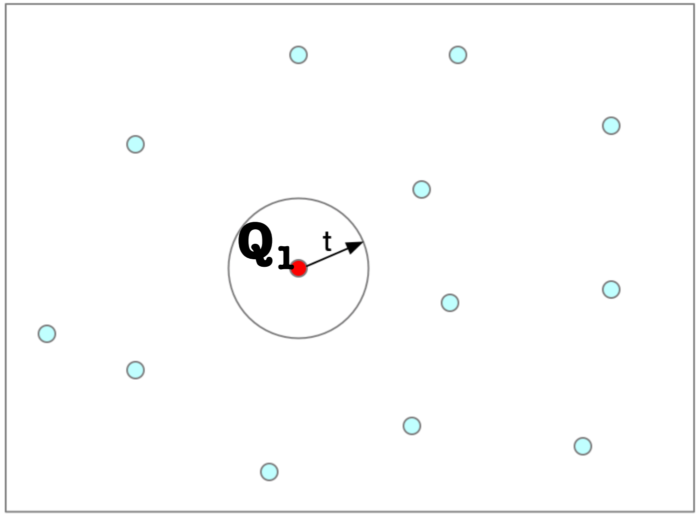
Volleyball	Football	Tennis
Soccer	Softball	Lacrosse
Baseball	Women_B asketball	basketball

Related Work

- Cardinality Estimator for Exact Queries
  - **Histogram**: Relative distance is defined on the given query.
  - **Sampling**: 0-tuple problem for high dimensionality.
  - **Data Model**: Hard to fit the sparse continuous data.



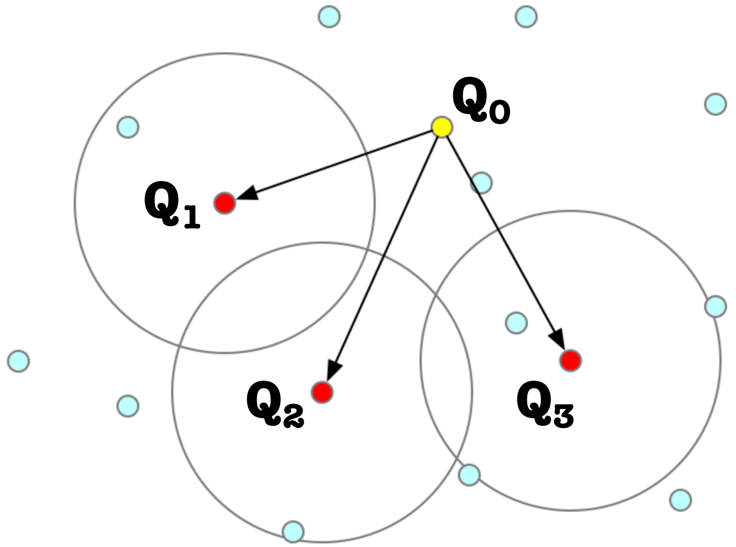
Histogram



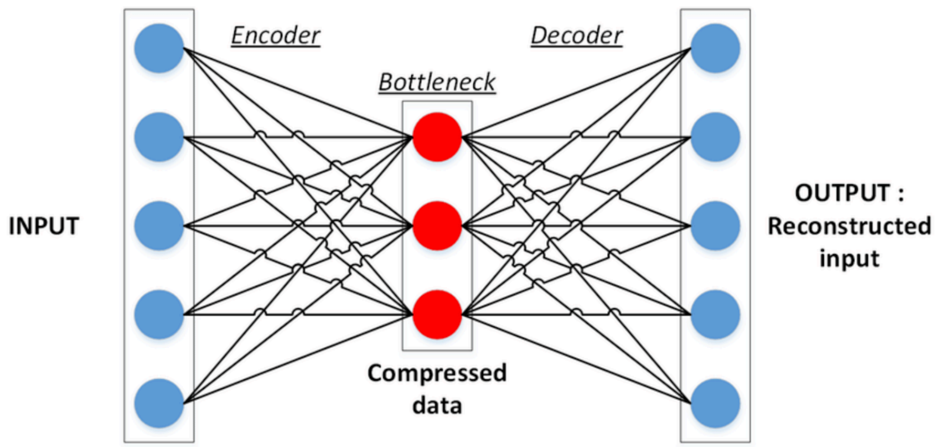
Sampling

Related Work

- Cardinality Estimator for Similarity Queries
  - **KDE**: 0-tuple problem for high dimensionality.
  - **Linear Mixture Model**: Less powerful for high dimensional data.
  - **VAE**: Low dimensional embedding is not a distance-aware representation.

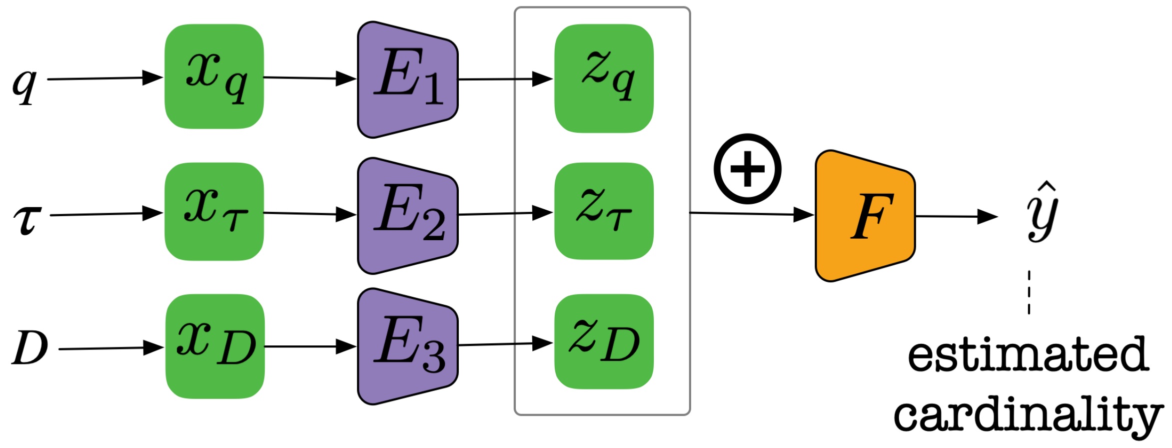


Mixture Model



Variational AutoEncoder(VAE)

Basic Model



q: query vector
$\boldsymbol{\tau}$: diatance threshold
D: data sample
E1, E2, E3, F: Neural Networks

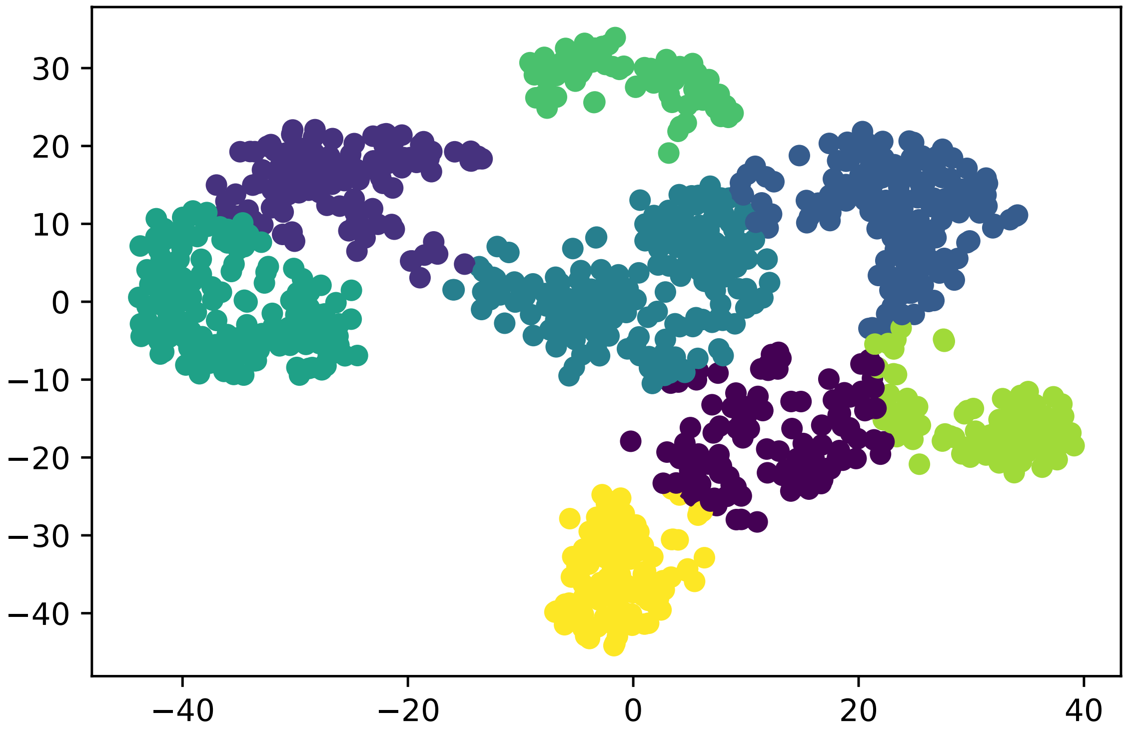
Observations & Opportunities

- Vectors far from the query can be ignored.

- The distance of two vectors is related to sum of distances on vector segments.
    - Hamming(Sigmod, Sigkdd) = Hamming(Sig,Sig)+Hamming(mod,kdd)
    - …

Observations & Opportunities

- Clustering



Fashion-MNIST

Observations & Opportunities

- Distance Decomposition

L_m-distance

$$\mathrm{dis}_{L_m}(u, v) = \sqrt[m]{\sum_{j=1}^{d}(u[j] - v[j])^m}$$
$$= \sqrt[m]{\sum_{i=1}^{n} \sum_{j=|u^{(i)}|\cdot(i-1)}^{|u^{(i)}|\cdot i} (u[j] - v[j])^m} = \sqrt[m]{\sum_{i=1}^{n}(\mathrm{dis}_{L_m}(u^{(i)}, v^{(i)}))^m}$$

Cosine distance

$$\mathrm{dis}_{\cos}(u, v) = 1 - \frac{u \cdot v}{|u| \cdot |v|} = |u| \cdot |v| - u \cdot v$$
$$= \frac{u^2 + v^2 - 2uv}{2} = \frac{\mathrm{dis}_{L_2}(u, v)}{2}$$

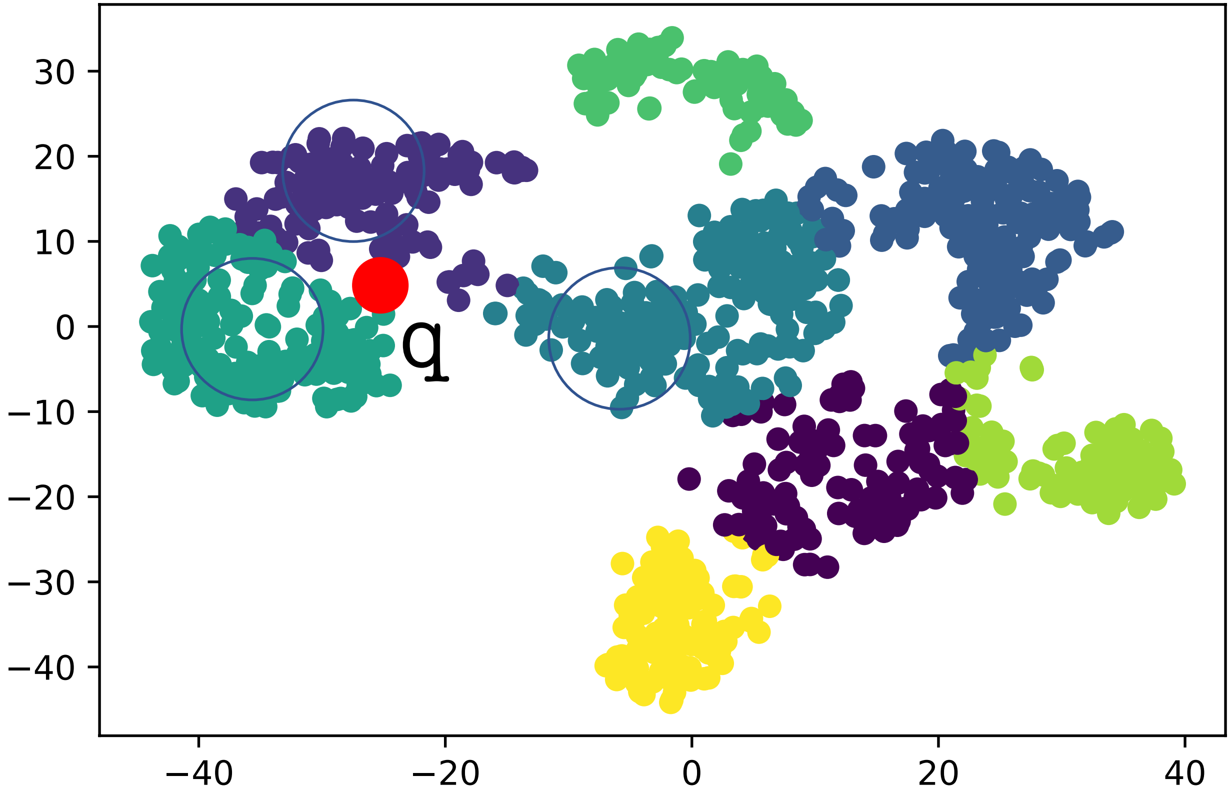
Angular distance

$$\mathrm{dis}_{\mathrm{angular}}(u, v) = \frac{\arccos \mathrm{dis}_{\cos}(u, v)}{\pi}$$

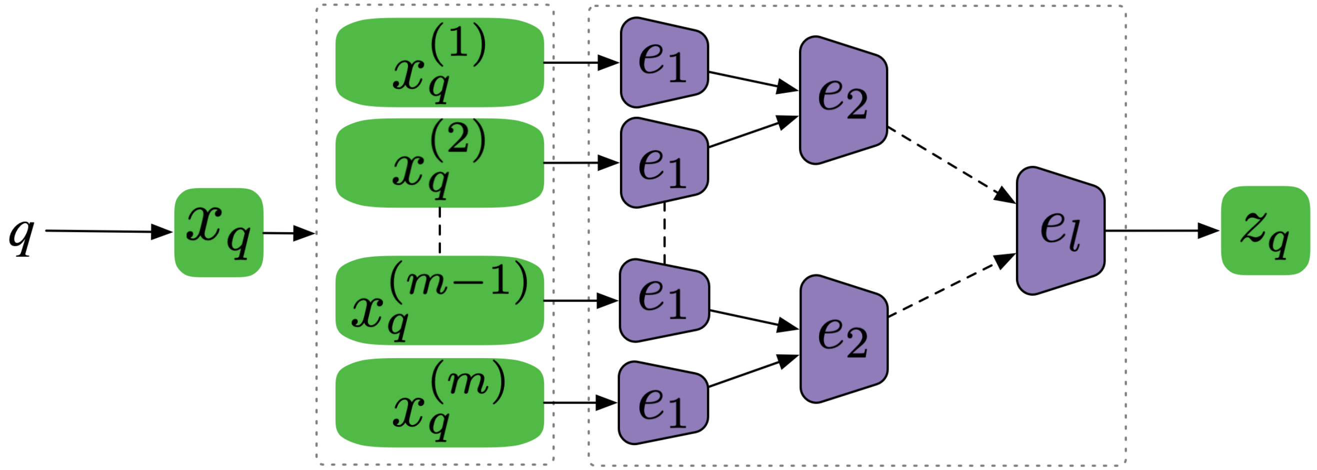
Hamming distance

$$\mathrm{dis}_{\mathrm{ham}}(u, v) = \sum_{j=1}^{d} \mathrm{equal}(u[j], v[j])$$
$$= \sum_{i=1}^{n} \sum_{j=|u^{(i)}|\cdot(i-1)}^{|u^{(i)}|\cdot i} \mathrm{equal}(u[j], v[j]) = \sum_{i=1}^{n} \mathrm{dis}_{\mathrm{ham}}(u^{(i)}, v^{(i)})$$

Data Segmentation

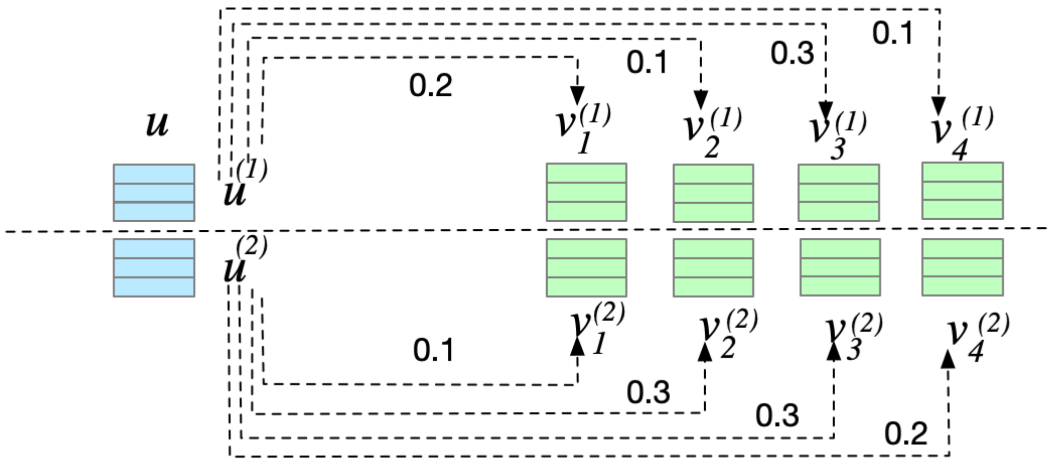


Query Segmentation



q: query
$\mathbf{x_q}$: input vector
$\mathbf{x_q^{(i)}}$: the i-th segment of input vector
e1, e2, el: Neural Networks

Query Segmentation



$u^{(1)}$

Distance	Distribute
0.1	0101
0.2	1000
0.3	0010

$u^{(2)}$

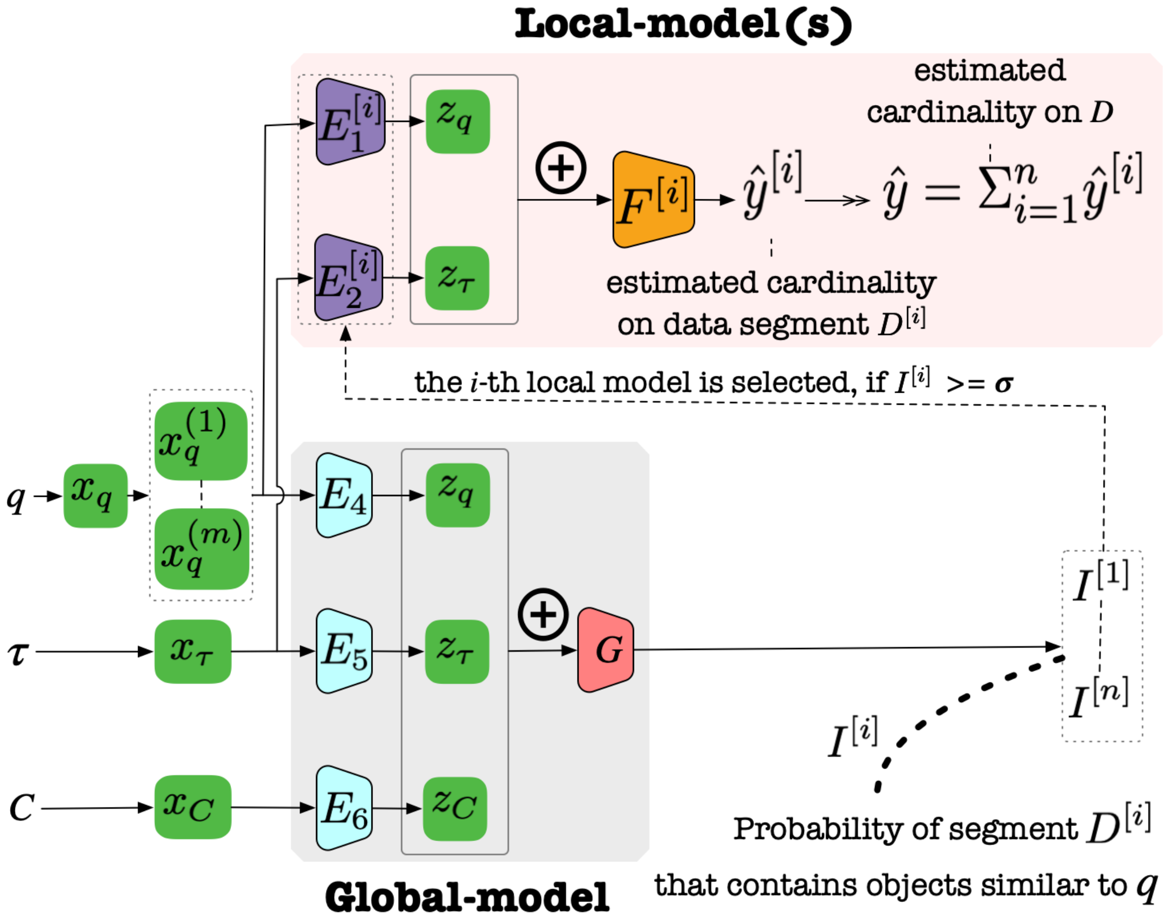
Distance	Distribute
0.1	1000
0.2	0001
0.3	0110

bitand

Distance	Distribute	Density
0.2	0000	0.0
0.3	1001	0.5
0.4	0100	0.25
0.5	0000	0.0
0.6	0010	0.25

function *f*( ) function *g*( )

Global-Local Model



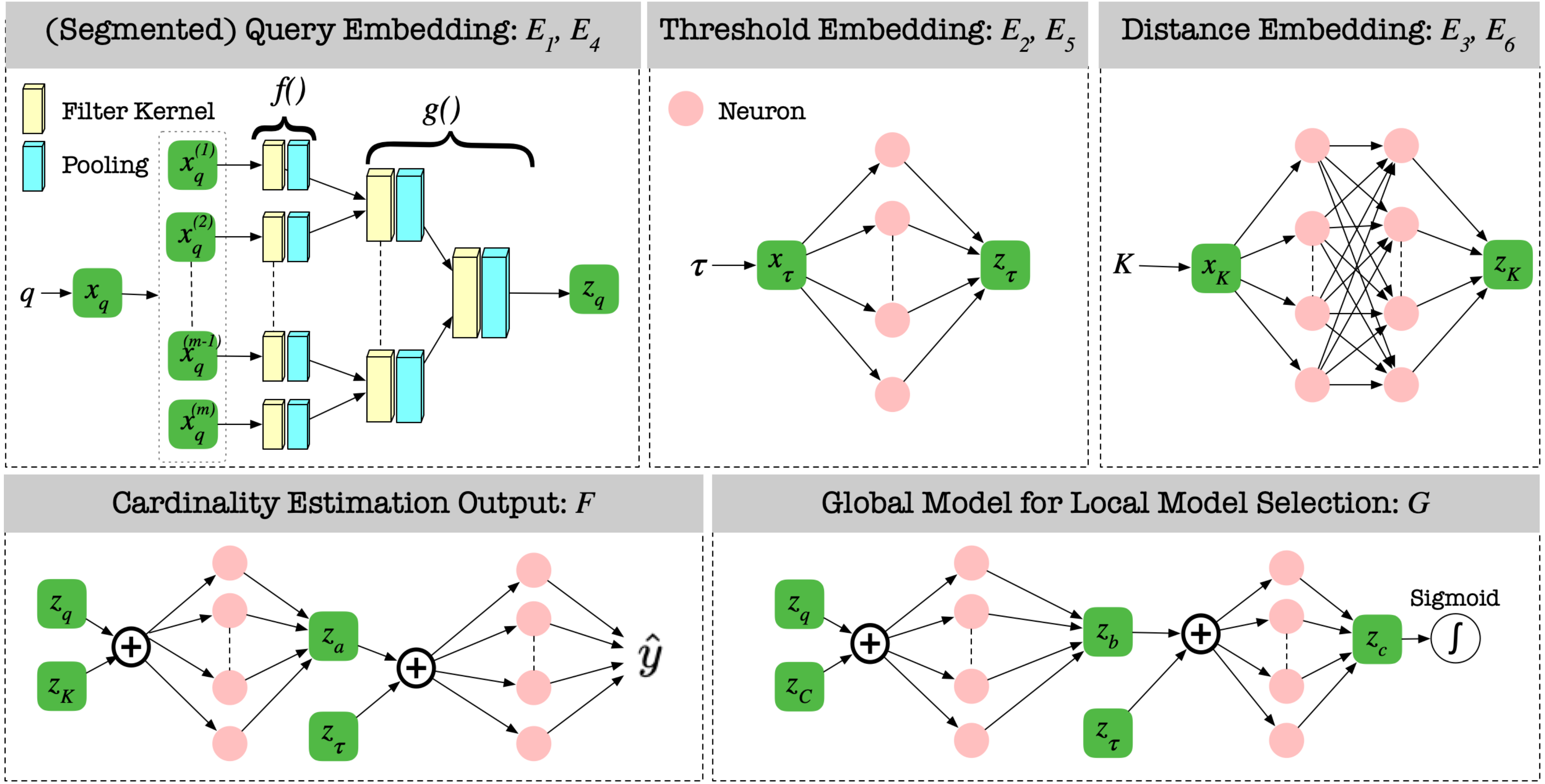
Global Loss Function

$$\epsilon^{\{j\}[i]} = \frac{\text{card}^{\{j\}[i]} - \min_i \text{card}^{\{j\}[i]}}{\max_i \text{card}^{\{j\}[i]} - \min_i \text{card}^{\{j\}[i]}}$$

$$\mathcal{L}(\theta) = \frac{1}{n \times B_S} \sum_{i=1}^{n} \sum_{j=1}^{B_S} R^{\{j\}[i]} \log(I^{\{j\}[i]})(1 + \epsilon^{\{j\}[i]}) + (1 - R^{\{j\}[i]}) \log(1 - I^{\{j\}[i]})$$

$$\mathcal{J}(\theta) = -\frac{1}{n \times B_S} \sum_{i=1}^{n} \sum_{j=1}^{B_S} R^{\{j\}[i]} \log(I^{\{j\}[i]})(1 + \epsilon^{\{j\}[i]}) + (1 - R^{\{j\}[i]}) \log(1 - I^{\{j\}[i]})$$

Implementation Details

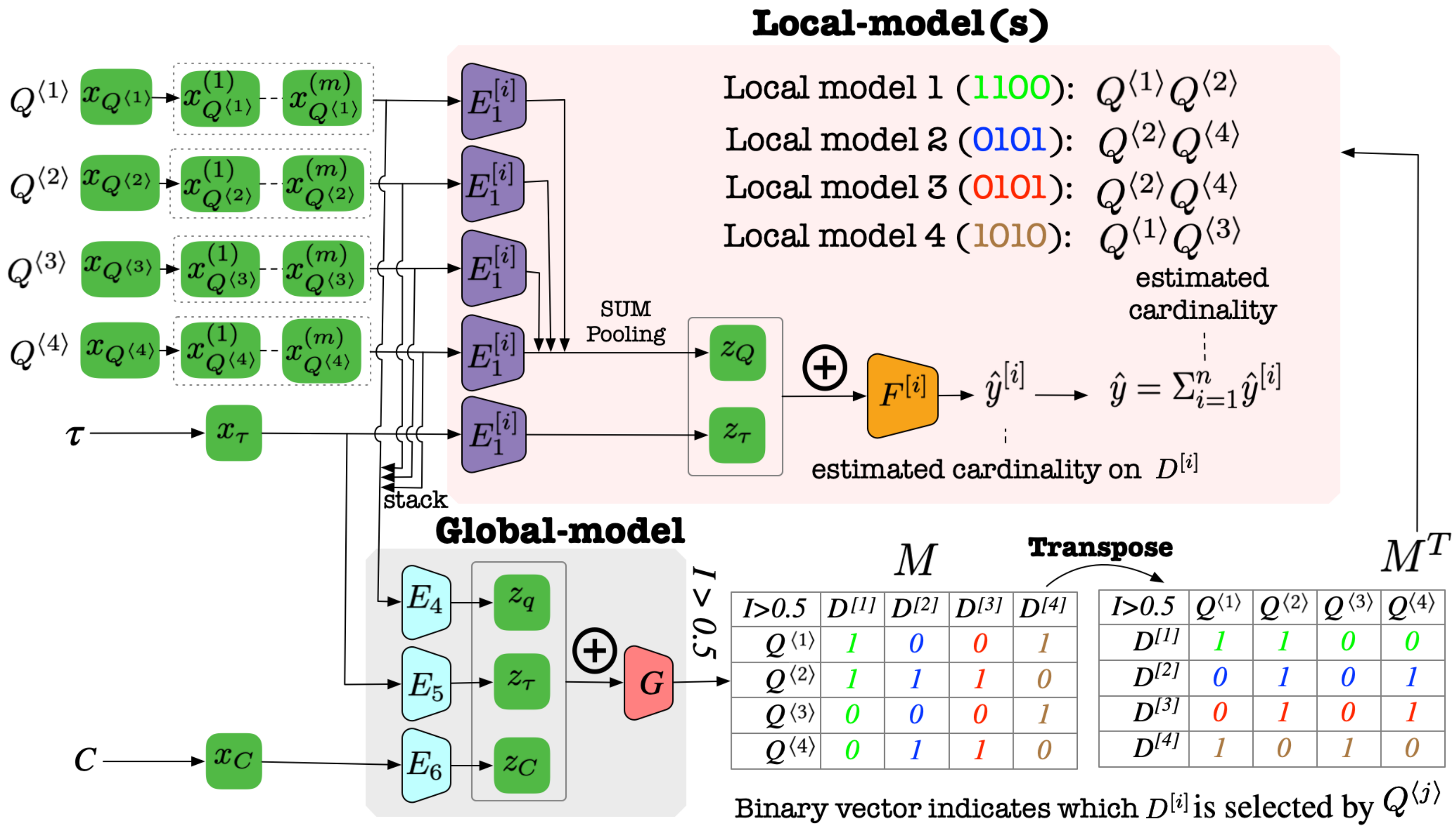


Cardinality Estimation for Similarity Joins

- What if the query is a set of vectors (Joins) ?
  - Q={q1,q2,q3,q4,…}

- A naïve way is to estimate for each vector and sum them up.
  - Card(Q) = Card(q1) + Card(q2) + Card(q3) + Card(q4)
  - Low efficiency

Cardinality Estimation for Similarity Joins



I>0.5	$D^{[1]}$	$D^{[2]}$	$D^{[3]}$	$D^{[4]}$
$Q^{\langle 1\rangle}$	1	0	0	1
$Q^{\langle 2\rangle}$	1	1	1	0
$Q^{\langle 3\rangle}$	0	0	0	1
$Q^{\langle 4\rangle}$	0	1	1	0

I>0.5	$Q^{\langle 1\rangle}$	$Q^{\langle 2\rangle}$	$Q^{\langle 3\rangle}$	$Q^{\langle 4\rangle}$
$D^{[1]}$	1	1	0	0
$D^{[2]}$	0	1	0	1
$D^{[3]}$	0	1	0	1
$D^{[4]}$	1	0	1	0

Experiments

- Datasets

Dataset	Dimension	#Data	#Training	#Testing	Metric	τ_{max}
BMS	512	515,597	8,000	2,000	Jaccard	0.50
GloVe300	300	1,917,494	8,000	2,000	Angular	0.60
ImageNET	64	1,431,167	8,000	2,000	Hamming	0.90
Aminer	2,943	1,712,433	4,000	1,000	Edit	0.05
YouTube	1,770	346,194	2,400	600	Euclidean	0.15
DBLP	5,373	1,000,000	2,400	600	Edit	0.20

Experiments

- Methods

id	Method	Embed	Auto-tuning	Framework	Opt	Data Segment
1	QES	CNN	No	Local	Select	No
2	Local+	CNN	Yes	Local	Select	Yes
3	GL-MLP	MLP	No	Global-Local	Select	Yes
4	GL-CNN	CNN	No	Global-Local	Select	Yes
5	GL+	CNN	Yes	Global-Local	Select	Yes
6	CardNet	VAE	No	Local	Select	No
7	Sampling	-	No	-	Select	No
8	Kernel-based	-	No	-	Select	No
9	MLP	MLP	No	Local	Select	No
10	SimSelect	-	-	-	Select	-
11	CNNJoin	CNN	No	Local	Join	No
12	GLJoin	MLP	No	Global-Local	Join	Yes
13	GLJoin+	CNN	Yes	Global-Local	Join	Yes

Experiments

- Query
  - Vectors: 80% training, 20% testing
  - Threshold: selectivity lower than 1%
  - Join Size: [1-100) training, [50-100), [100-150), [150,200) testing
- Environment
  - Intel(R) Xeon(R) CPU E5-2630v4@2.20GHz
  - 128 Gigabytes memory
  - PyTorch 1.0.1

Experiments (Accuracy)

- Cardinality Estimation for Similarity Search

Dataset	Method	Mean	Median	90th	95th	99th	Max
BMS	GL+	2.34	1.09	2.47	4.32	19.7	111
	Local+	2.37	1.05	2.51	4.36	18.4	98.3
	Sampling (10%)	5.18	1.83	11.2	17.4	55.0	165
	GL-CNN	3.50	2.42	8.21	10.6	15.7	291
	GL-MLP	4.41	3.02	9.78	12.8	19.7	439
	QES	7.27	5.05	16.5	21.6	32.2	644
	CardNet	12.4	5.16	31.3	48.8	99.1	335
	MLP	11.2	8.03	36.8	47.7	71.0	700
	Kernel-based	12.8	8.81	29.7	39.2	59.5	135
	Sampling (equal)	12.3	7.0	31.0	41.0	74.0	111
	Sampling (1%)	19.6	13.0	55.0	66.9	74.0	200

Dataset	Method	Mean	Median	90th	95th	99th	Max
Aminer	GL+	1.54	1.07	2.05	2.98	7.79	152
	Local+	1.61	1.12	2.36	3.01	6.46	321
	Sampling (10%)	2.41	1.72	3.90	5.26	14.2	31.0
	GL-CNN	1.83	1.27	4.21	5.39	8.38	154
	GL-MLP	3.09	2.14	7.10	9.18	14.2	290
	QES	5.22	3.63	11.9	15.4	24.4	541
	CardNet	5.45	2.05	7.59	12.9	43.1	3526
	MLP	8.39	5.80	19.4	25.1	38.6	780
	Kernel-based	9.85	6.91	22.6	28.7	44.6	117
	Sampling (equal)	66.5	42.0	182	245	245	245
	Sampling (1%)	19.5	4.20	56.0	75.0	136	245

Experiments (Accuracy)

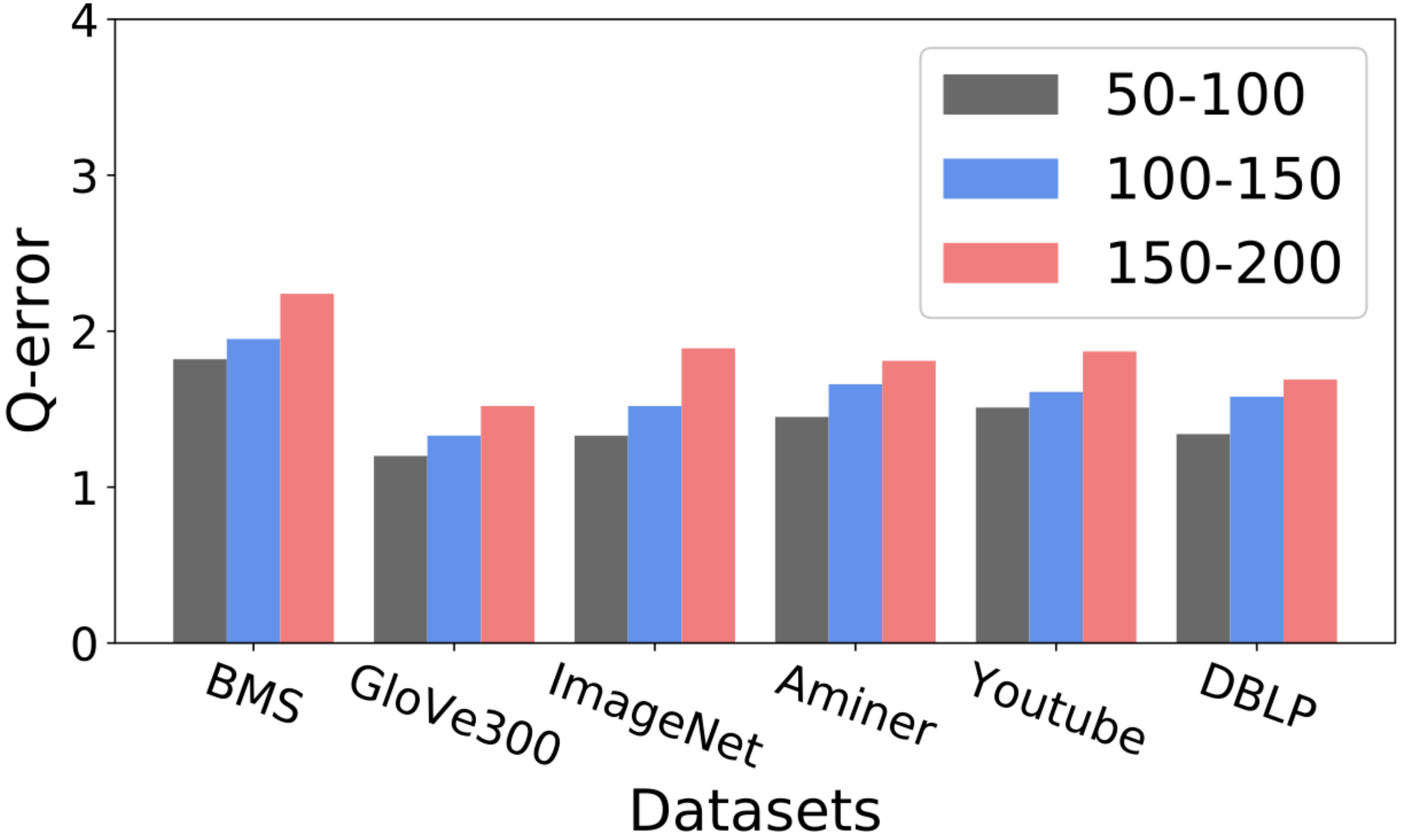
- Cardinality Estimation for Similarity Join

Dataset	Method	Mean	Median	90th	95th	99th	Max
BMS	GLJoin+	**1.87**	**1.31**	**4.31**	**5.51**	**8.55**	174
	GL+	2.01	1.36	4.59	6.12	9.34	205
	Sampling (10%)	3.99	2.18	8.46	13.5	23.1	**37.0**
	GLJoin	2.51	1.72	5.78	7.56	11.5	265
	CNNJoin	5.63	3.90	12.9	16.9	26.2	508
	CardNet	8.35	5.88	19.1	25.2	37.2	857
	Sampling (equal)	19.3	2.50	15.2	40.9	302	451
	Sampling (1%)	144	3.86	451	800	1505	2701

Dataset	Method	Mean	Median	90th	95th	99th	Max
Aminer	GLJoin+	**1.42**	**1.08**	3.26	4.16	6.26	121
	GL+	1.70	1.18	3.95	5.10	7.94	171
	Sampling (10%)	2.06	1.90	**2.90**	**3.35**	**4.57**	**5.12**
	GLJoin	2.02	1.40	4.66	5.94	9.25	193
	CNNJoin	6.58	4.67	15.2	19.6	30.5	788
	CardNet	5.16	3.55	11.7	15.2	24.3	766
	Sampling (equal)	124	7.77	371	501	909	1221
	Sampling (1%)	5.96	1.94	3.98	5.21	86.2	151

Experiments (Accuracy)

- Cardinality Estimation for Similarity Join



Experiments (Efficiency)

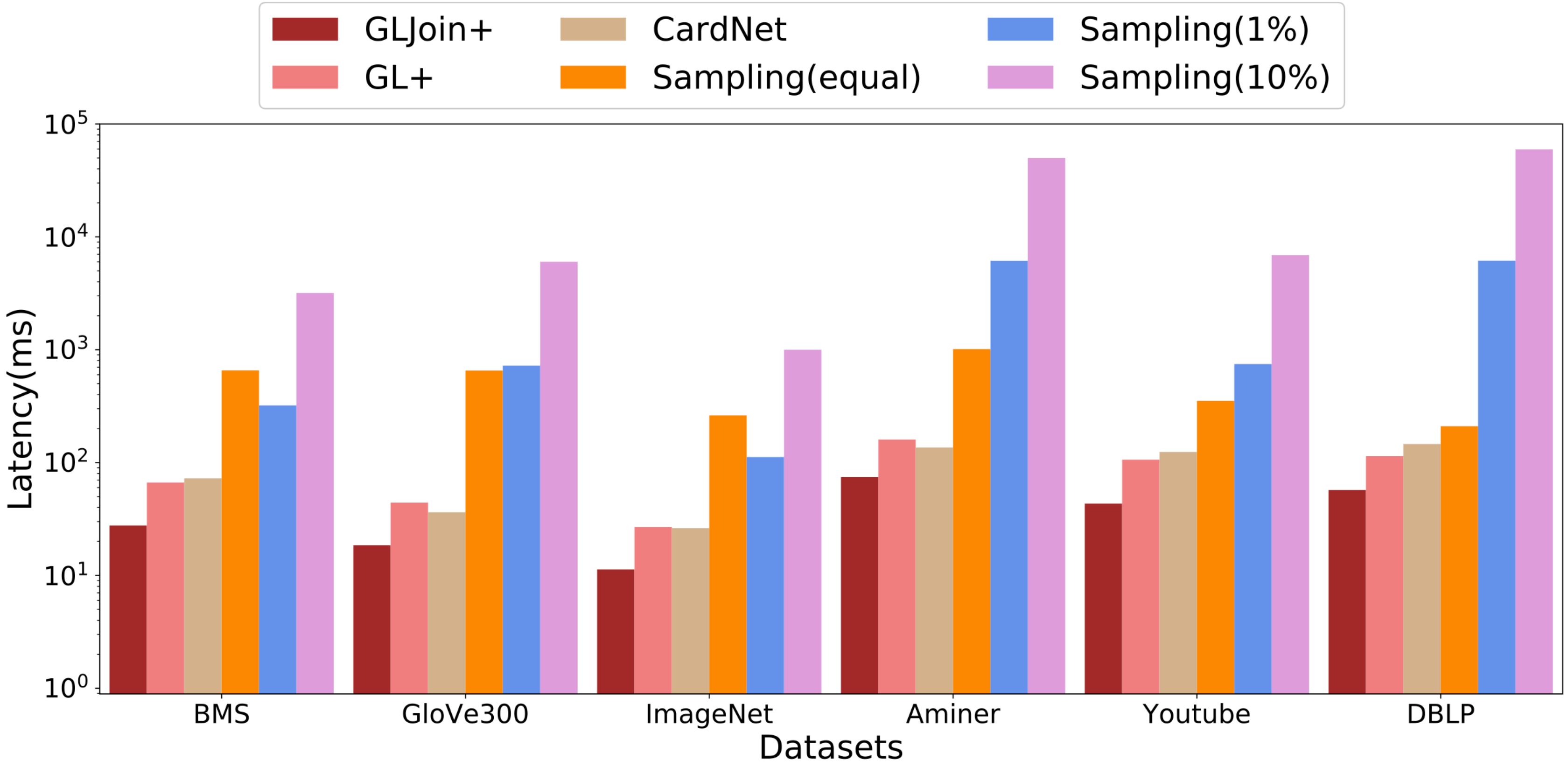
Model	BMS	GloVe300	ImageNET	Aminer	Youtube	DBLP
Sampling (1%)	12.7	27.7	3.66	243	24.5	239
MLP	4.11	3.09	3.21	9.01	8.23	15.3
QES	0.25	0.17	0.18	0.41	0.35	0.58
CardNet	38.8	35.3	16.2	54.5	52.8	55.1
GL-MLP	111	106	101	176	171	203
GL-CNN	29.2	21.3	7.32	35.6	32.1	55.6
GL+	28.3	22.1	7.51	34.2	30.7	50.1
GLJoin+	30.1	21.5	9.04	35.9	31.8	59.1

Model Size (MB)

Model	BMS	GloVe300	ImageNET	Aminer	Youtube	DBLP
SimSelect	3.96	12.1	5.22	5.87	12.5	18.6
Kernel-based	10.3	15.1	6.43	125	21.3	138
Sampling (10%)	30.9	70.1	10.5	587	69.5	598
Sampling (equal)	6.78	6.77	2.31	9.56	3.26	2.55
Sampling (1%)	3.21	7.23	1.12	61.4	7.46	61.5
CardNet	0.36	0.18	0.13	0.68	0.62	0.73
Local+	1.46	1.12	0.79	5.12	2.55	3.24
GL-MLP	0.51	0.65	0.28	3.43	2.35	3.69
GL-CNN	0.35	0.21	0.15	0.81	0.49	0.55
GL+	0.33	0.22	0.13	0.80	0.53	0.57
MLP	0.14	0.11	0.046	0.18	0.15	0.27
QES	0.015	0.012	0.007	0.042	0.021	0.032

Estimation Efficiency for Similarity Search (Milliseconds)

Experiments (Efficiency)



Estimation Efficiency for Similarity Join (Milliseconds)

Conclusion

sun-j16@mails.tsinghua.edu.cn

We make following contributions:

- We propose a basic neural network model for cardinality estimation of similarity queries.
- We propose Query segmentation and Data segmentation to improve performance of model.
- We extend model to support similarity join.
- We conduct Comprehensive experiments on real datasets.

We special thank to: