Unsupervised Hashing with **S**imilarity **D**istribution **C**alibration

Kam Woh Ng[1,2], Xiatian Zhu[1,3], Jiun Tian Hoe[4],
Chee Seng Chan[5], Tianyu Zhang[6], Yi-Zhe Song[1,2], Tao Xiang[1,2]

[1]CVSSP, University of Surrey [2]iFlyTek-Surrrey Joint Research Centre [3]Surrey Institute for People-Centred AI
[4]Nanyang Technological University [5]CISiP, Universiti Malaya [6]GAC R&D Centre

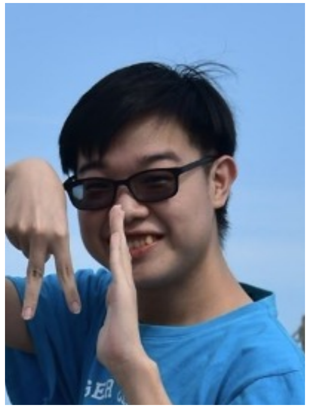
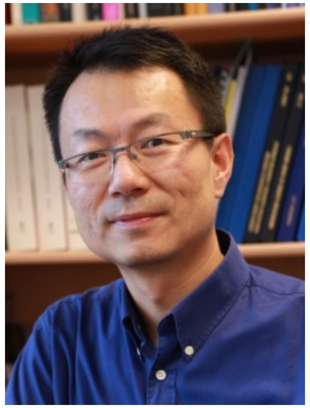
      

Image Retrieval

Image Retrieval

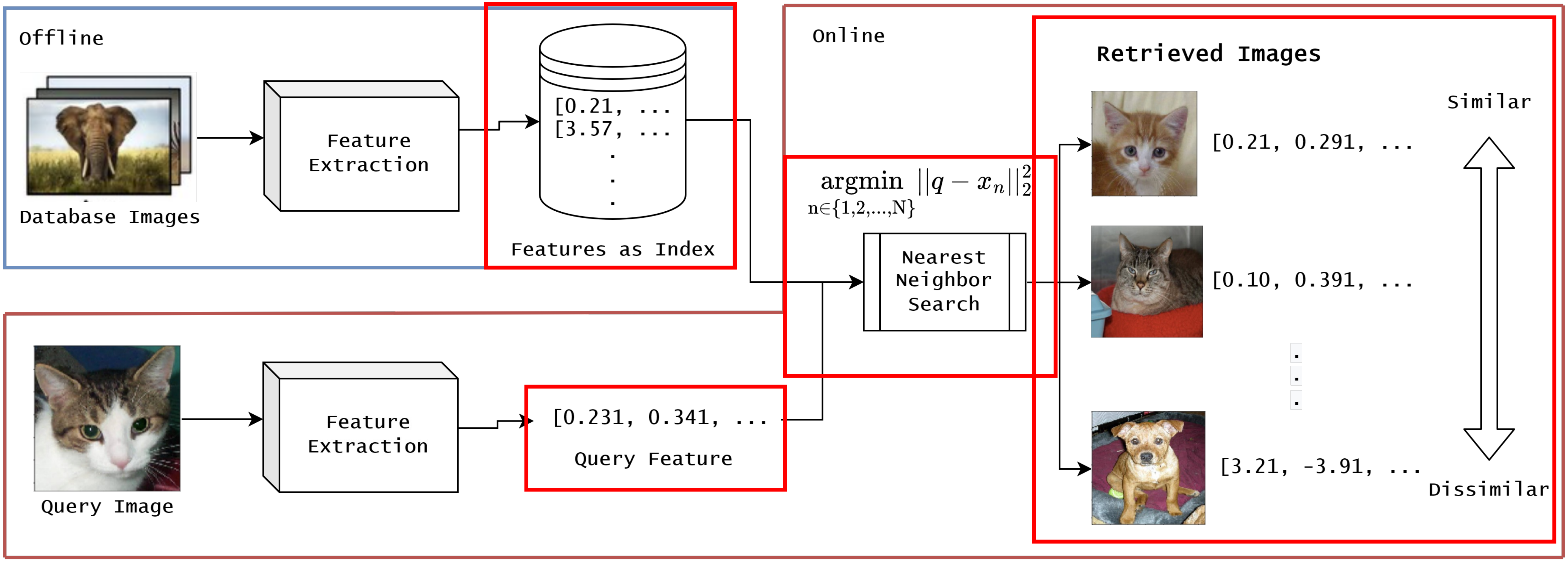
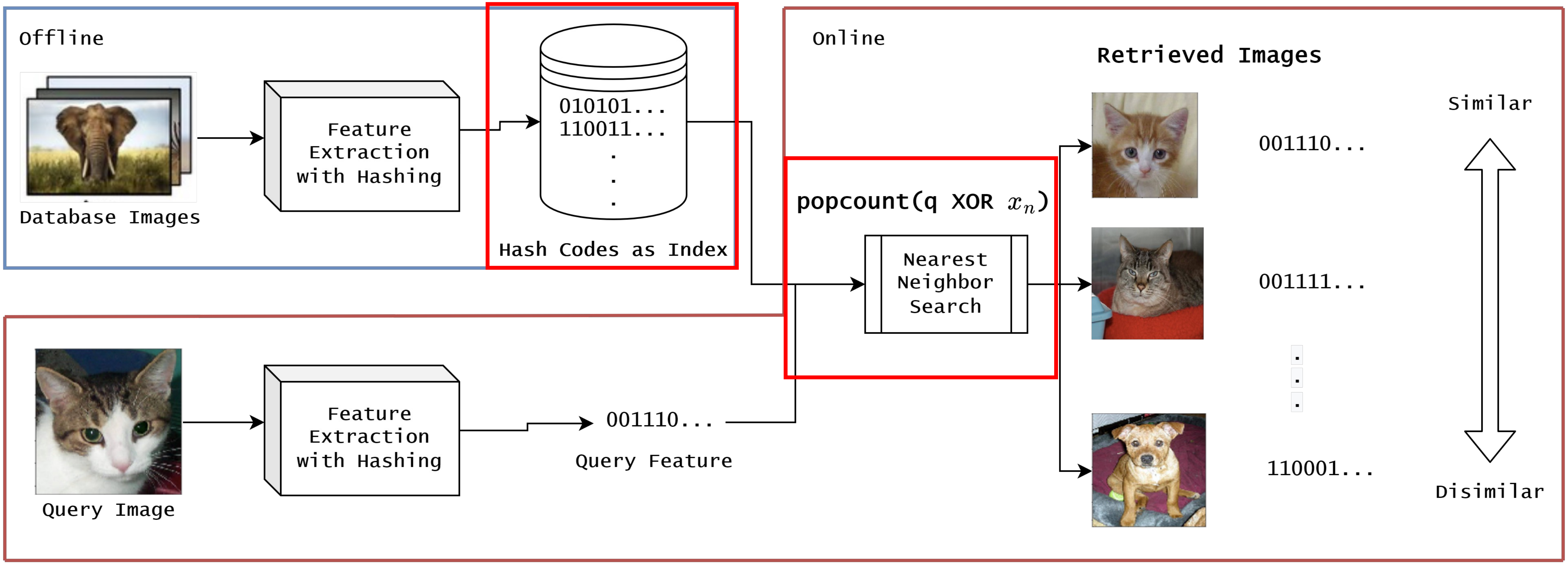


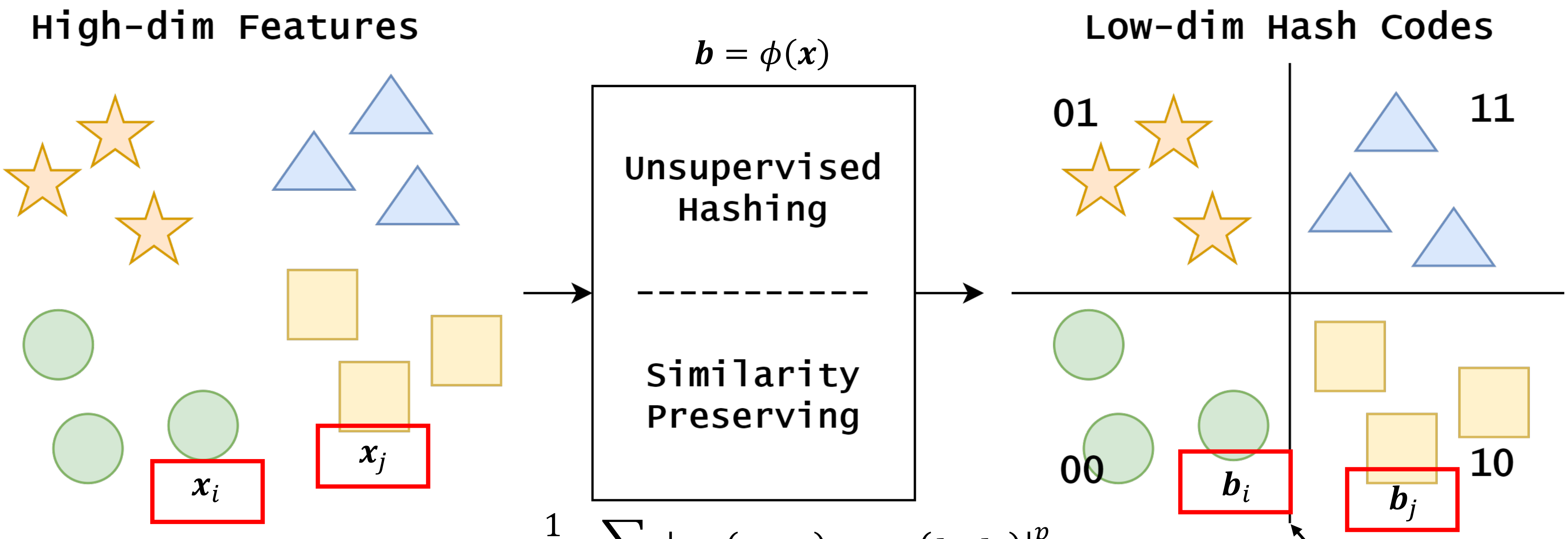
Image Retrieval

e.g., 2048 bytes (512-d vector) → 8 bytes (64-bits binary) per image → 256x smaller!!



Unsupervised Hashing

Unsupervised Hashing

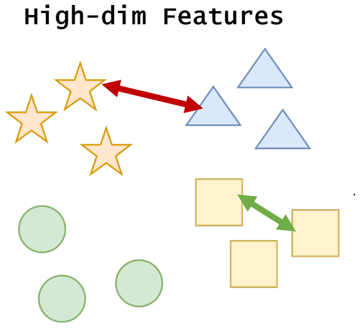


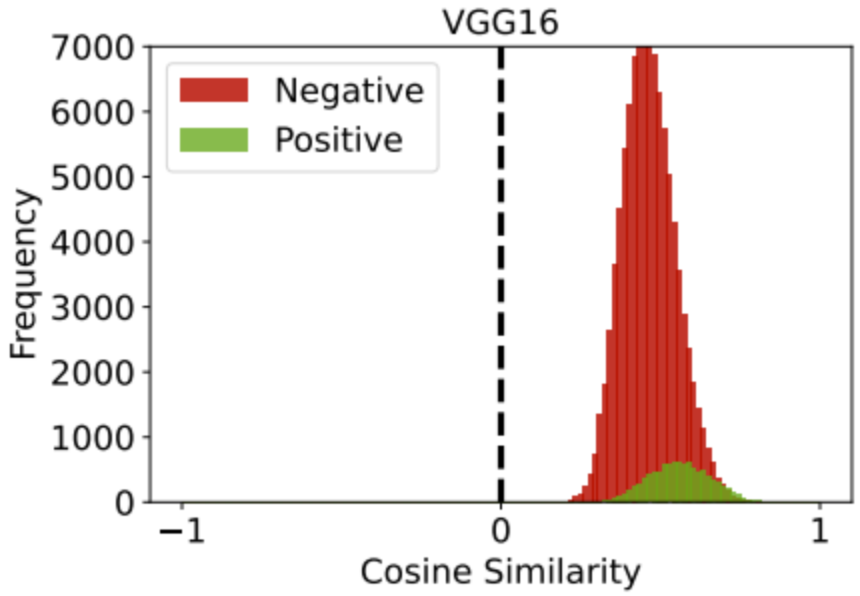
$$L = \frac{1}{|N|} \sum_{(i,j)\in N} \left| \cos(\boldsymbol{x}_i, \boldsymbol{x}_j) - \cos(\boldsymbol{b}_i, \boldsymbol{b}_j) \right|^p$$

objective: "reconstruct" the feature similarity

Hyperplane that "split" data into two region (a single bit)

Expected Outcome

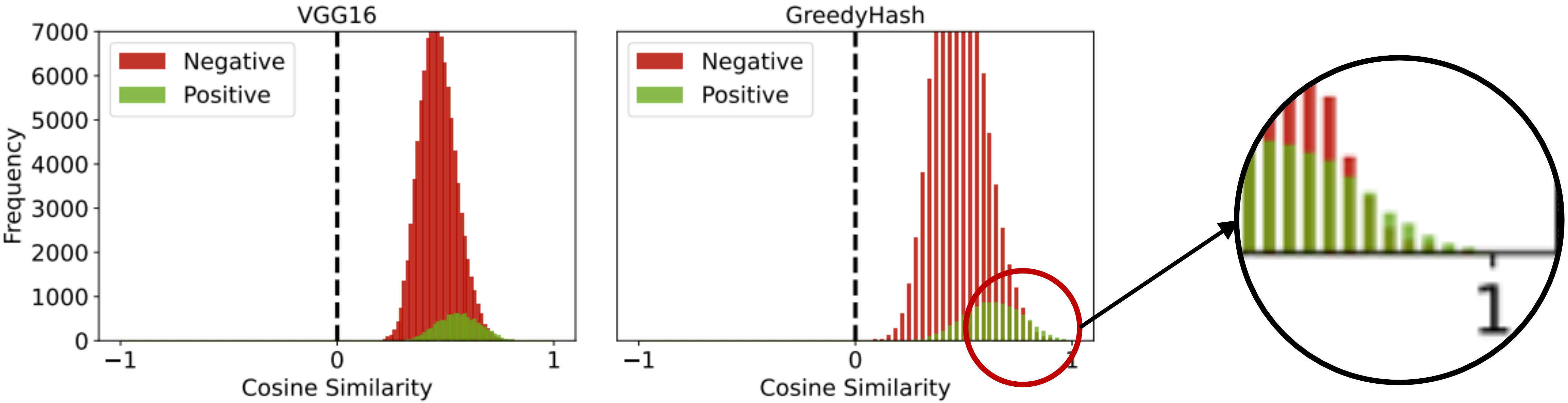




Similarity Distribution:
- We randomly sample many data pairs and compute their similarities, then plot them as histogram.

After hashing:
- Reproduce the similarity distribution!

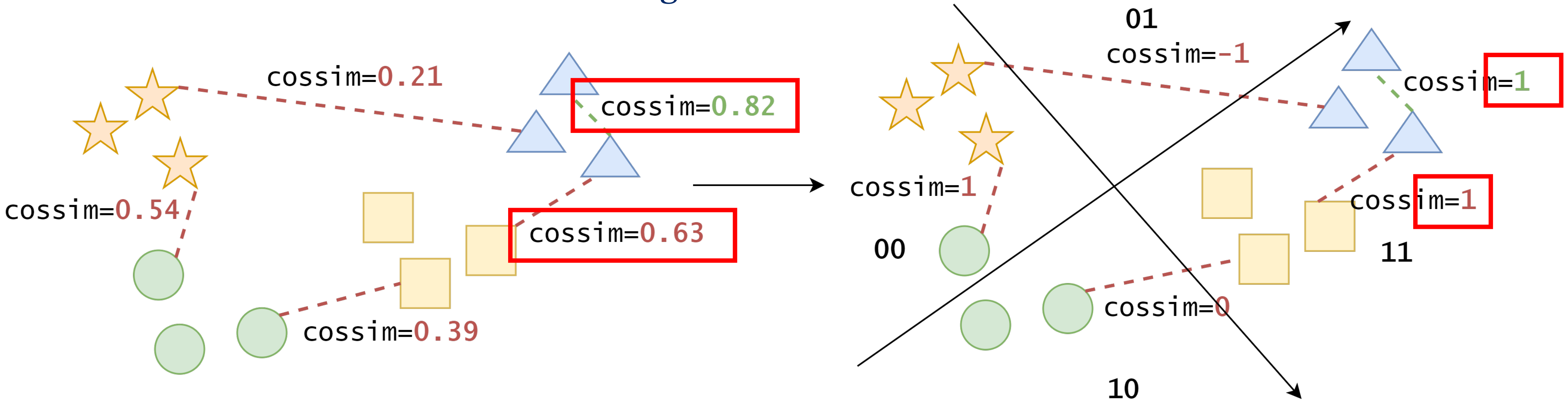
What is the problem?



The positive and negative pair have the same "similarity" --- similarity collapsed

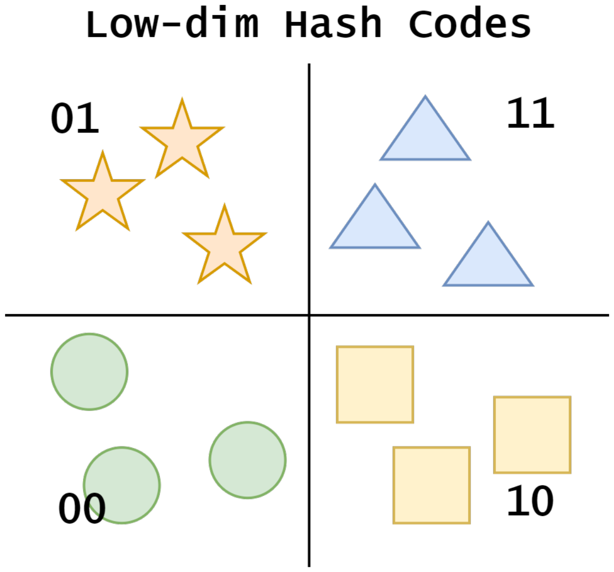
What is the problem?

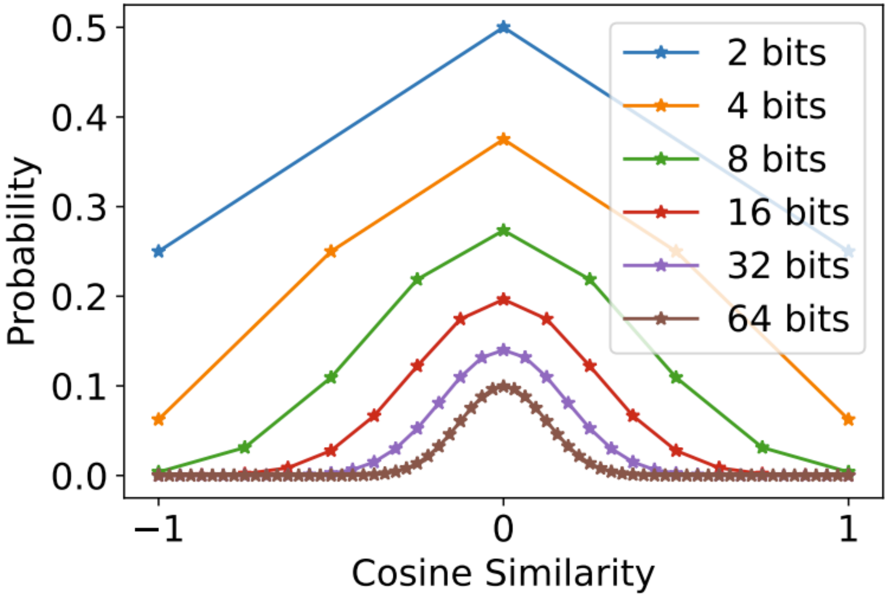
- **Similarity Collapsing** happens because the distance loss did not consider the ability of the hash codes representing similarity in the hash space.
- Low-dimensional K-bits hash codes:
  - Having discrete distances $L = \frac{1}{|N|} \sum_{(i,j) \in N} \left| \cos(x_i, x_j) - \cos(b_i, b_j) \right|^p$
  - The discrete distance is limited in range from 0 to K



What do we observe?

- Assuming the hash space are fully utilized, when we randomly sample two items from a database, what will be the similarity distribution?



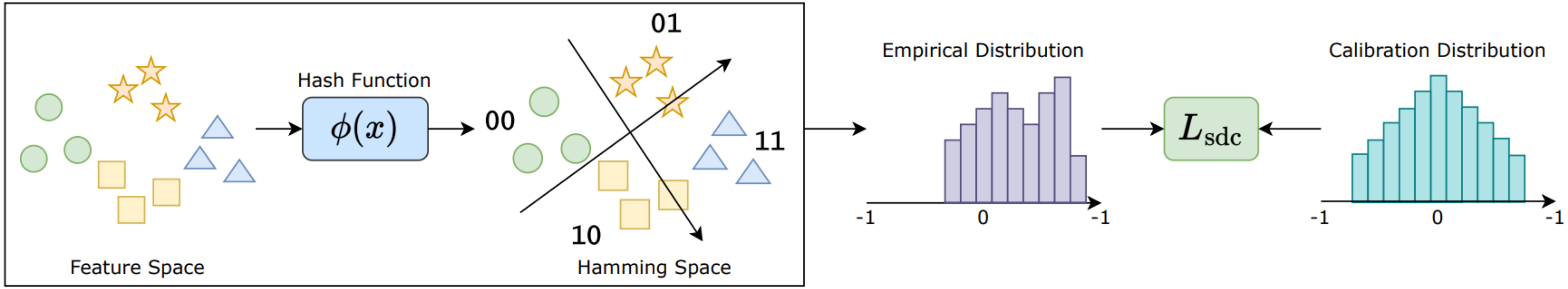


2 1 0

Hamming Distance

Similarity Distribution Calibration

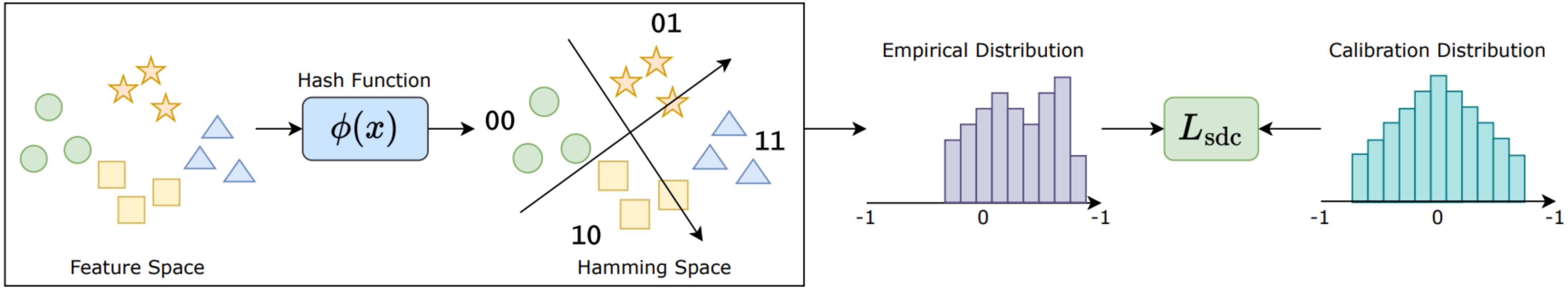
- Can we align the hash code similarity distribution with a calibration distribution (such as Beta distribution)?



- Empirical Distribution is the hash code similarity distribution (constructed with feature similarity to achieve similarity preserving)
- Calibration Distribution is the target distribution (e.g., Beta distribution)

Similarity Distribution Calibration

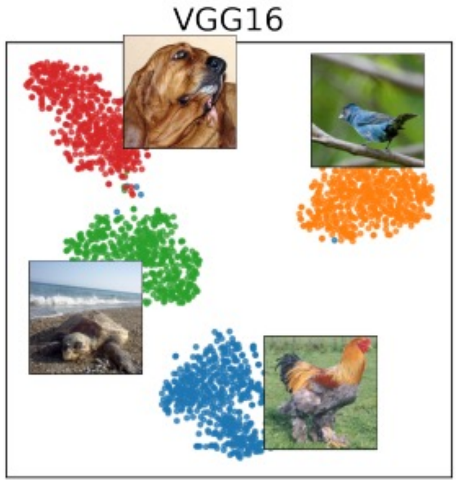
- Can we align the hash code similarity distribution with a calibration distribution (such as Beta distribution)?
- **Proposal: To minimize the Wasserstein distance.**



$$L_{Wasserstein} = \int_0^1 |F^{-1}(z) - C^{-1}(z)|dz \longrightarrow L_{sdc} = \frac{1}{|N|} \sum_{(i,j)\in \textbf{\textit{sorted}}(N)} \left|\cos(b_i, b_j) - \mathrm{C}^{-1}\left(\frac{2i-1}{2|N|}\right)\right|$$

Show me the result!!

Visualization of toy example



Toy example: Learning a 2-bits hash function for 4 well separable object classes

Observation:

1. Simply preserving the distance collapses the similarity scores completely across all the classes (GreedyHash)
2. Adding code balance layer (BiHalf) reduces the degree of collapsing to two groups.
3. Through aligning the similarity distribution with a calibration distribution, our SDC solves this collapsing problem well

Retrieval example

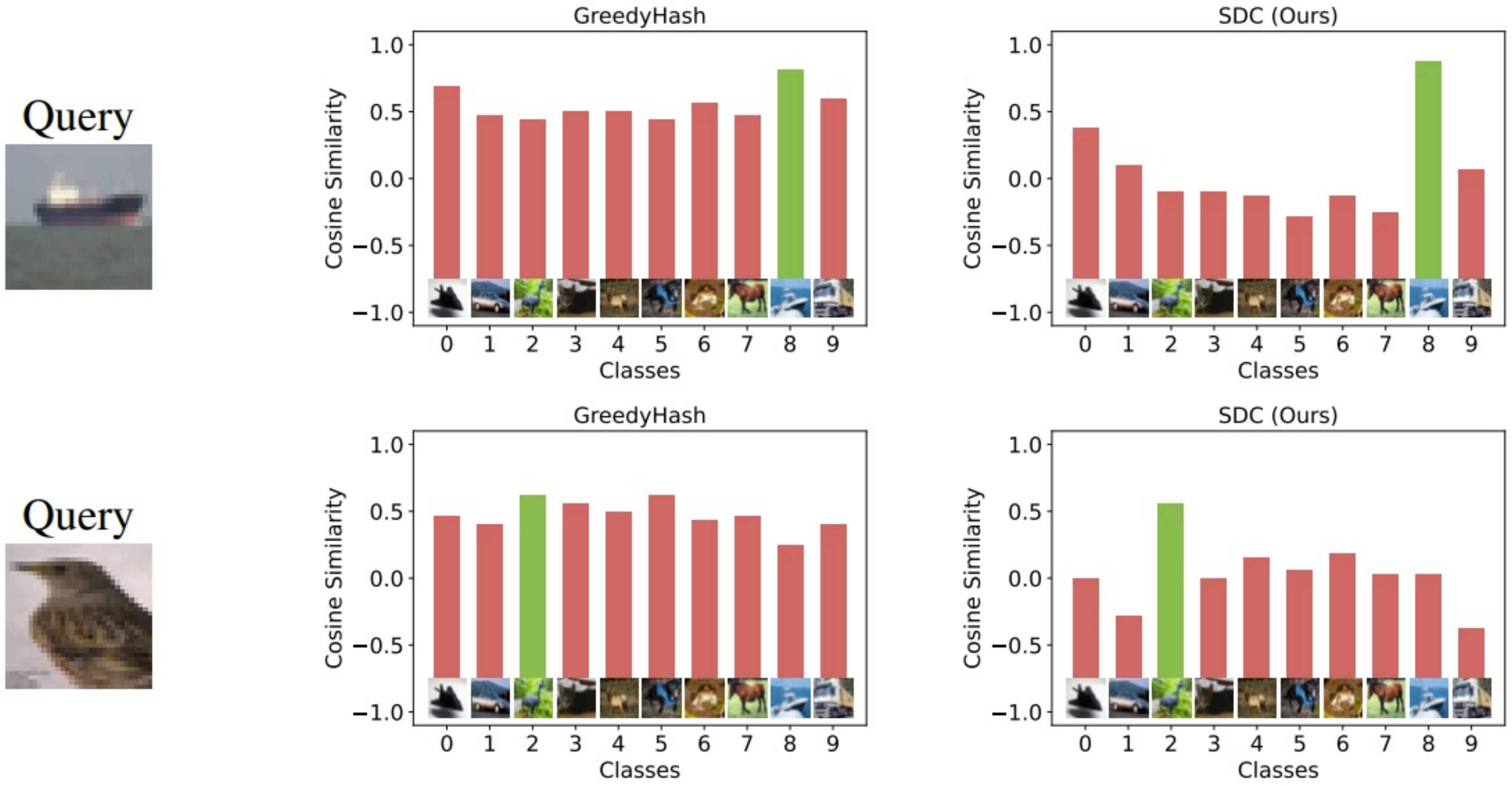


Figure 8: Two qualitative object image retrieval examples on CIFAR10. Green: Positive class; Red: Negative class.

Experimental result

Methods	Reference	CIFAR10			ImageNet100			NUSWIDE			MS-COCO		
		16	32	64	16	32	64	16	32	64	16	32	64
VGG16													
LsH [22]	STOC'98	23.9	29.6	37.6	14.7	29.7	48.7	51.0	59.3	67.1	45.2	51.6	59.8
SH [59]	NeurIPS'08	41.8	42.1	43.5	35.1	50.9	60.9	63.0	60.9	64.0	59.4	64.8	66.2
ITQ [16]	TPAMI'12	46.8	51.3	54.4	45.5	62.1	72.7	73.2	75.0	77.1	67.6	72.9	75.4
SSDH [51]	IJCAI'18	41.0	39.6	38.5	32.3	40.1	44.6	66.8	67.8	66.7	53.9	56.7	57.4
GreedyHash [57]	NeurIPS'18	44.9	51.9	55.7	54.4	68.7	74.7	70.0	76.2	79.3	66.8	73.2	77.4
TBH [54]	CVPR'20	48.2	50.2	50.7	42.9	44.5	48.3	75.8	77.8	78.5	68.8	72.6	74.8
CIBHash† [48]	IJCAI'21	56.2	59.2	61.2	63.9	71.4	74.6	77.1	79.7	80.9	73.3	77.0	78.5
BiHalf [55]	AAAI'21	54.7	58.1	60.6	60.7	71.2	76.0	77.4	80.1	81.9	71.2	75.6	78.0
SDC†	**Ours**	**59.8**	**64.0**	**66.3**	**72.8**	**78.5**	**80.6**	**80.7**	**82.3**	**83.4**	**76.9**	**79.8**	**81.2**
ResNet50													
NSH† [56]	IJCAI'22	70.6*	73.3*	75.6*	-	-	-	75.8*	81.1*	82.4*	74.6*	77.4*	78.3*
SDC†	**Ours**	**74.2**	**75.8**	**78.4**	**80.7**	**83.8**	**85.7**	**81.2**	**83.2**	**84.2**	**78.3**	**81.1**	**82.6**
ViT-B/16													
WCH† [53]	ACCV'22	77.5	79.3	80.6	69.4	76.9	80.8	70.7	75.6	78.6	73.0	78.8	81.4
SDC†	**Ours**	**87.4**	**88.4**	**89.0**	**76.4**	**82.6**	**84.9**	**81.8**	**83.3**	**84.0**	**79.2**	**83.3**	**84.5**

Table 1: Unsupervised hashing results. *: Originally reported. †: Using contrastive learning.

Conclusion

- We analyzed a severe problem in unsupervised hashing, namely Similarity Collapsing in Hash Space
  - Hash space does not have enough "similarity range" during comparison, causing positive and negative pair to have the same similarity value

- We propose Similarity Distribution Calibration (SDC)
  - Aligning the hash code similarity distribution towards a calibration distribution
  - This distribution has sufficient spread across the similarity range.

Thank you!

Github: https://github.com/kamwoh/sdc