# High Area/Energy Efficiency RRAM CNN Accelerator with Pattern-Pruning-Based Weight Mapping Scheme

**Songming Yu**, Lu Zhang, Jingyu Wang, Jinshan Yue, Zhuqing Yuan,
Xueqing Li, Huazhong Yang, Yongpan Liu

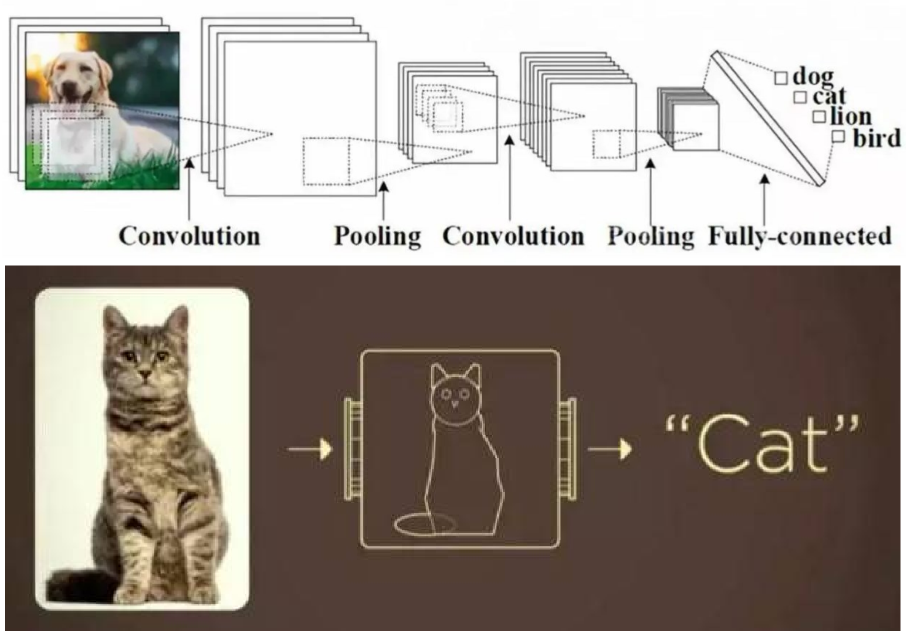
Email: ysm20@mails.tsinghua.edu.cn

Tsinghua University

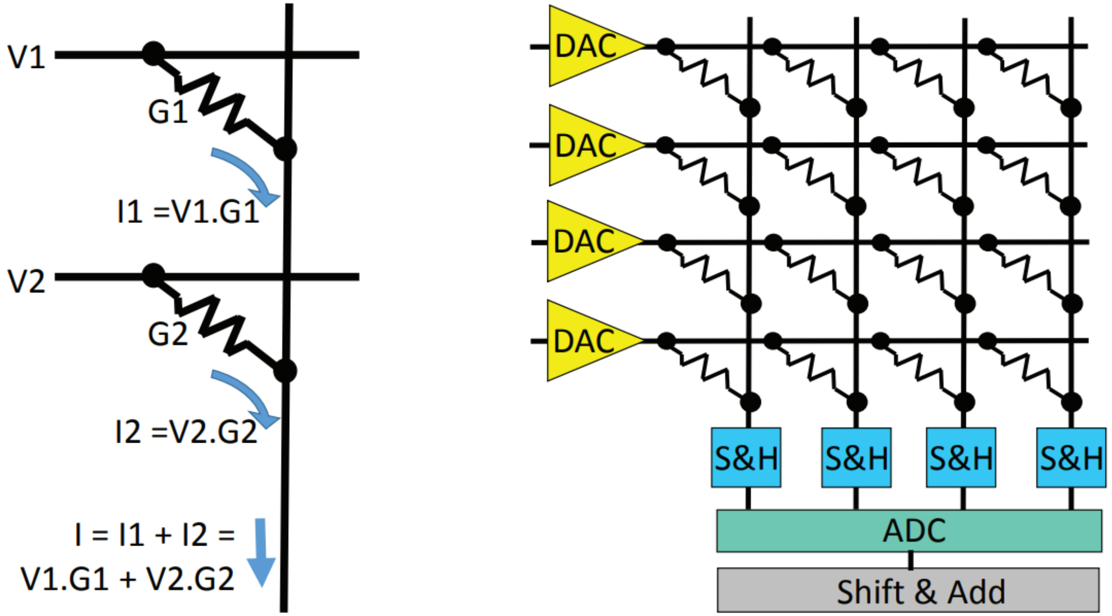
# Outline

- Introduction
- Background and motivation
- Design
  - Pattern-pruning-based weight mapping scheme
  - Architecture design
- Evaluation and results

# Introduction




Widely used convolutional neural networks (CNN)




(a) Multiply-Accumulate operation

(b) Vector-Matrix Multiplier

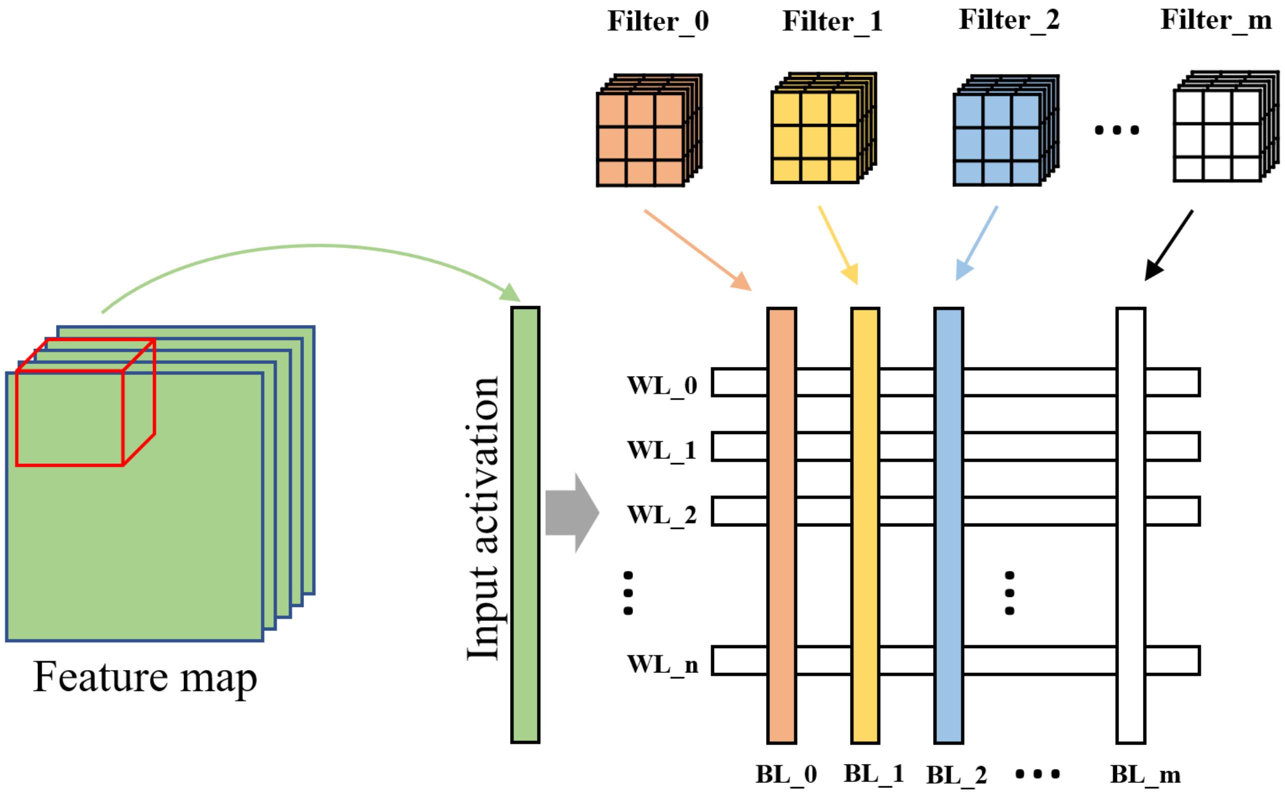
RRAM array for data storage and computing, figure from[1].

(a) Using a bitline to perform an analog sum of products operation.

(b) A RRAM crossbar used as a vector-matrix multiplier.


[1]John, Paul, Strachan, et al. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars[J]. Computer Architecture News, 2016.

# Background and motivation



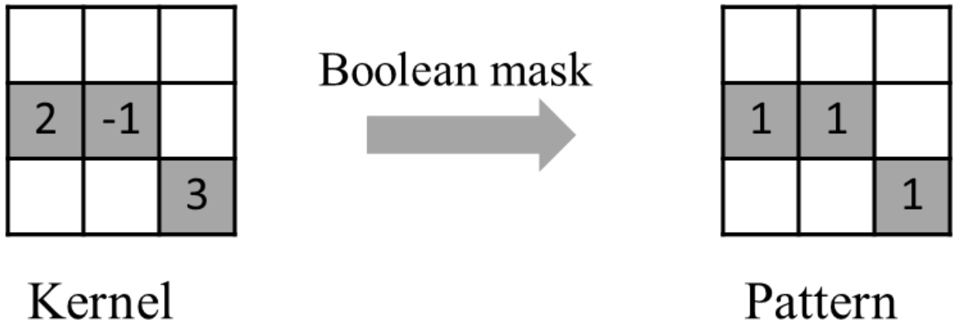

A straightforward weight mapping scheme for CNN in RRAM array

Weight mapping:
Every filer is mapped to one column

Executing:
Unroll the activation to vectors and fed into each row.

Disadvantages:
Hard to exploit the sparsity of the neural network
Overidealized. Activate a whole array will cause severe accumulate error.

# Background and motivation



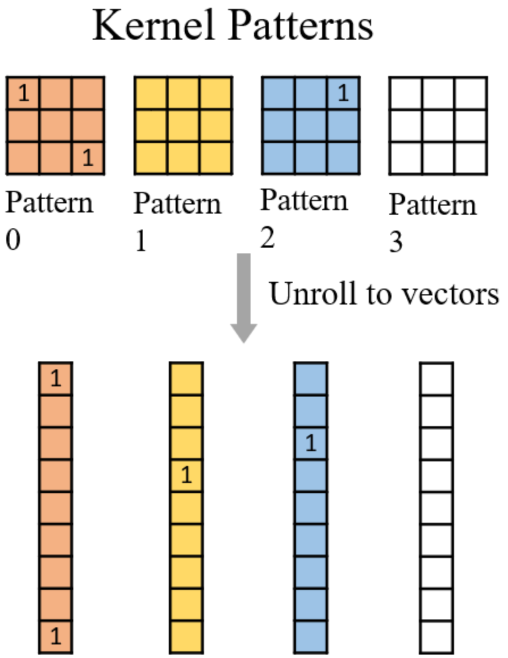

Max pattern number: $2^{3x3} = 512$

**Pattern pruning**: new opportunity for exploiting weight sparsity on RRAM-based CNN accelerator.

A intermediate type between non-structured pruning and structured pruning.

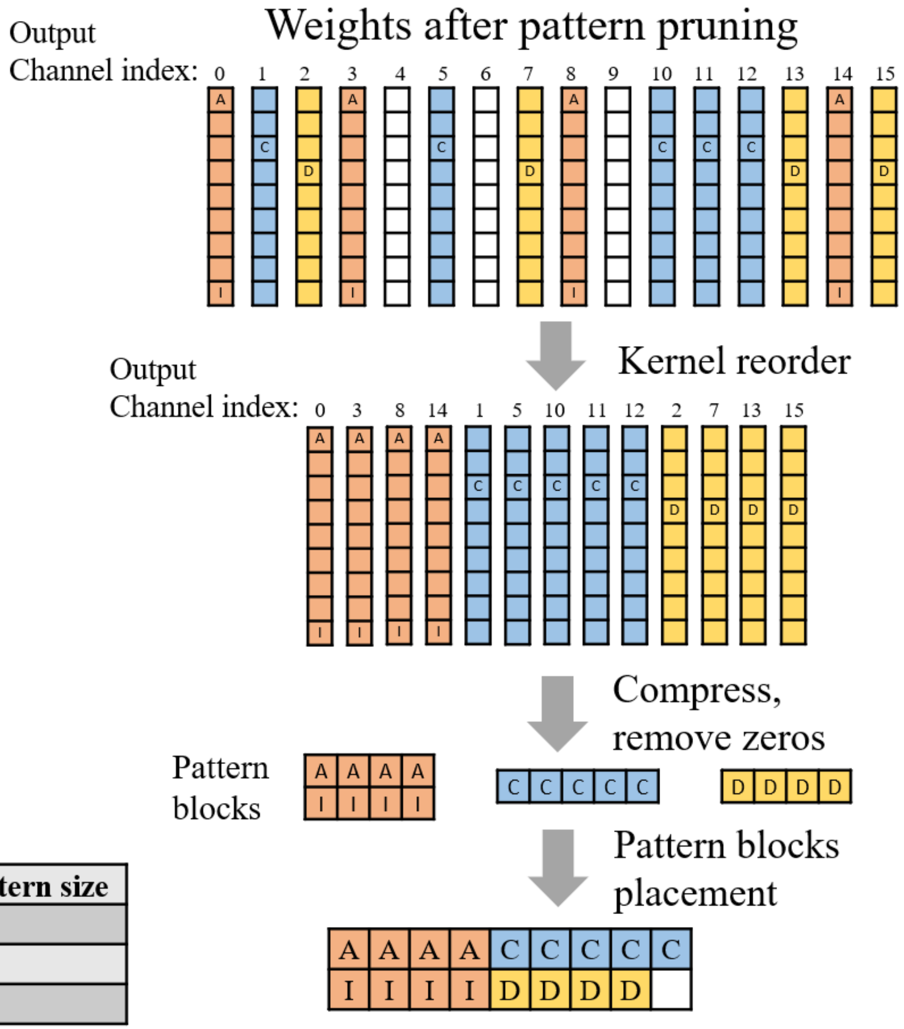
High accuracy & high sparsity,
with high regularity level

# Mapping scheme

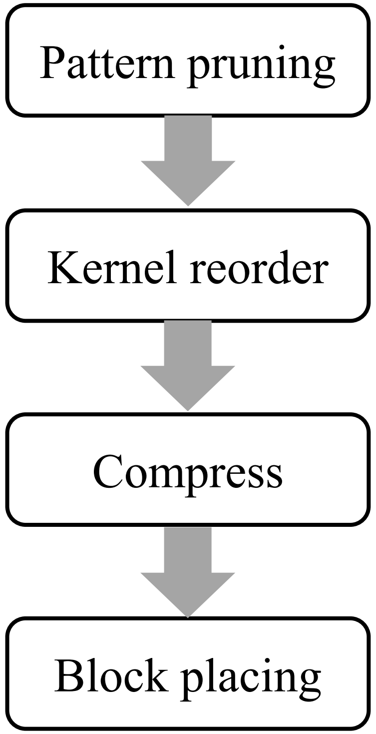



| Pattern ID | OC index | Pattern size |
|---|---|---|
| Pattern 0 | 0 3 8 14 | 2 |
| Pattern 1 | 1 5 10 11 12 | 1 |
| Pattern 2 | 2 7 13 15 | 1 |

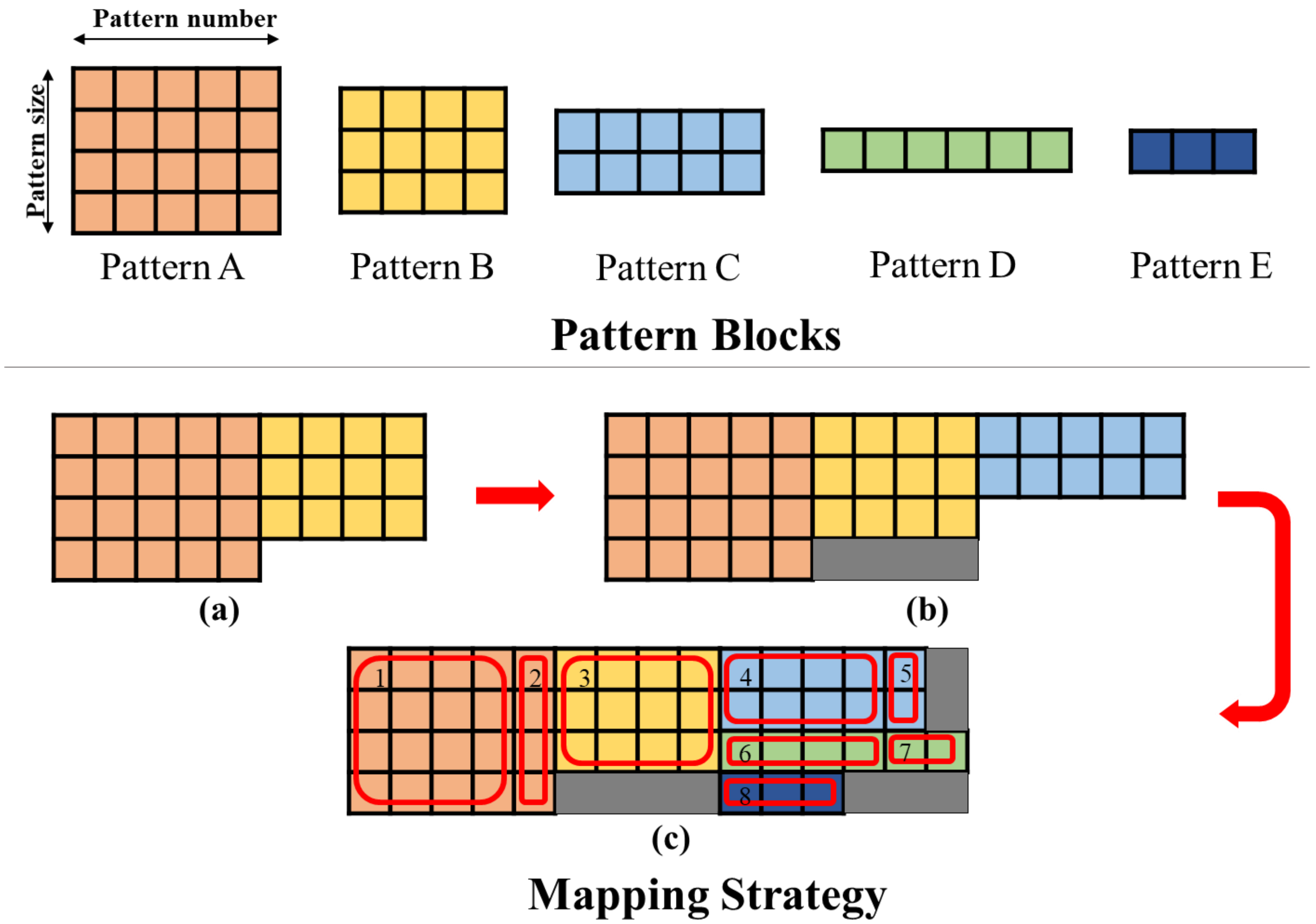
**Weight index**



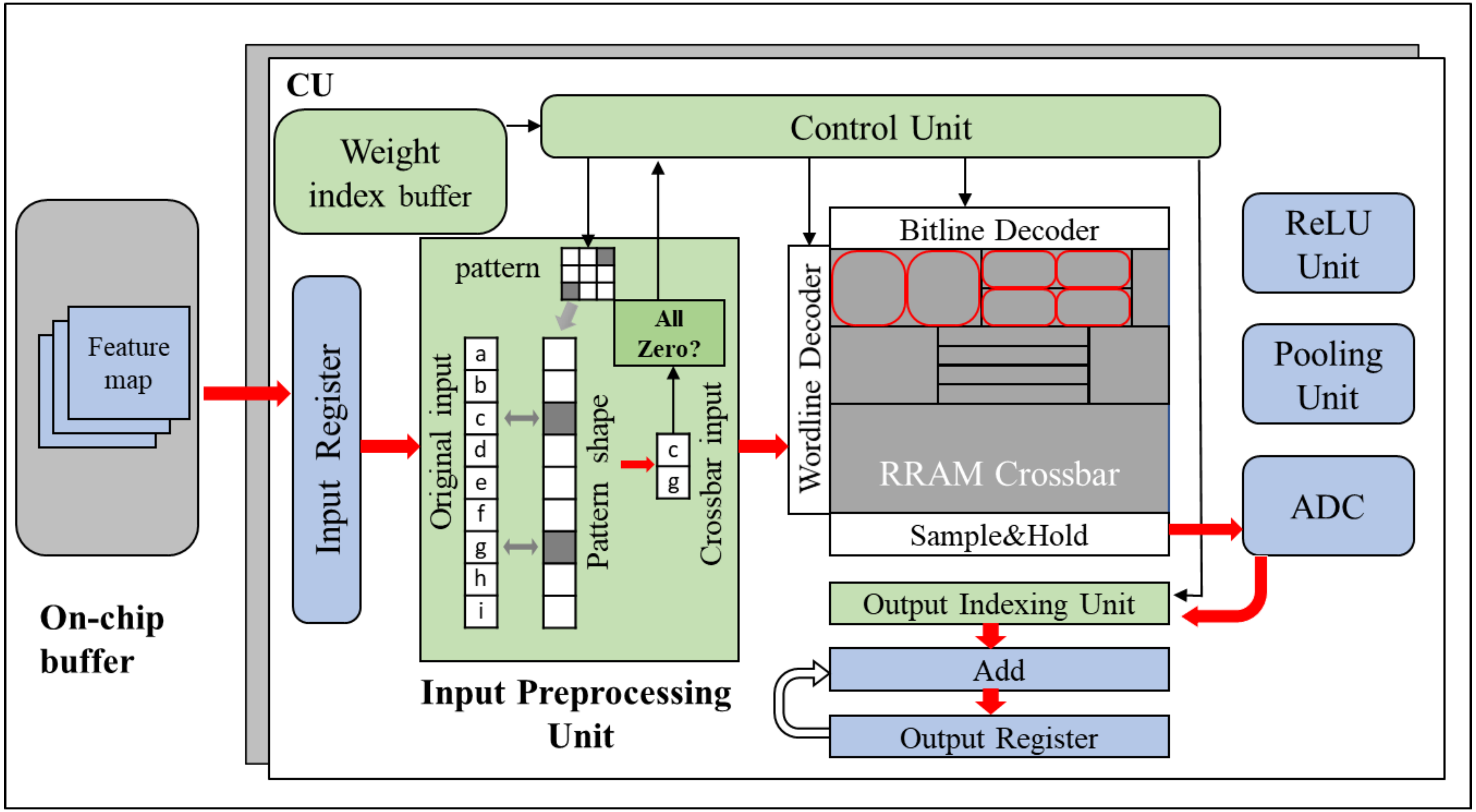

**Pattern-pruned mapping result**



# Pattern block placement



# Architecture design




**Input preprocessing unit**:
Compress input, skip all-zero

**Output indexing unit**:
Reorder the outputs and store them into right address

# Evaluation and results

| Components | Parameters | Spec | energy |
|---|---|---|---|
| ADC | Precision<br>Frequency | 8 bits<br>1.2 GSps | 1.67 pJ/op |
| DAC | Precision<br>Frequency | 4 bits<br>18 MSps | 0.0182 pJ/op |
| RRAM Array | OU size<br>bits per cell<br>size | $9 \times 8$<br>4<br>$512 \times 512$ | 4.8 pJ/OU/op |

## Evaluation setup

**Hardware Parameter** for energy consuming: table on the left.

**Simulator**: behavior level, built in python.

**Network**: VGG16 trained on CIFAR-10, CIFAR-100 and ImageNet.

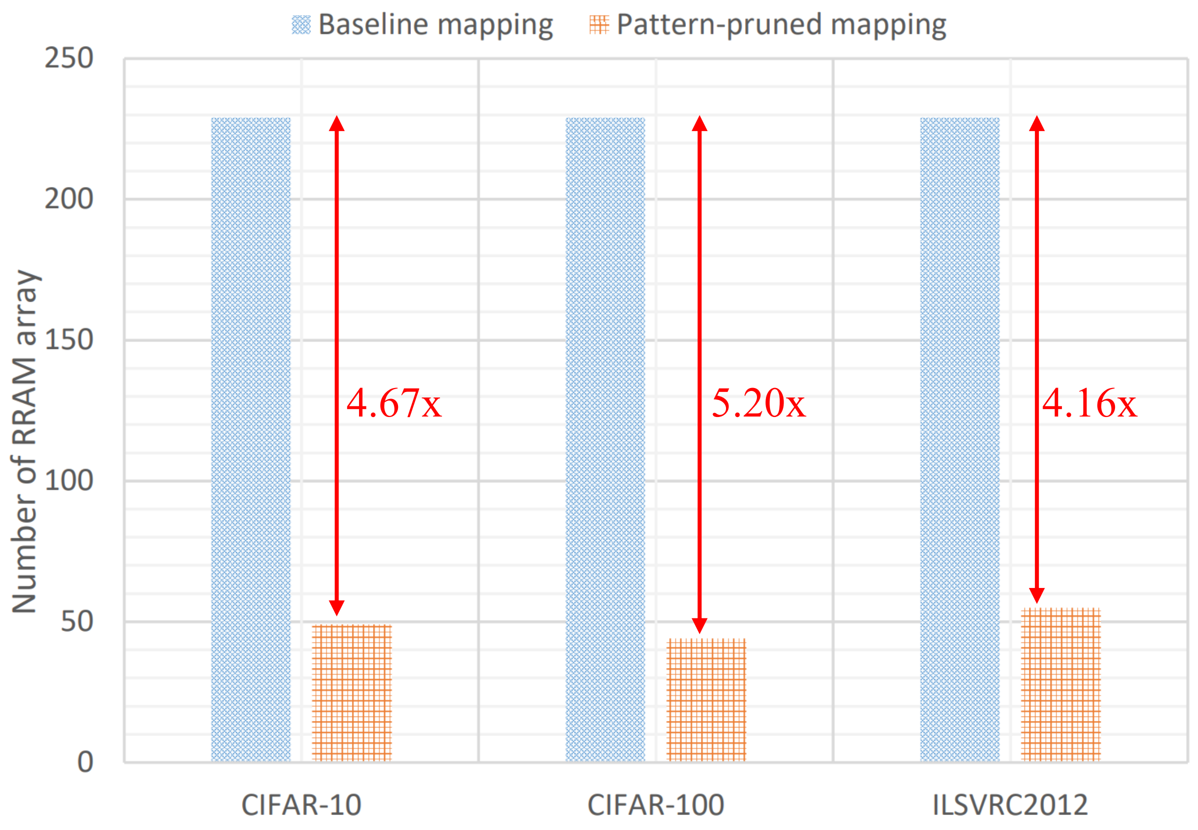
**Baseline**: a straightforward weight mapping scheme.

| Dataset | Sparsity | Pattern Numbers in Each Conv layer | Total | top-1 | top-5 |
|---|---|---|---|---|---|
| CIFAR-10 | 86.03%(+4.08%) | [2, 2, 2, 6, 8, 8, 8, 6, 5, 4, 6, 6, 8] | 71 | 92.63%(-0.09%) | / |
| CIFAR-100 | 85.23%(+3.28%) | [2, 2, 2, 2, 2, 8, 8, 8, 5, 6, 7, 6, 8] | 66 | 72.73%(+0.01%) | 92.23%(+0.79%) |
| ImageNet | 82.48%(-0.90%) | [2, 2, 2, 2, 2, 9, 12, 12, 9, 10, 6, 4, 4] | 76 | 71.15%(-0.75%) | 89.98%(-0.51%) |

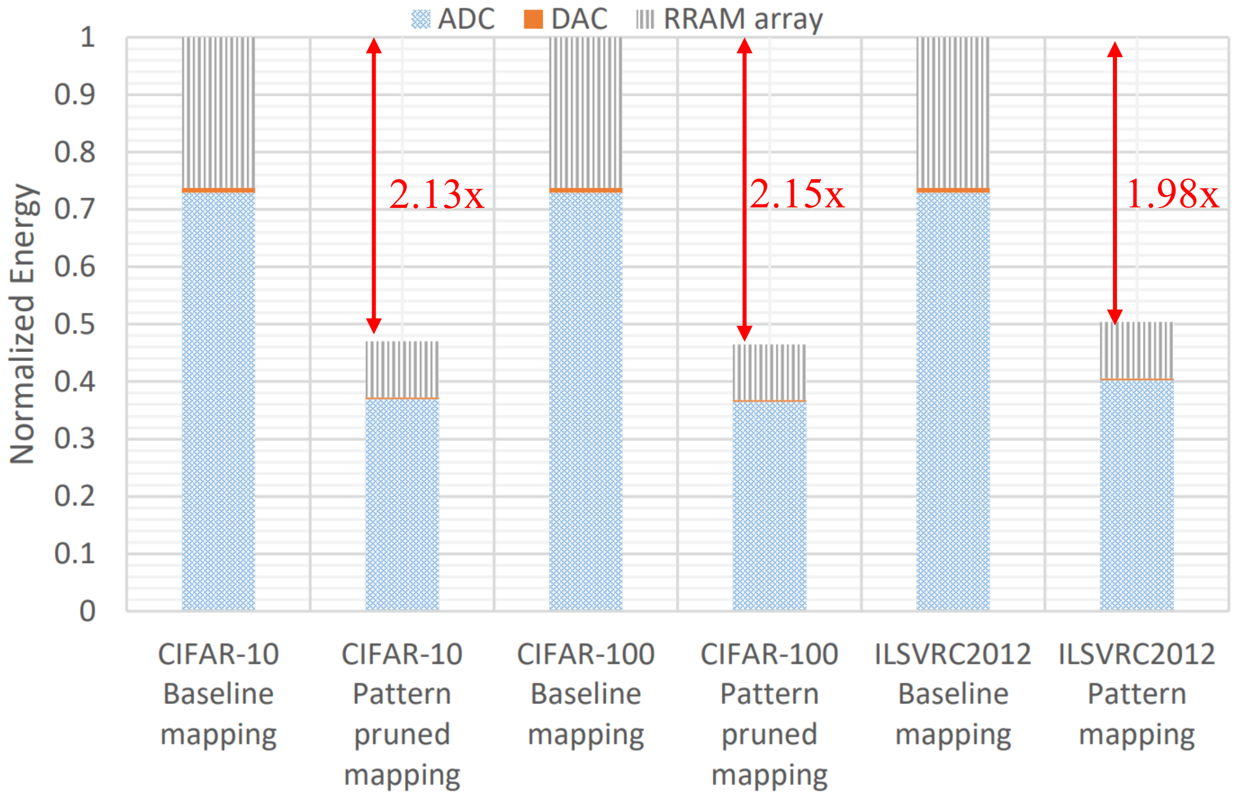
Pattern pruning results

# Evaluation and results

## RRAM area




## Normalized energy



# Experiment results summary

- **Area efficiency**: 4.67x/5.20x/4.16x for networks trained on CIFAR-10, CIFAR-100, and ImageNet, respectively. This means that we save78.5%/80.8%/76.0% RRAM array comparing to the baseline.
- **Energy efficiency**: 2.13x/2.15x/1.98x on CIFAR-10, CIFAR-100, and ImageNet, respectively (only RRAM, ADCs and DACs in energy evaluation).
- **Performance Speedup**: 1.35x/1.15x/1/17x on CIFAR-10, CIFAR-100, and ImageNet, respectively. The speedup is achieved mainly by the deleted all-zero patterns which are neither stored in RRAM nor computed.

# Conclusion

- A novel area-efficiency weight mapping scheme based on pattern pruning
  - High area efficiency
- An RRAM-based sparse CNN accelerator architecture
  - High energy efficiency

# Thank you!

Songming Yu
Email: ysm20@mails.tsinghua.edu.cn