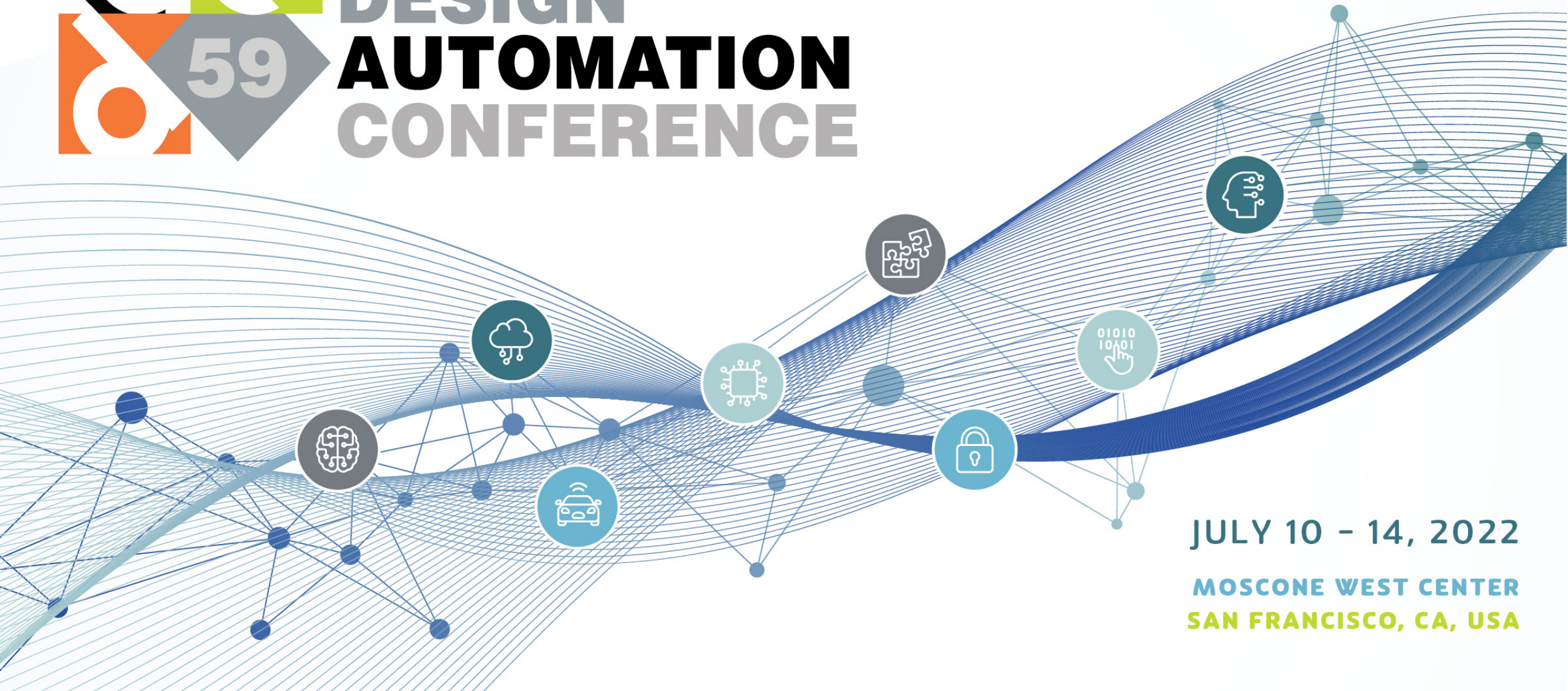
59 DESIGN AUTOMATION CONFERENCE



JULY 10 - 14, 2022

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

PIM-DH: ReRAM-based Processing-in-Memory Architecture for Deep Hashing Acceleration

Fangxin Liu (Speaker)

Wenbo Zhao, Yongbiao Chen, Zongwu Wang, Zhezhi He, Rui Yang, Cheng Zhuo, and **Li Jiang***

Shanghai Jiao Tong University

先进计算机体系结构实验室
Advanced Computer Architecture Laboratory

Outline

- Background and motivation
- Proposal: ReRAM-based Processing-in-Memory Architecture for Deep Hashing Acceleration
- Design and implementation details
- Experiment results
- Conclusion

Large-scale Image Search

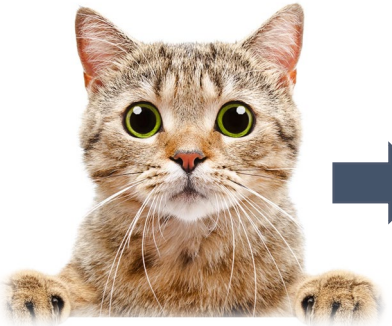
Recommendation systems

Search Engines

Finding visually similar images

Challenge in big data applications

- **Facebook**: more than 1 billion images/month
- **Taobao**: more than 28.6 billion images



Query Image

DataBase

Retrieved Image

Image Retrieval in General

Image retrieval is reduced to nearest neighbor search in high dimensional space

Nearest Neighbor (NN) Search:

- Searching: **Slow** retrieval efficiency
- Storage: **High** memory consumption

Challenges

Energy

Latency

Deep Hashing

Images are represented by binary codes

Hash(dolphin)

0011101

Hash(cat)

1100000

Hash(cat)

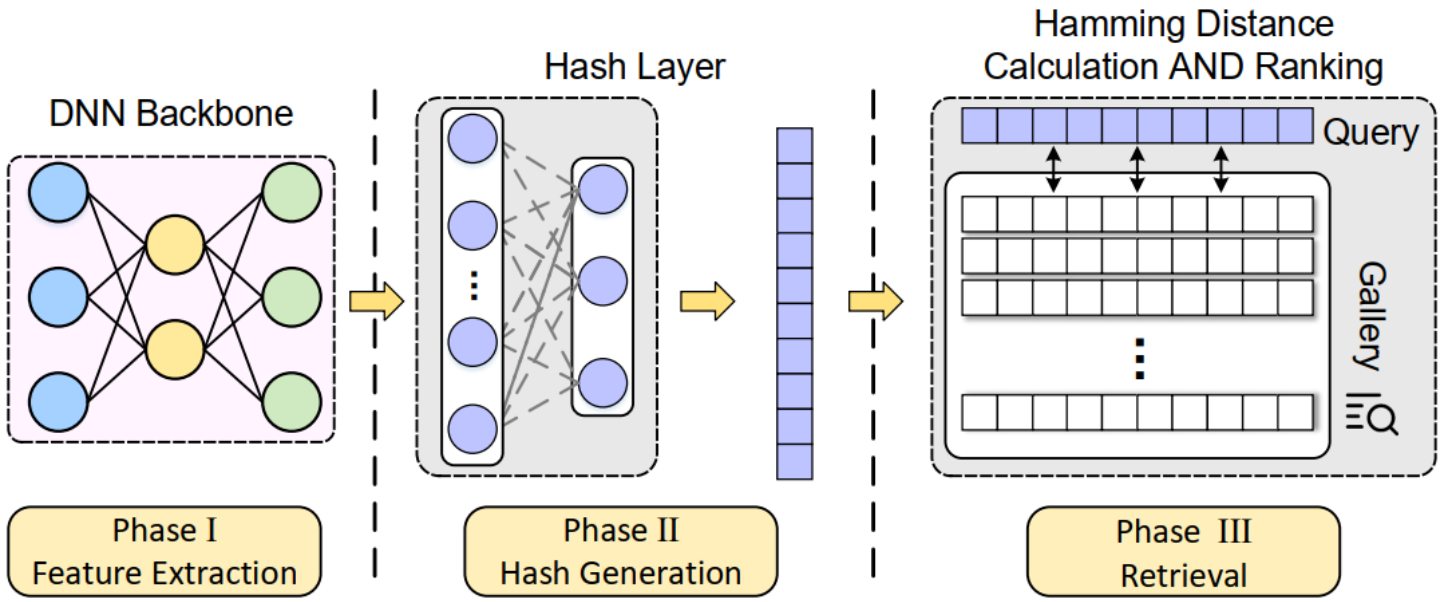
1100001

Dist ↑ Sim ↓

Dist ↓ Sim ↑

Fast search can be carried out via Hamming distance measurement. (XOR operation)

Deep Hashing

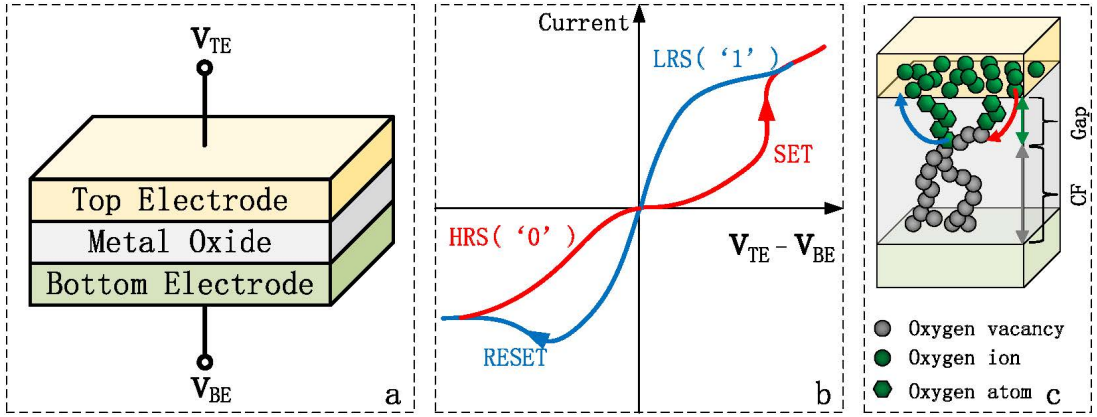


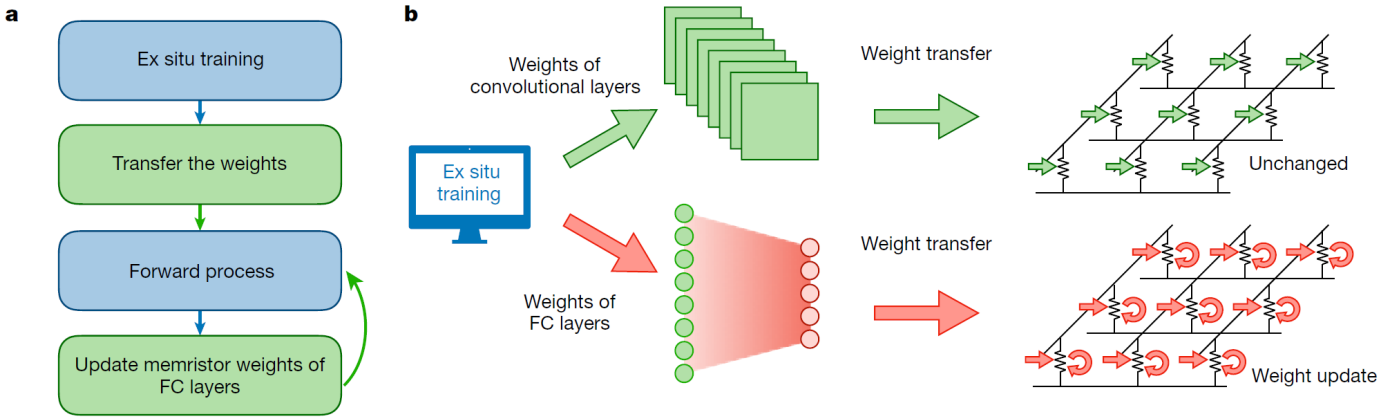
Benefits:

- High compression ratio (scalability)
- Fast similarity calculation with Hamming distance (efficiency)

- **Considerable computational resources:**
  - Feature extraction
  - Hamming distances calculation
- **Recommending platform in Taobao:**
  - requires hash computations on 600 billion entries.

RRAM-based Multiply-Accumulate Computation

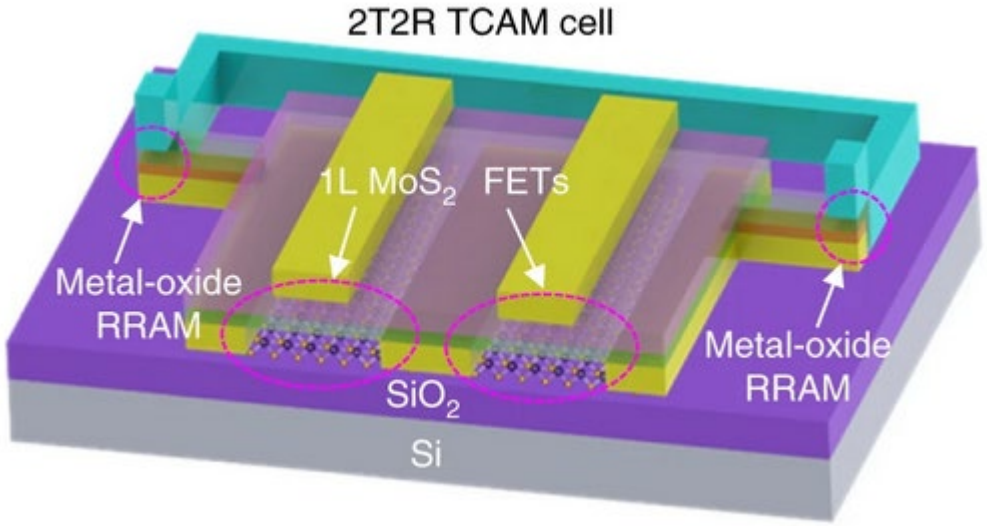


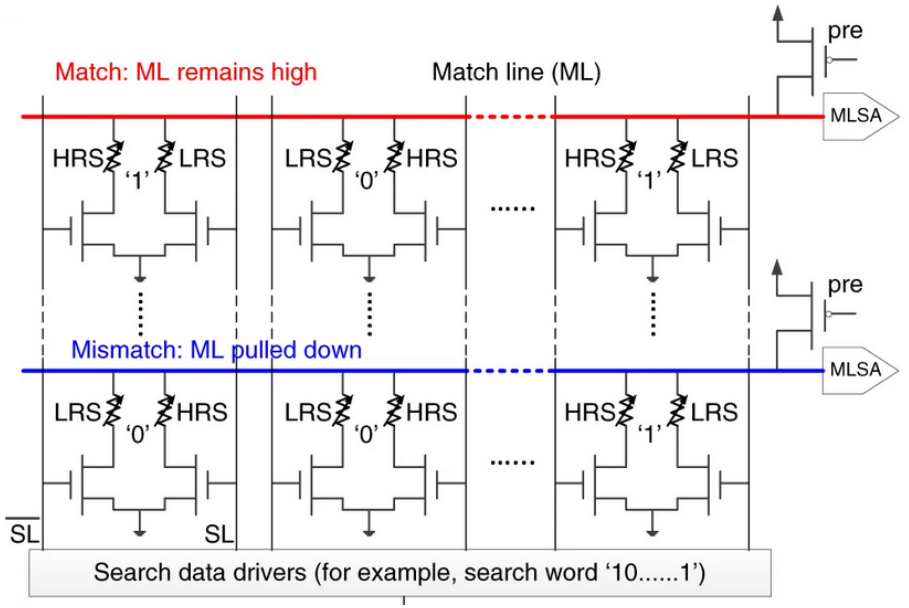


- Saves the weights on ReRAM to avoid massive data movement
- Execute GEMM by gathering the analog currents in vertical bit-lines, effectively reduce the computing complexity from $O(n^2)$ to $O(1)$.

In-situ analog MAC capabilities of the crossbar memory structures: an effective approach to the memory wall.

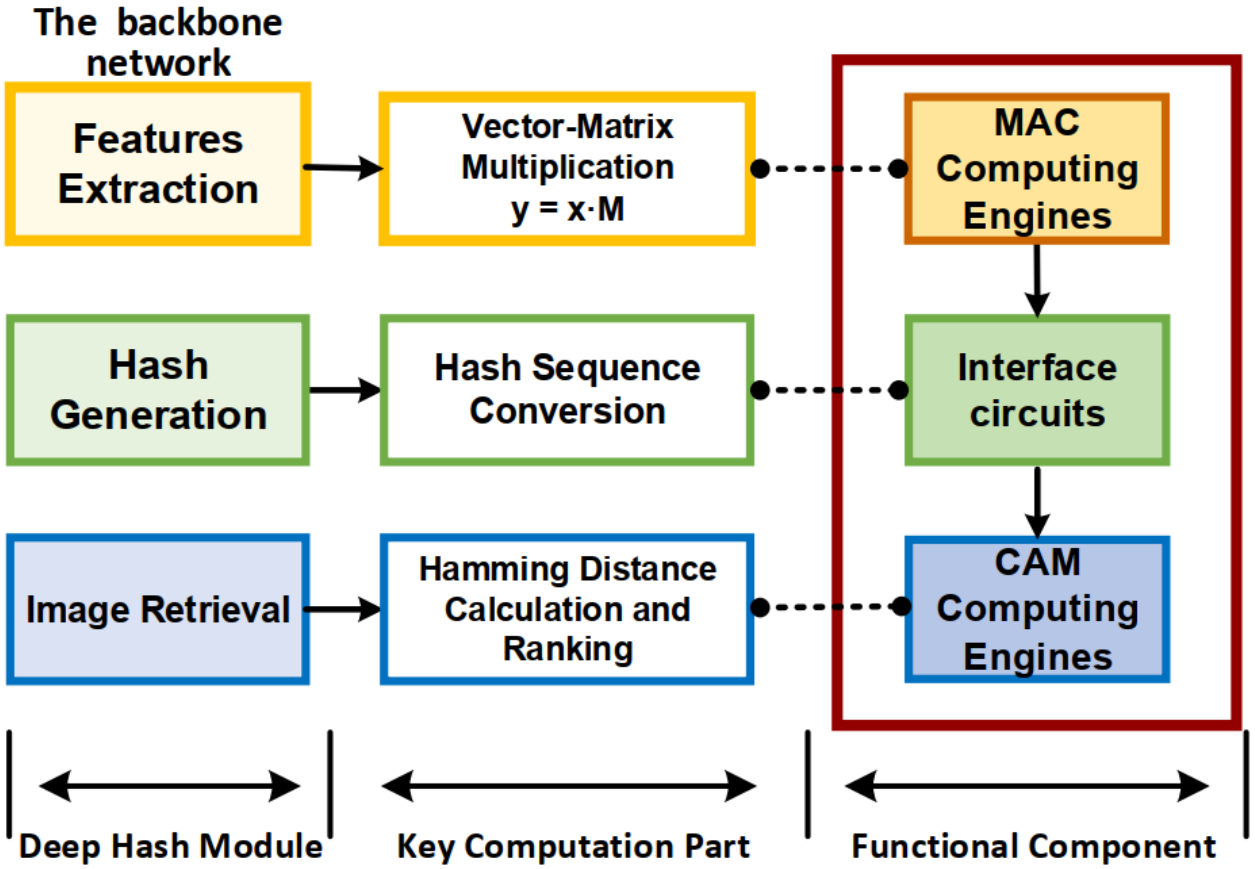
ReRAM-Based Content-Addressable Memory





- ReRAM-based TCAM (Ternary CAM) realizes bitwise XNOR-based search operations on each pair of cells by applying complementary bias voltages to the ReRAM devices
- TCAM is often used in hardware implementation of in-memory computing for parallel search of large datasets because of its high speed and energy-efficiency.

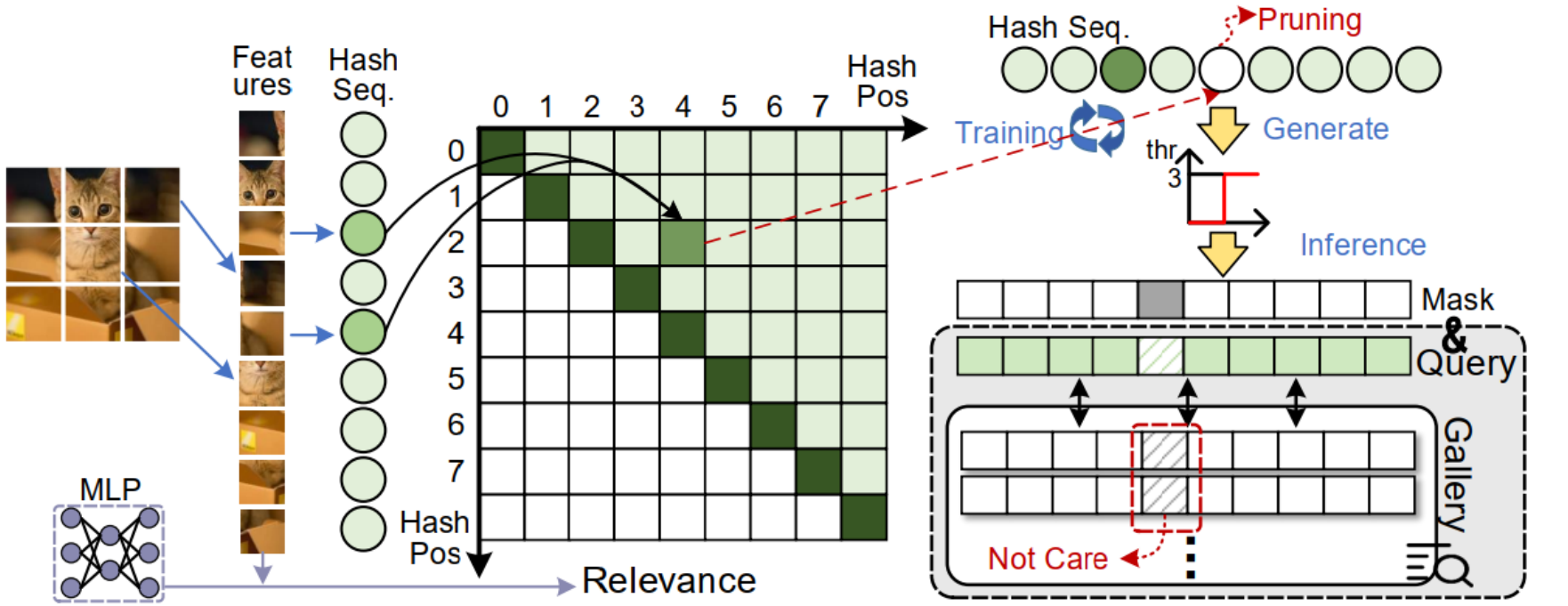
MOTIVATION AND KEY IDEA



Challenges

- **Massive number of searches**
  - the leakage current mechanism can check only whether two contents are equal or not
- **Extreme CAM overhead**
  - The gallery hash sequences stored in the ReRAM CAM are determined by the length of hash sequences.

Overview of Our PIM-DH Algorithm

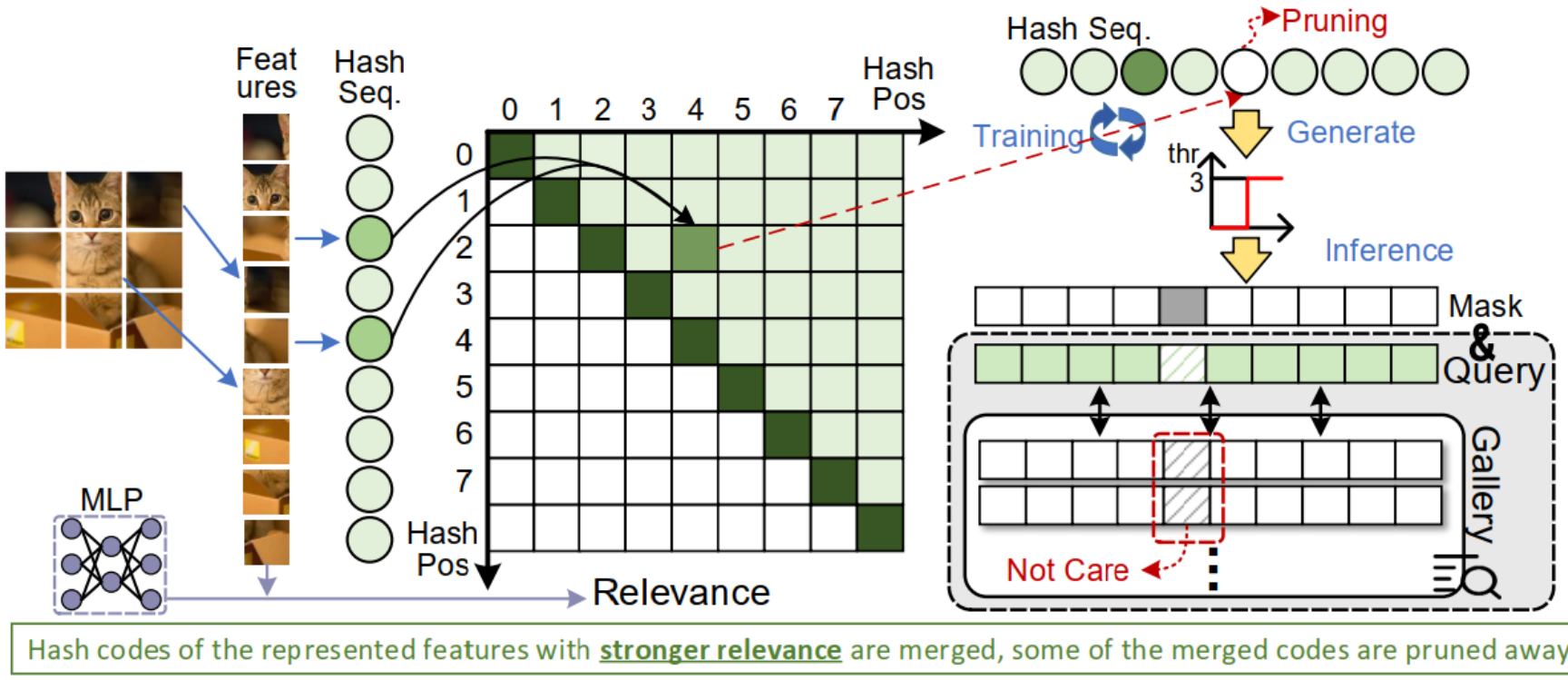


Hash codes of the represented features with **stronger relevance** are merged, some of the merged codes are pruned away.

Goal:

- represent the whole query hash sequence with fewer hash codes while guaranteeing the retrieval accuracy of images.

Overview of Our PIM-DH Algorithm



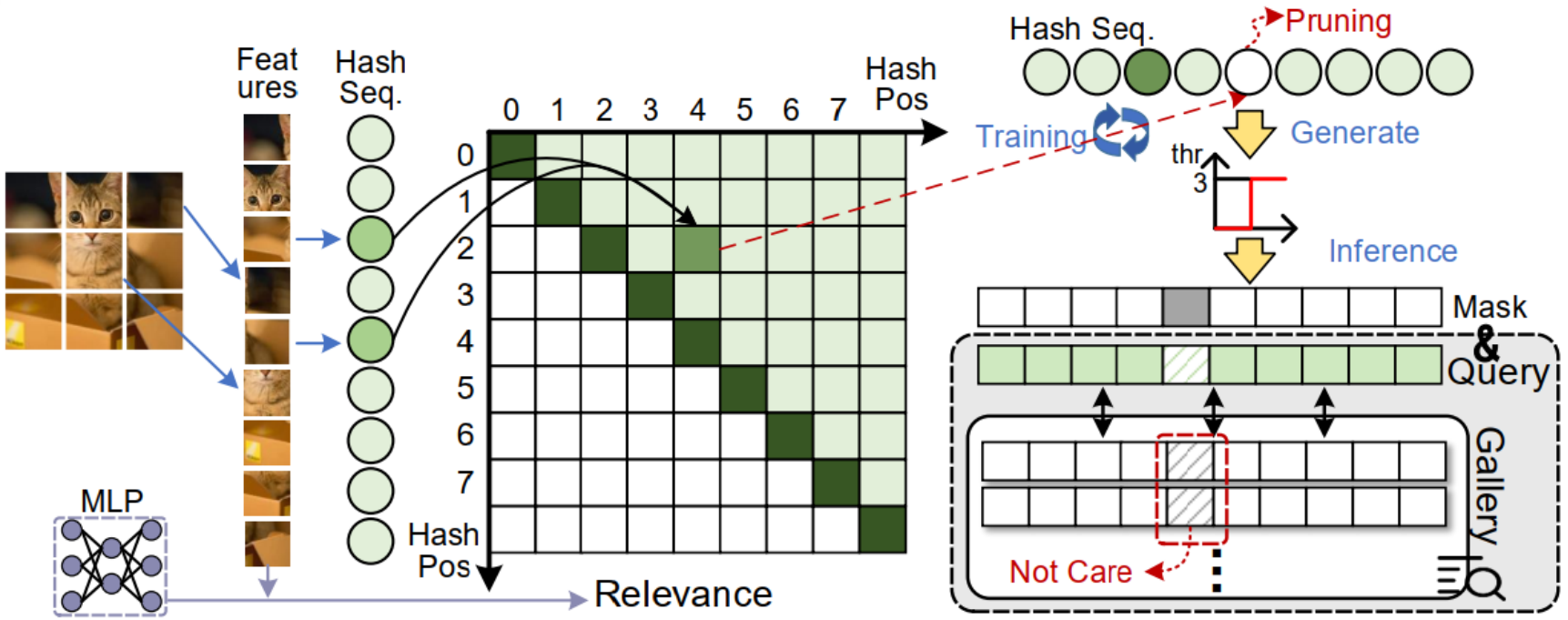
Hash codes of the represented features with **stronger relevance** are merged, some of the merged codes are pruned away.

Forward pass

- Step1: the relevance among the hash codes is made as the MLP output

We integrate the training process to evolve hash code sparsity by enforcing relevance-wise restrictions at every training iteration.

Overview of Our PIM-DH Algorithm



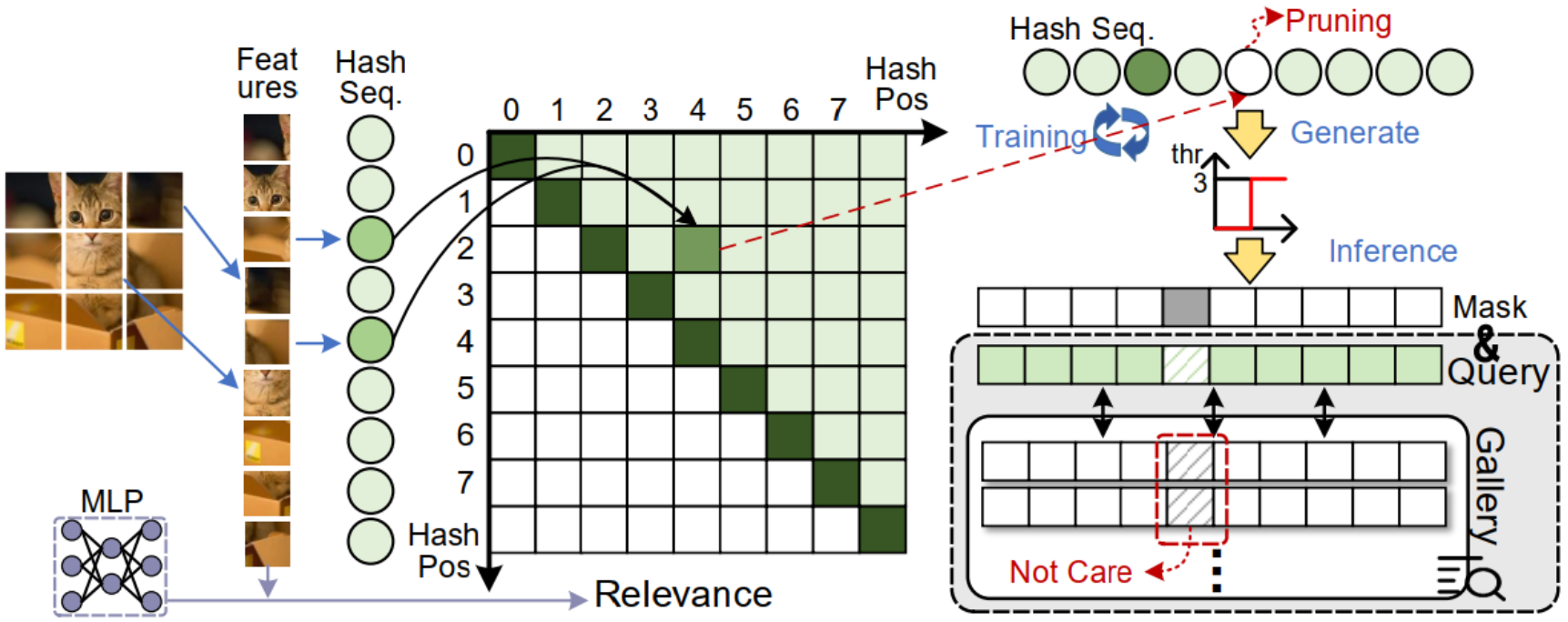
Hash codes of the represented features with **stronger relevance** are merged, some of the merged codes are pruned away.

Forward pass

- Step2: the hash sequence is sparsified based on the relevances

We integrate the training process to evolve hash code sparsity by enforcing relevance-wise restrictions at every training iteration.

Overview of Our PIM-DH Algorithm



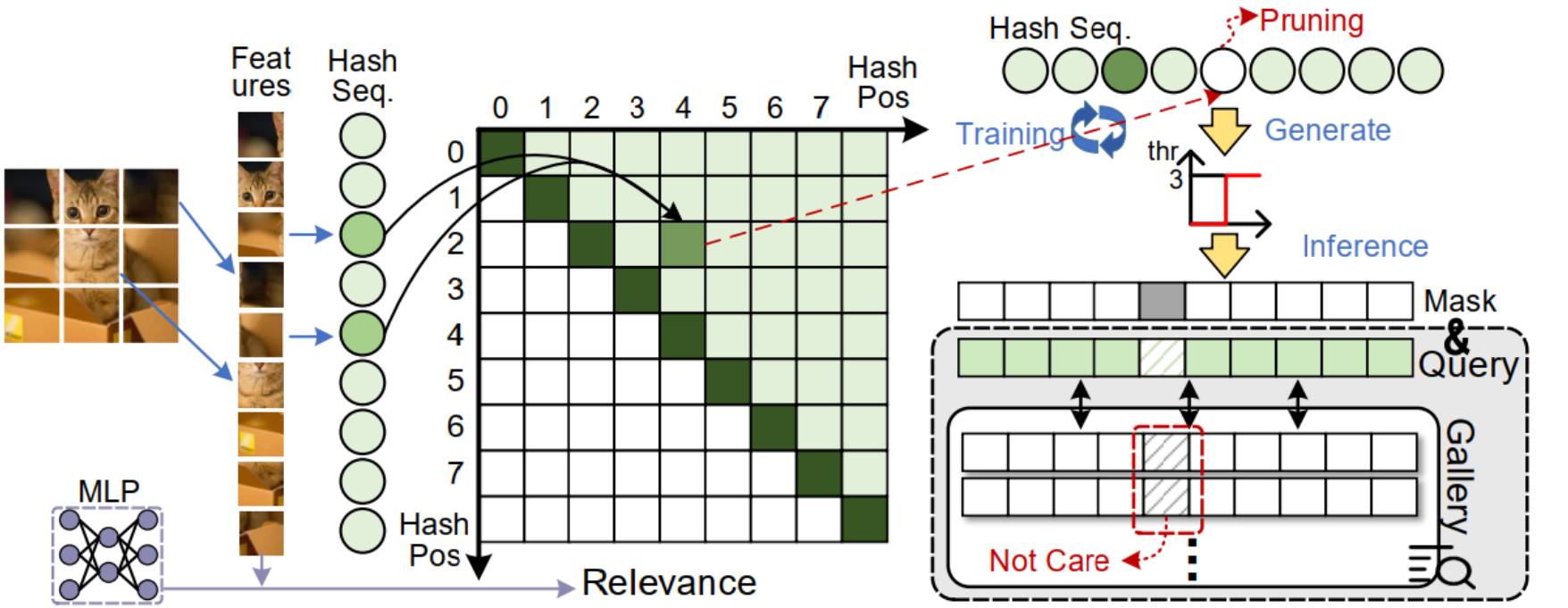
Hash codes of the represented features with **stronger relevance** are merged, some of the merged codes are pruned away.

Forward pass

- Step3: the hash computations are carried out with the sparse version of the hash sequence.

We integrate the training process to evolve hash code sparsity by enforcing relevance-wise restrictions at every training iteration.

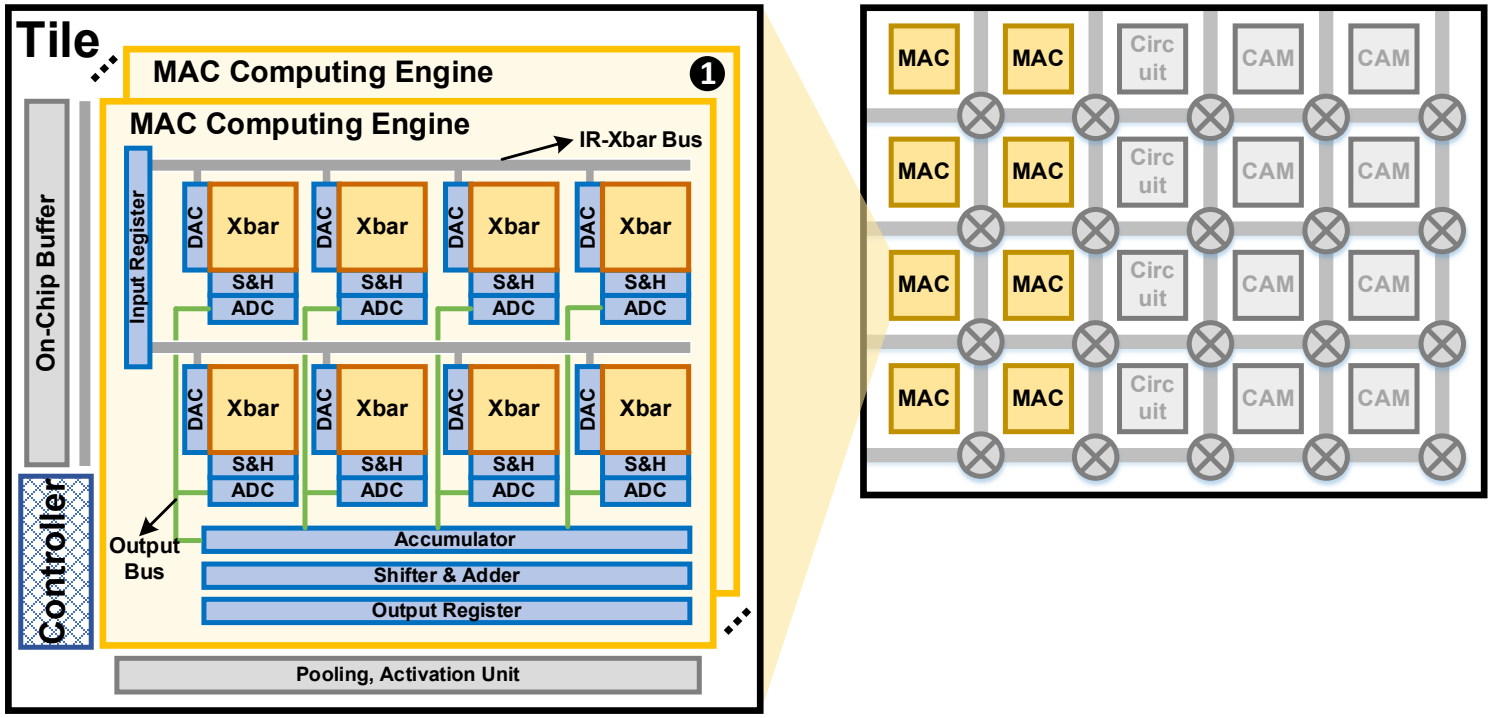
Overview of Our PIM-DH Algorithm



Hash codes of the represented features with **stronger relevance** are merged, some of the merged codes are pruned away.

- The r-percentile of relevance of features, which exceeds r*L of them, is recorded.
- The average value of features of all these r-percentiles is denoted as threshold, which can eliminate the outputs in the top r portion.

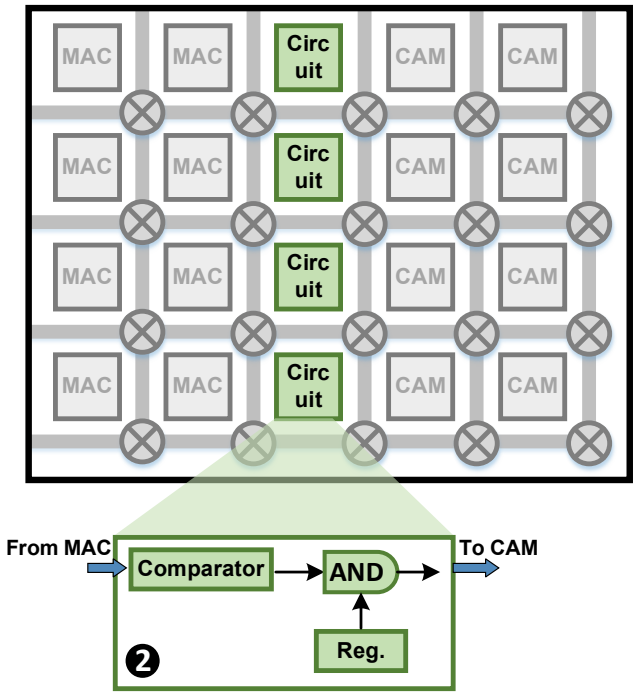
Overview of Our PIM-DH Architecture



Q3: How to support efficient Deep Hashing algorithm?

- **Vector-Matrix Multiplication**: can be efficiently completed by **MAC Compute Engines**, consisting of ReRAM crossbars ①.

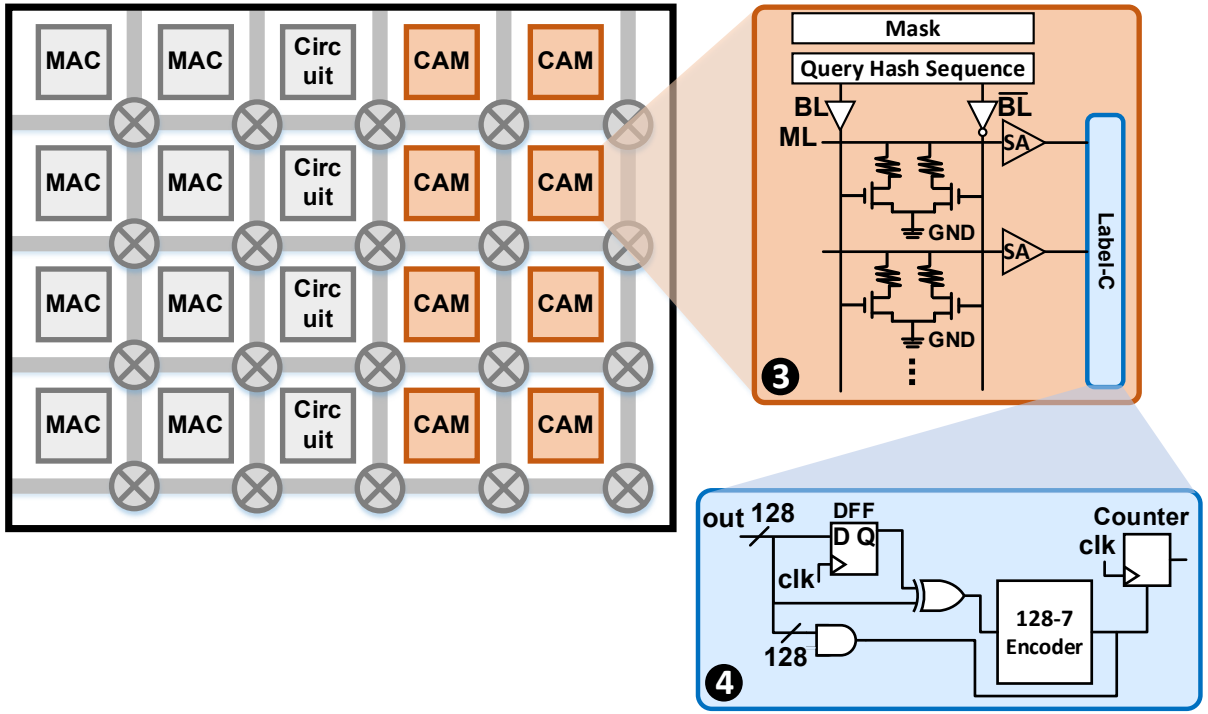
Overview of Our PIM-DH Architecture



Q3: How to support efficient Deep Hashing algorithm?

- **Hash Sequence Conversion**: compares the image signatures generated by feature extraction with the threshold to yield the binary hash sequence for image retrieval. This can be supported by Interface Circuits ②.

Overview of Our PIM-DH Architecture

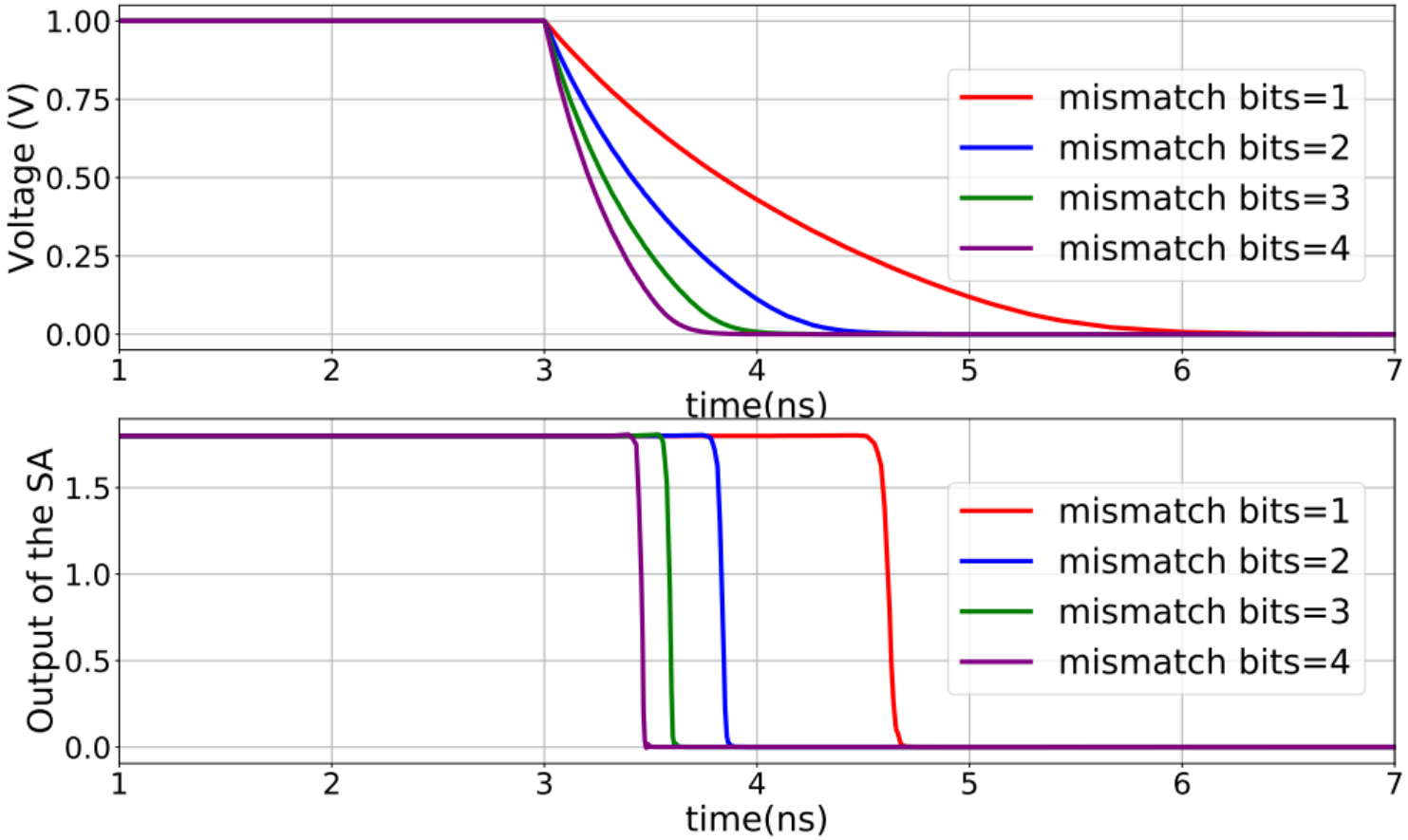


Q3: How to support efficient Deep Hashing algorithm?

- **Hamming Distance Calculation and Ranking**: can be efficiently processed by **CAM compute engine,** consisting of CAM crossbar③ assisted with **dedicated lightweight circuit④.**

The main idea is to architect an extra circuit to capture the latency of leakage current when the mismatch happens among the query and gallery sequences.

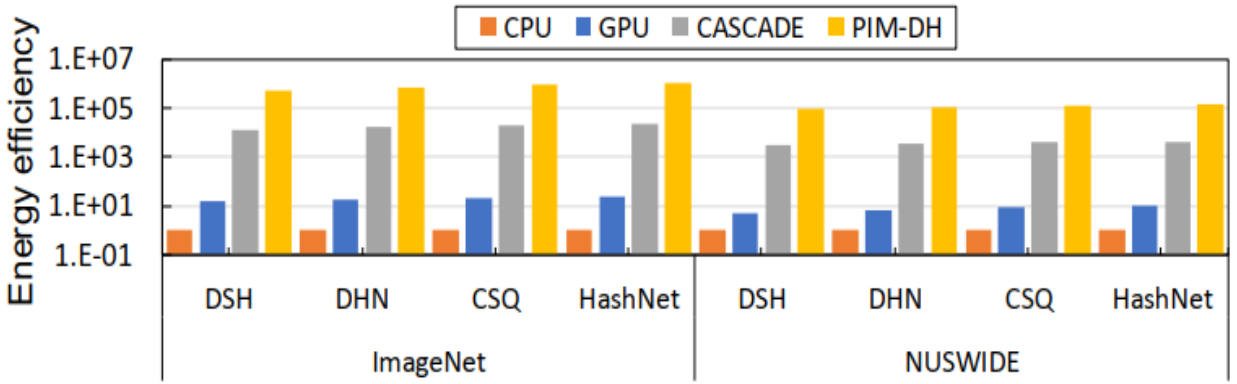
Experiment Results — mismatched bits of CAM

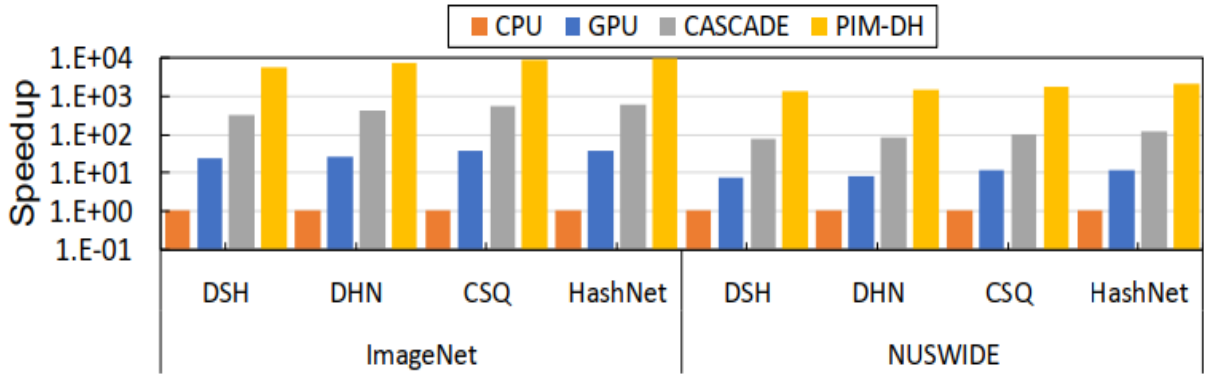


The voltage of the match line and the output of the SA versus mismatched bits

- The voltage pull-down is attributed to the increment of mismatched bits on the same match line.
- PIM-DH records the time of discharge to identify the number of mismatched bits by the designed circuits.

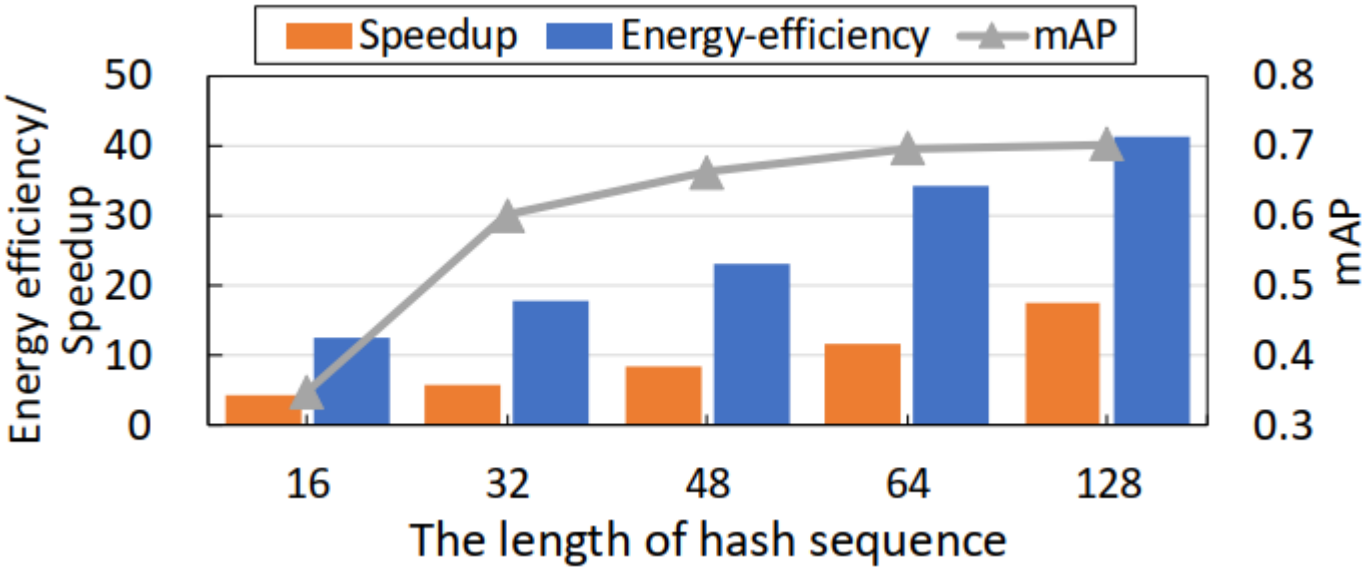
Experiment Results — Energy & Performance





- PIM-DH achieves 4.75E+03 speedup and 4.64E+05 energy reduction over CPU, 2.30E+02 speedup and 3.38E+04 energy reduction over GPU on average, respectively.
- PIM-DH can also achieve an average 17.49× speedup and 41.38× energy reduction over PIM design.

Experiment Results — length of hash sequence



- HashNet with a short hash sequence shows the best performance on PIM-DH.
- HashNet with a long hash sequence shows the most significant energy efficiency on PIM-DH.

Conclusion

- A novel hash sequence pruning algorithm
  - filter out redundant hash codes
- An efficient execute-search dual-engine PIM-based architecture
  - MAC compute engine
  - interface circuits
  - tailored CAM compute engine
- Keep high accuracy while gaining large performance improvement

Thank you !

PIM-DH: ReRAM-based Processing-in-Memory Architecture for Deep Hashing Acceleration

Fangxin Liu (Speaker)
Wenbo Zhao, Yongbiao Chen, Zongwu Wang, Zhezhi He, Rui Yang, Cheng Zhuo, and Li Jiang*
Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

59 DESIGN AUTOMATION CONFERENCE