Edge Inference Engine for Deep & Random Sparse Neural Networks with 4-bit Cartesian-Product MAC Array and Pipelined Activation Aligner

Kota Ando, Jaehoon Yu, Kazutoshi Hirose, Hiroki Nakahara, Kazushi Kawamura, Thiem Van Chu, and Masato Motomura

AI Computing Research Unit (ArtIC), Tokyo Institute of Technology

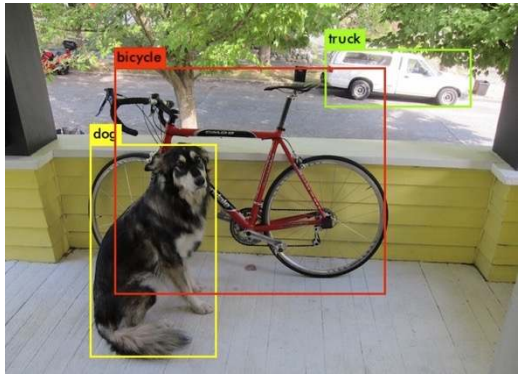
August 23-24, 2021

Abstract

A 4b-quantized convolutional neural network (CNN) inference engine for edge-AI is presented featuring a Cartesian-product MAC array and pipelined activation aligners targeting deep-/random-pruned models. A 40nm prototype with 32x32 MACs and 5Mb SRAM runs at 534 MHz, 1.07 TOPS, 352 mW at 1.1V, and attains 5.30 dense TOPS/W, 234 MHz at 0.8V. Sparse TOPS/W reaches 26.5 when running a randomly pruned model (after 88% pruning). Training algorithms for obtaining highly efficient sparse/quantized models are also proposed.

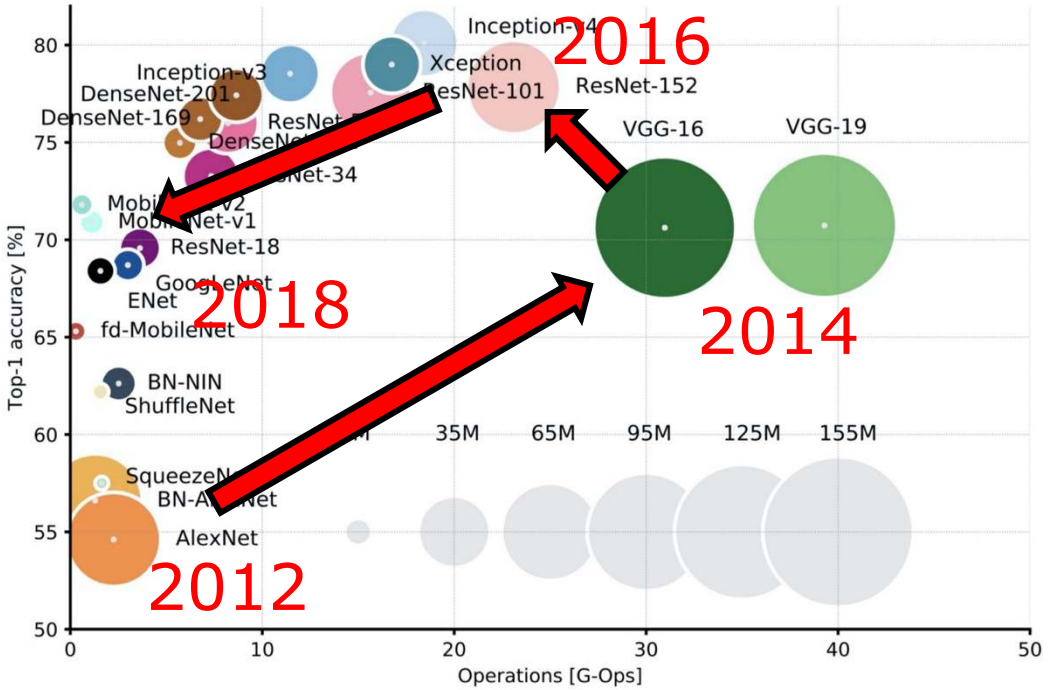
Background

- Fast evolution of DNN models
  - Getting more complex and huge for achieving higher accuracy
  - Algorithms for compact models on edge devices
    - ✓ Quantization
    - ✓ Separable Conv: MobileNet (2018)
    - ✓ Random sparsity
    - ✓ Compact yet irregular structure; Less opportunity of reusing data
  - Efficient and accurate processing of **sparse** DNN is desired



https://pjreddie.com/darknet/yolo/

Freepik.com



https://culurciello.medium.com/analysis-of-deep-neural-networks-dcf398e71aae

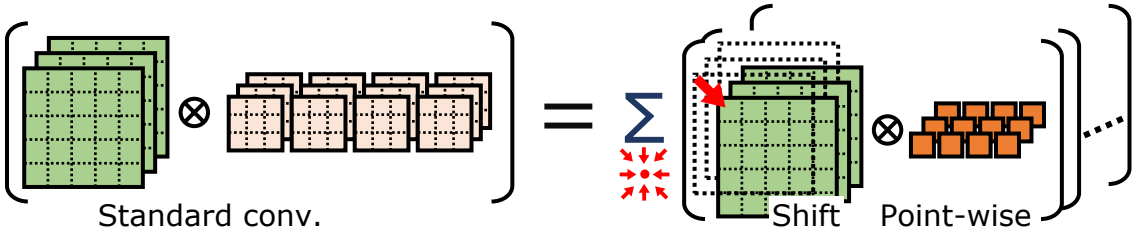
Motivation

Toward "efficient" hardware processing ……

- **Reuse data** as much as possible (OPs/byte ↗)
- **Omit unneeded computation** and data transfer

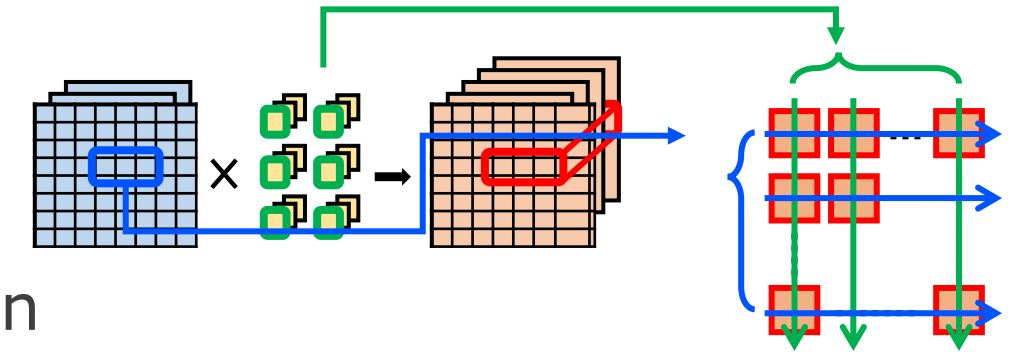
Point-wise Conv as the primitive

- ✓ Any Conv can be factorized into a superposition of PW Convs
- ✓ Easily exploits random sparsity



Proposal

- **Cartesian-product MAC** array with **activation aligners** for efficient PW Conv processing



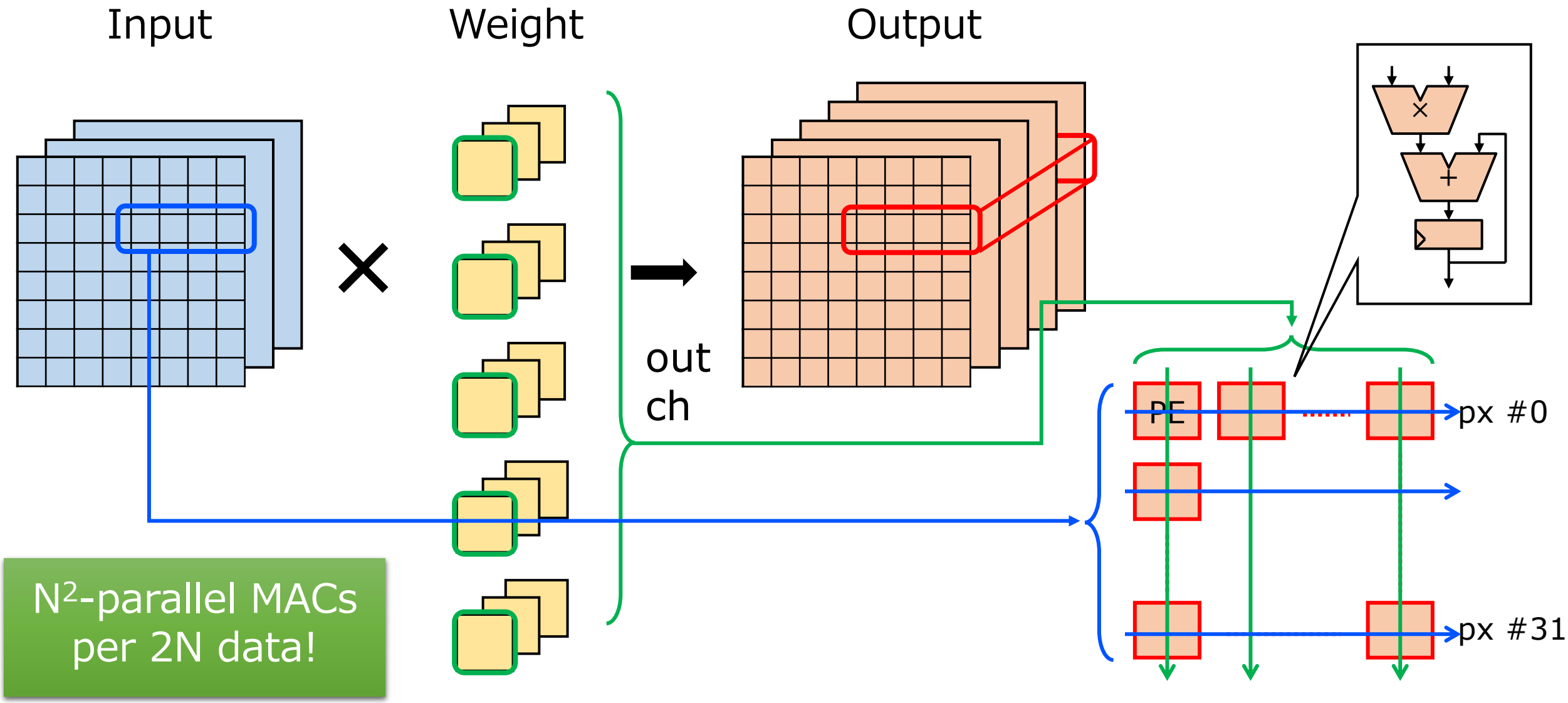
- Sparsity-aware model reconstruction

Outline

- Background
- Proposal:

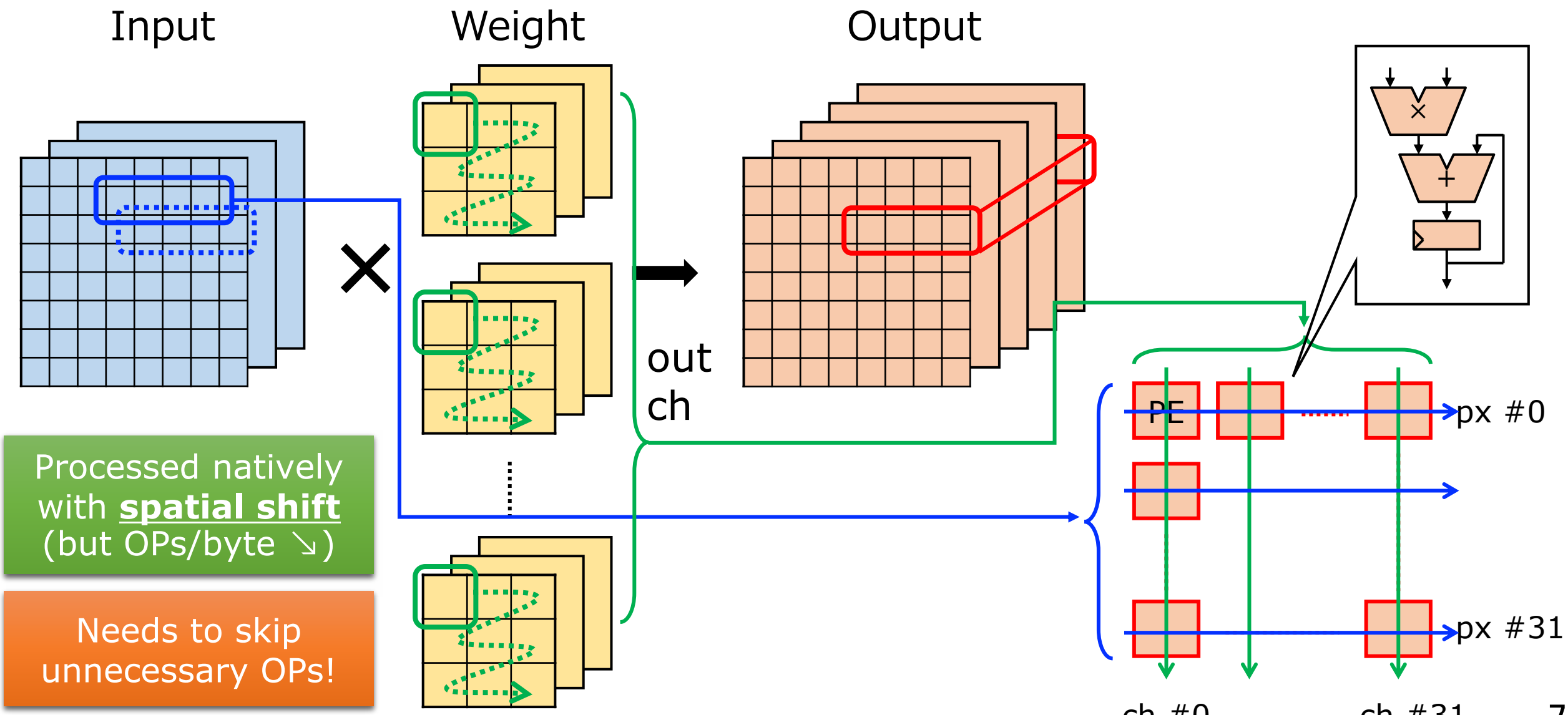
  Cartesian-product MAC array with activation aligners
- Model construction and evaluation
- Chip evaluation
- Conclusion

Cartesian-product MAC array – Point-wise Conv



N^2-parallel MACs per 2N data!

Cartesian-product MAC array – Standard Conv

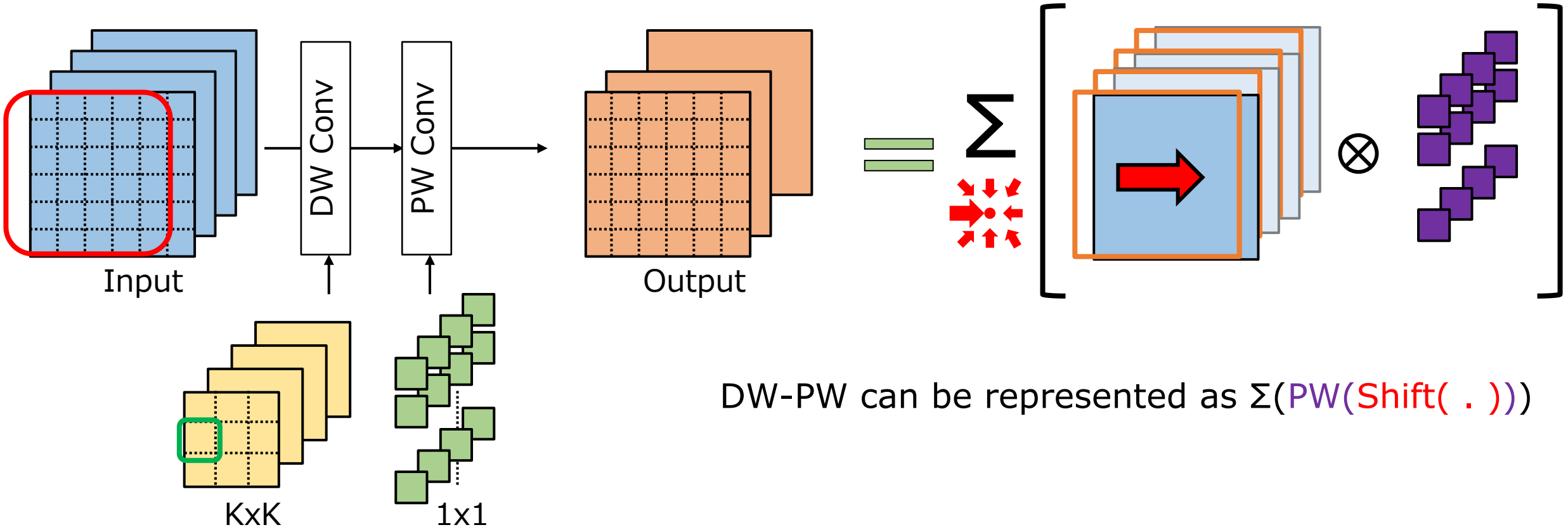


Processed natively with **spatial shift** (but OPs/byte ↘)

Needs to skip unnecessary OPs!

Shift-based convolution

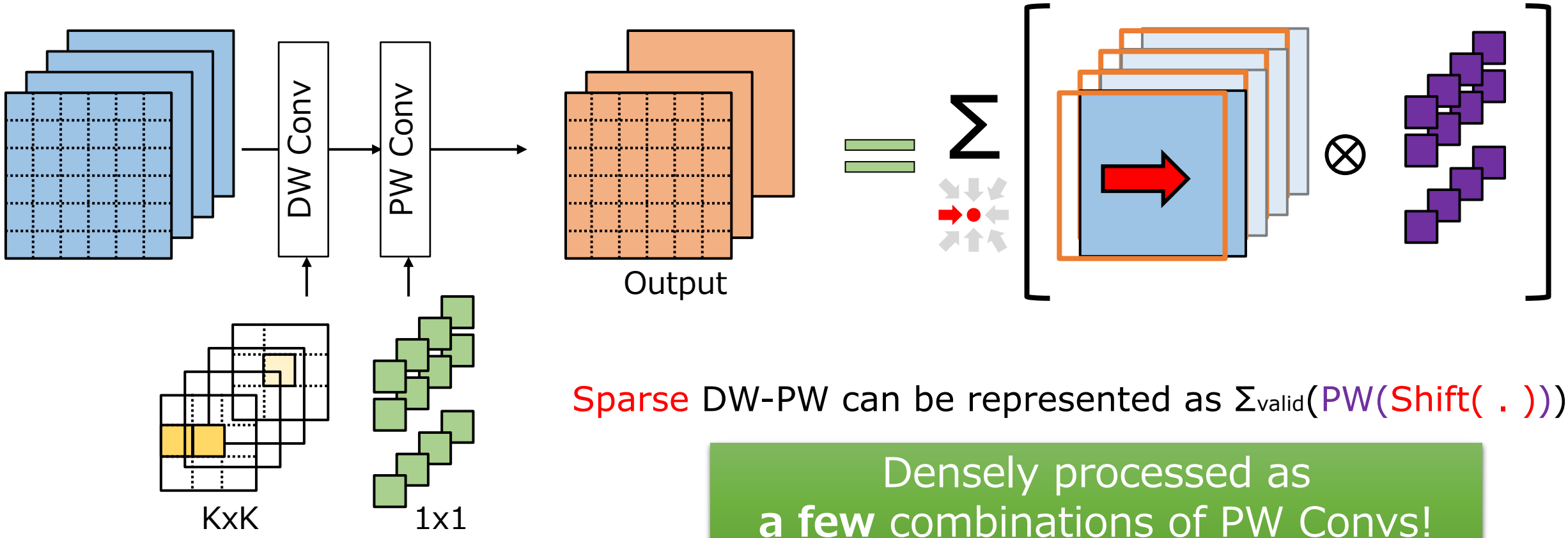
- Any Conv can be represented as superposition of "Shift-PW"
  - Depth-wise Conv is (almost always) followed by point-wise Conv



DW-PW can be represented as Σ(PW(Shift(.)))

Shift-based convolution

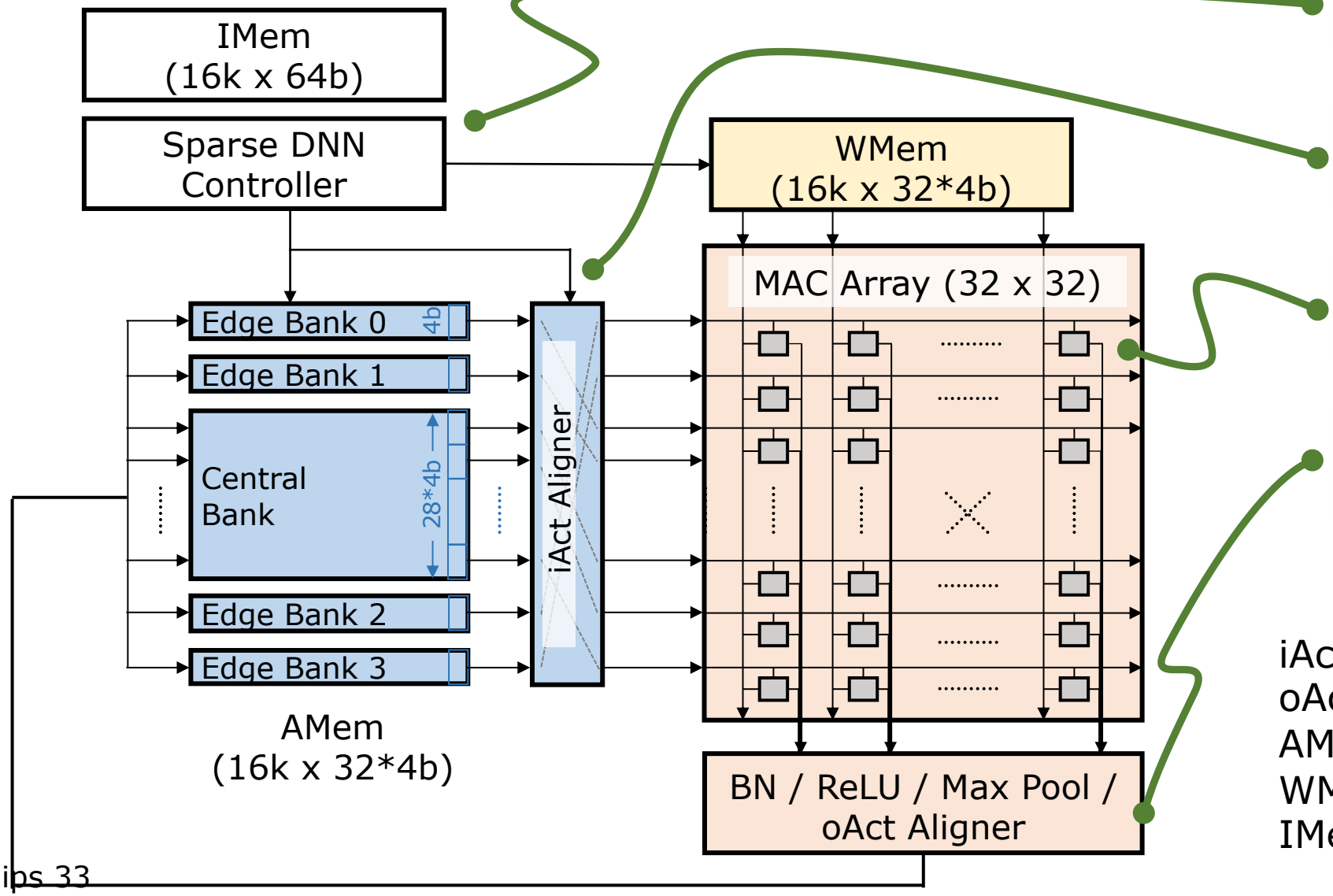
- How about pruned DW kernels?
  - Can skip unnecessary Shifts!
  - Shift Net [B. Wu, CVPR'18], Sparse Shift Layer [W. Chen, arxiv, 2019]



Sparse DW-PW can be represented as $\Sigma_{valid}(PW(Shift(\ .\ )))$

Densely processed as
a few combinations of PW Convs!

Overall architecture

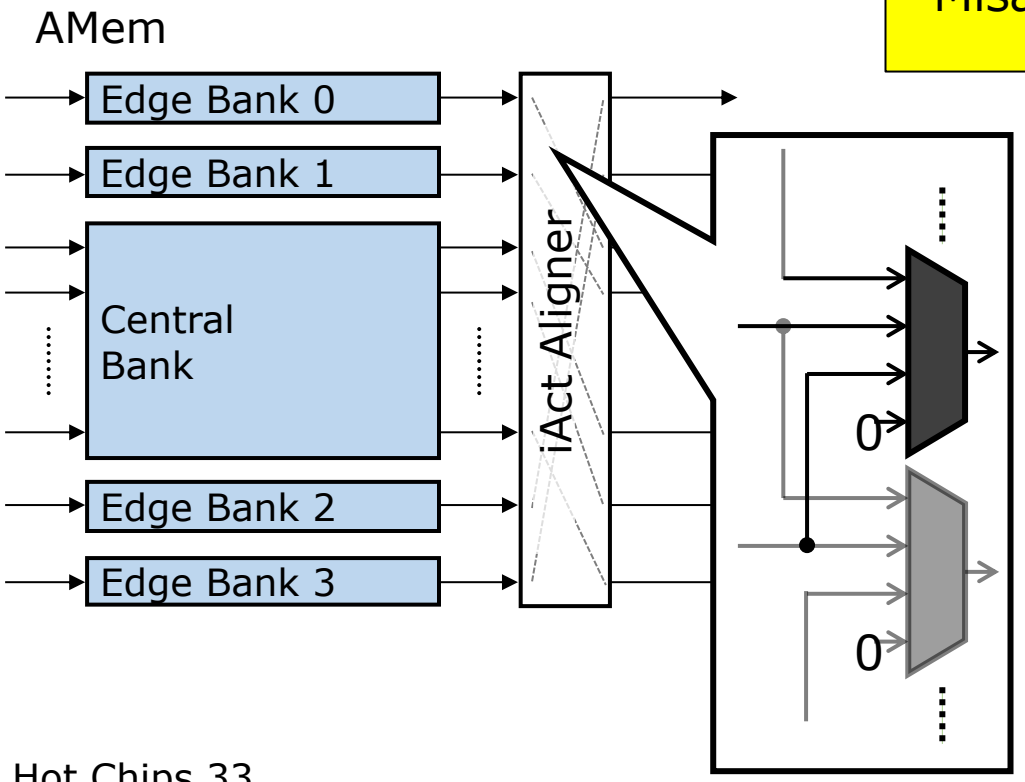


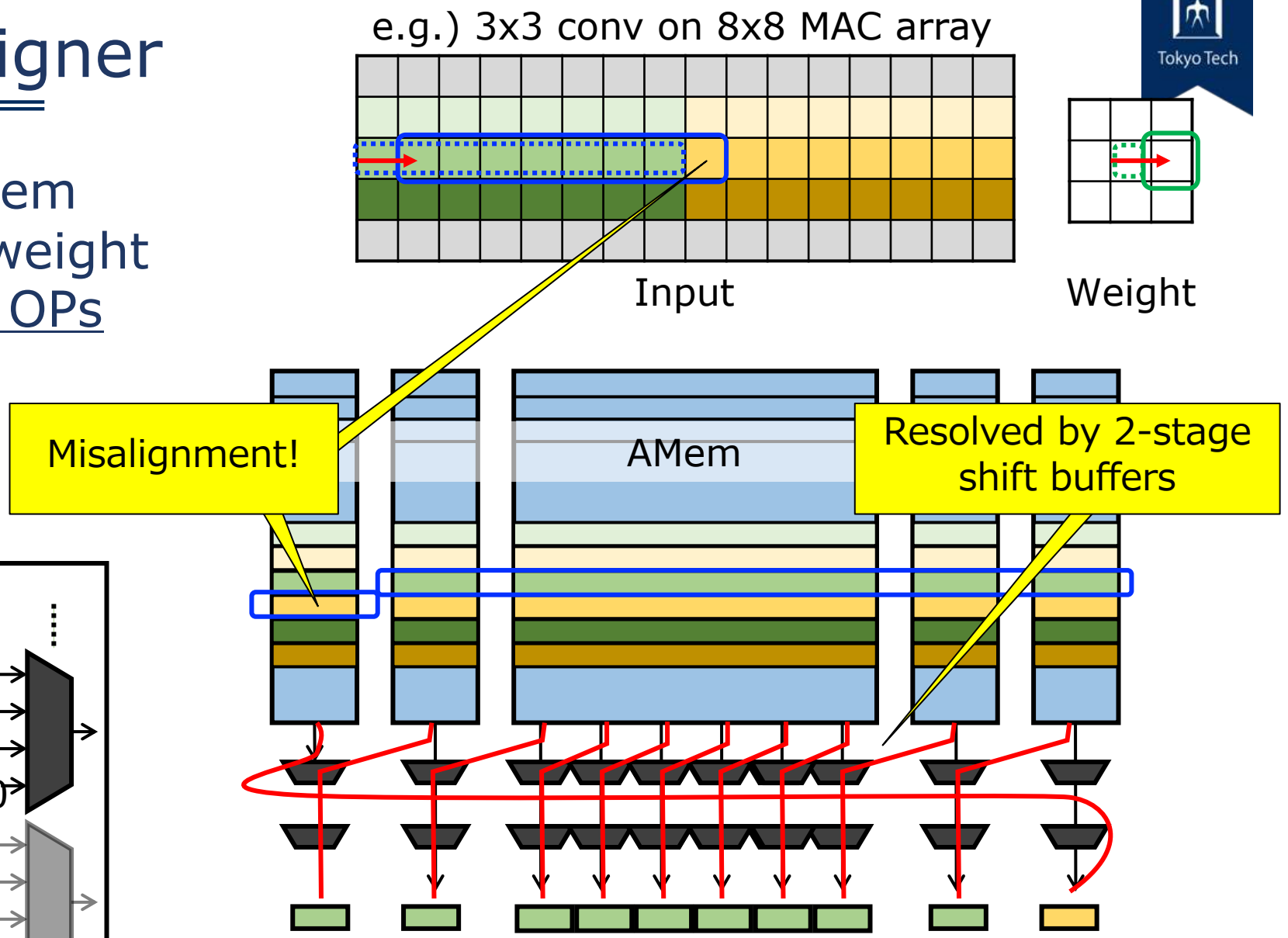
- Weight zero-skipping
- Built-in spatial shift with no OPs
- Cartesian-product MAC
- Pipelined post-processing OPs

iAct = Input Activation
oAct = Output Activation
AMem = Activation Memory
WMem = Weight Memory
IMem = Instruction Memory

Input activation aligner

- Fetches iActs from AMem corresponding to the weight position with no extra OPs

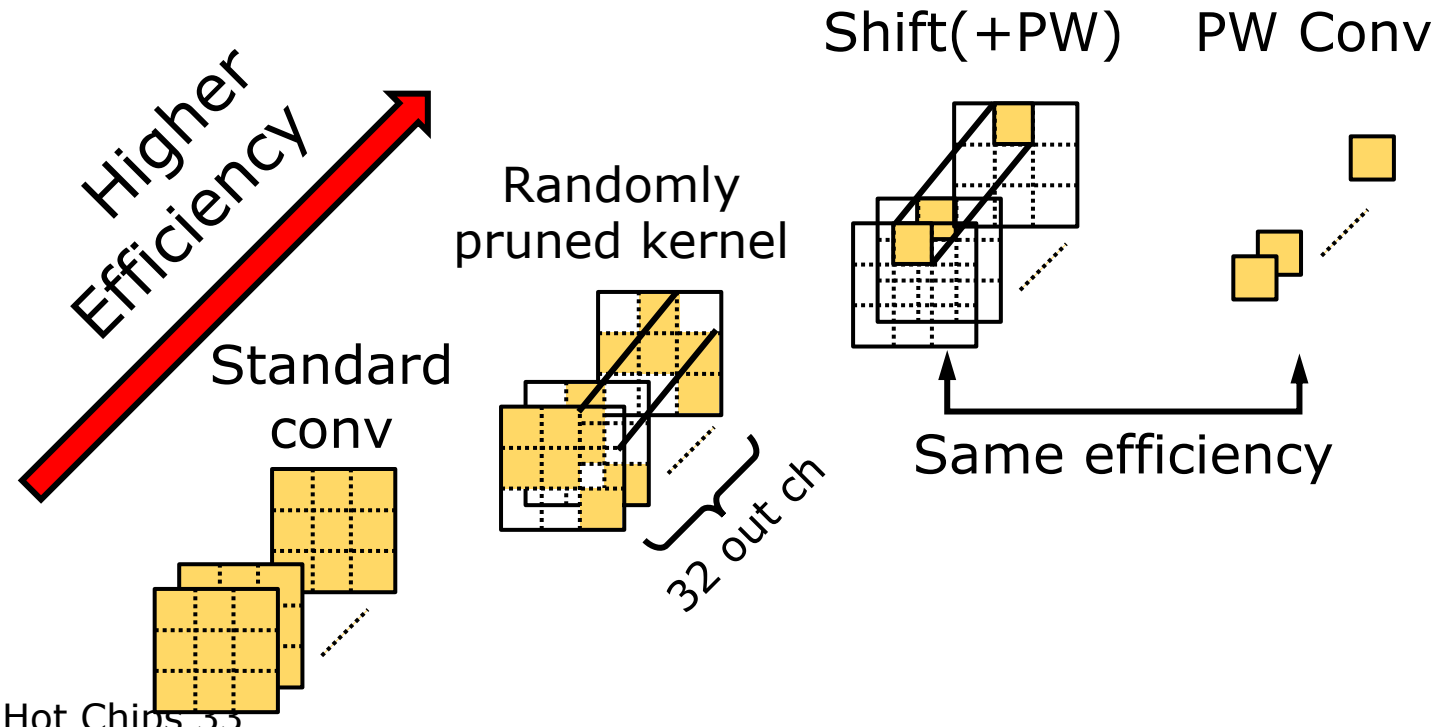


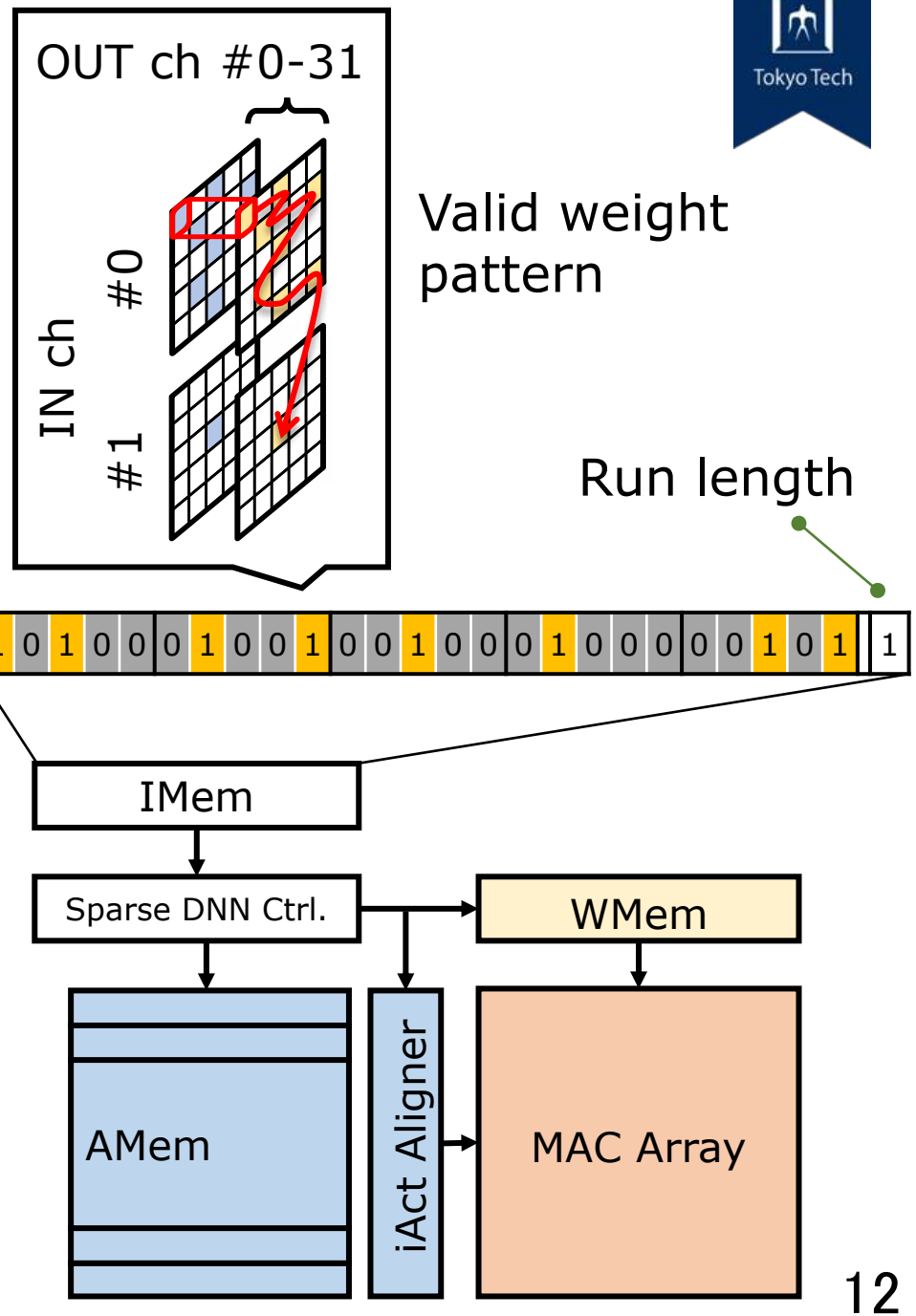


Shift buffers and independently-addressable banks resolve data misalignments

Sparsity control

- Weight sparsity can be exploited
  - Every pixel in each 32 output ch (= WMem width) can be skipped according to "valid W pattern" in IMem
  - Only "valid" weights are stored in WMem



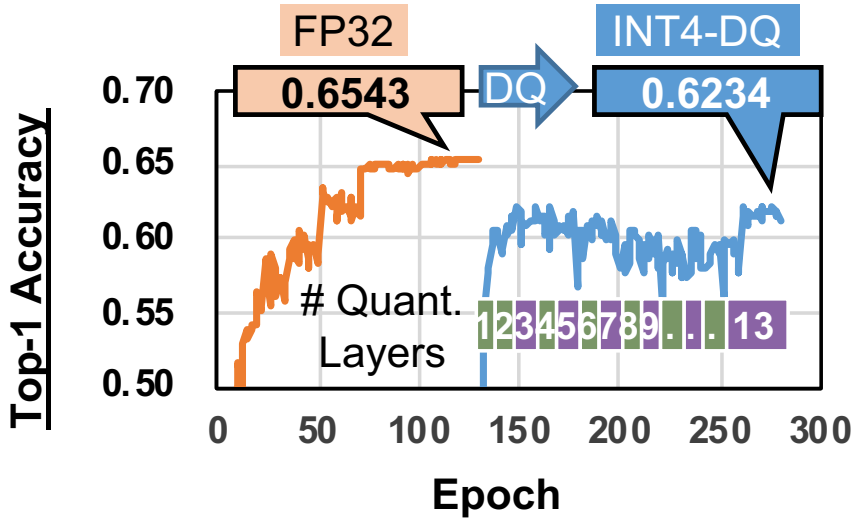


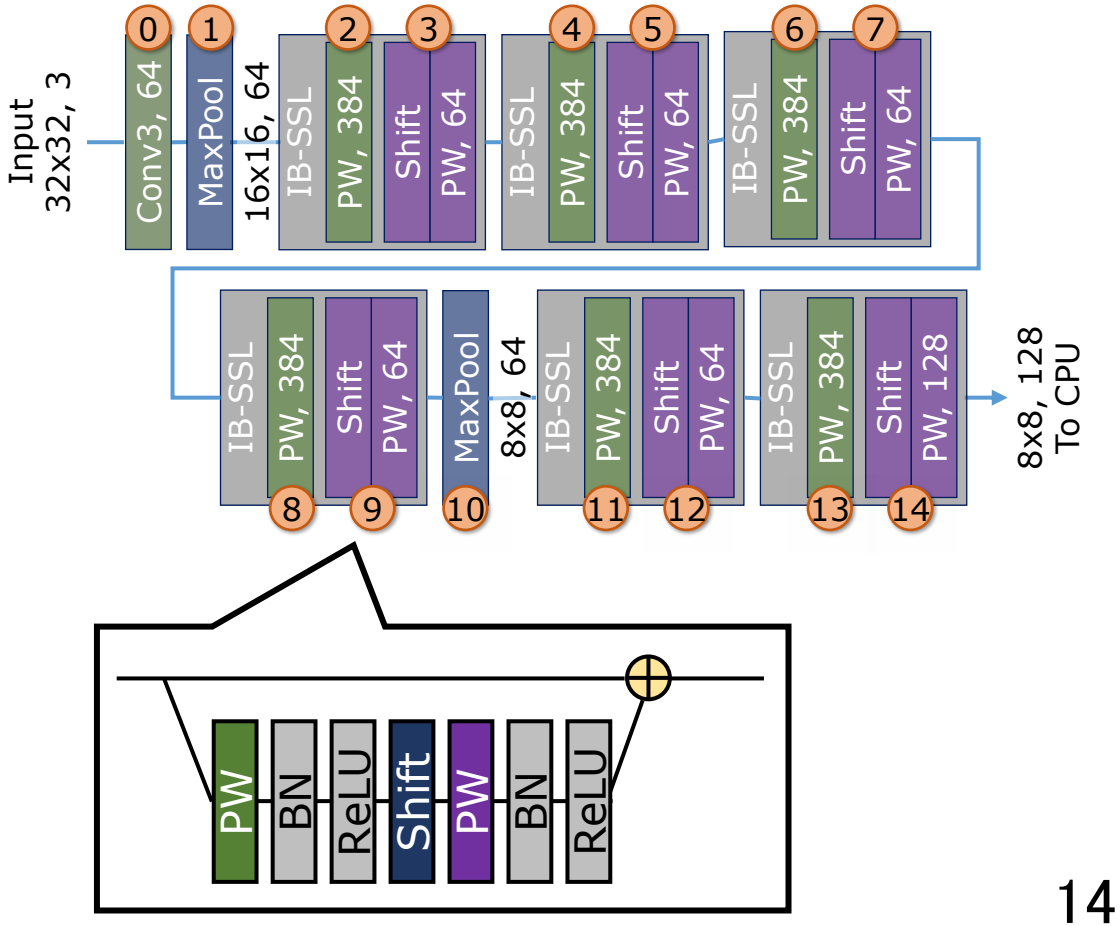
Outline

- Background
- Proposal:
  - Cartesian-product MAC array with activation aligners
- Model construction and evaluation
- Chip evaluation
- Conclusion

Model 1: Prototype CIFAR-100 model

- A prototype consisting of PW Conv and spatial shift only
  - Based on Inverted Bottleneck Sparse Shift Layers (IB-SSL) [W. Chen, arxiv, 2019]
- Dynamic quantization (DQ)
  - Pretrain a model in FP32
  - Retrain in INT4 adjusting the distribution of weights and activations layer by layer
- Results



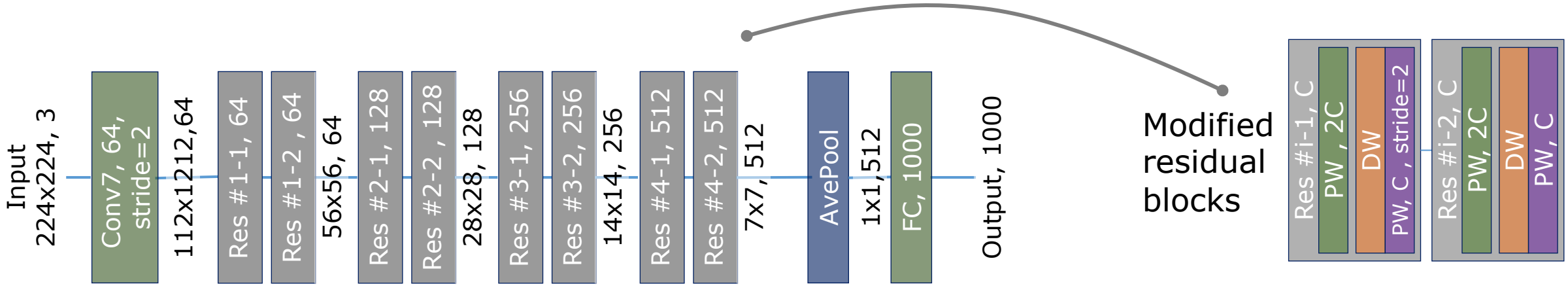


Model 2: Pruned ResNet-18 for ImageNet

Tokyo Tech

ResNet-18-like model

- Residual blocks → separable PW-DW-PW blocks
- DW kernels are gradually pruned in favor of the "Shift-PW" concept; the block is processed as $\Sigma_{valid}(PW(Shift(\ PW(\ .\ )\ )))$
- Then, DQ is applied to the pruned weights (next page)

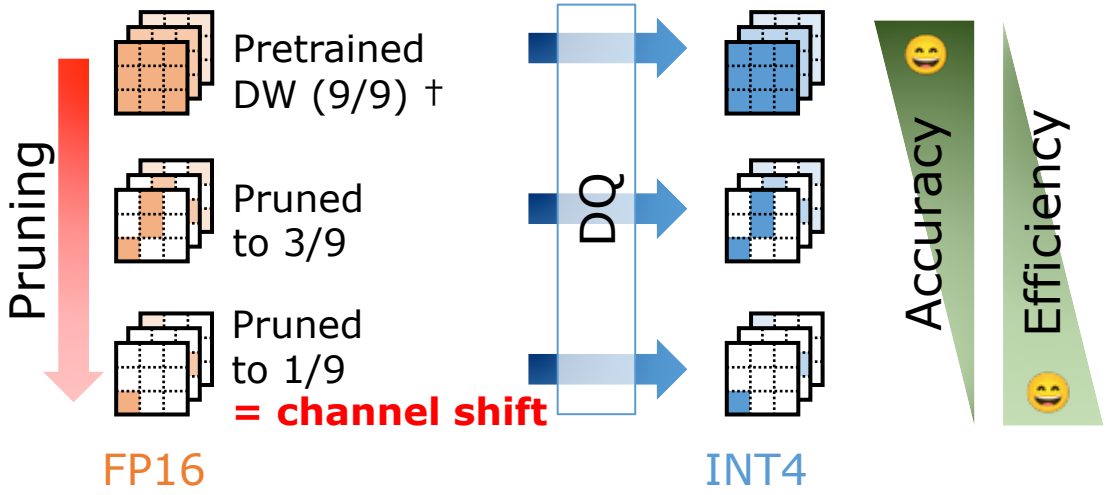


Model 2: Pruned ResNet-18 for ImageNet

Gradual Pruning and Dynamic Quantization

1. Pretrain the model in FP16
2. Retrain to increase the zero elements in 3x3 DW kernels gradually
3. Apply DQ to each of the pruned models

➢ **Controllable accuracy-efficiency trade-off** by selecting the level of pruning



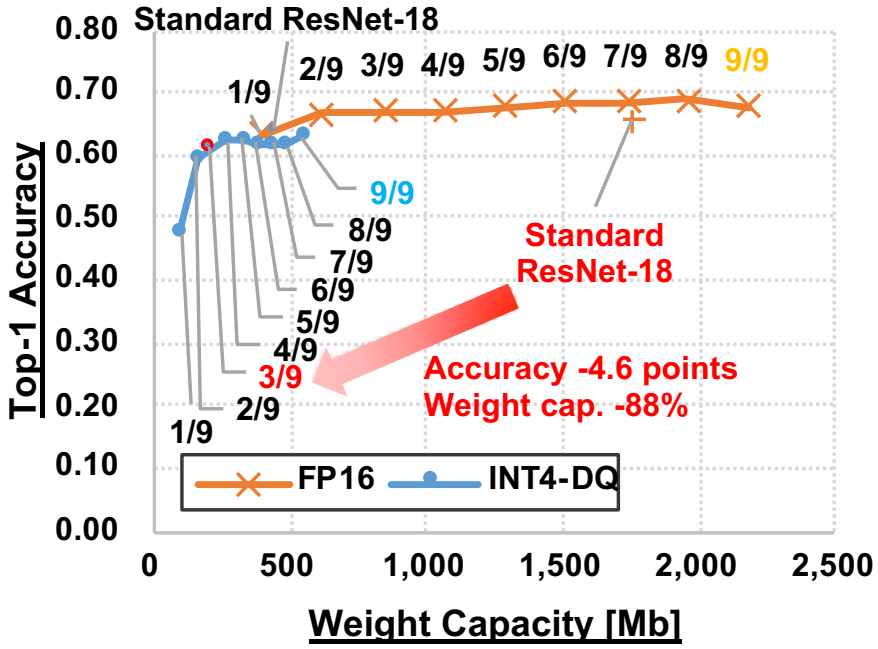
Results

➢ No pruning (9/9)
- 67.6%@FP16
- 62.9%@INT4

➢ Pruned to 3/9
- 61.2%@INT4

➢ Achieves accuracy -4.6% with weight capacity -88% from the base ResNet-18

† Sparsity = (#remaining weights) /(#total weights)



Outline

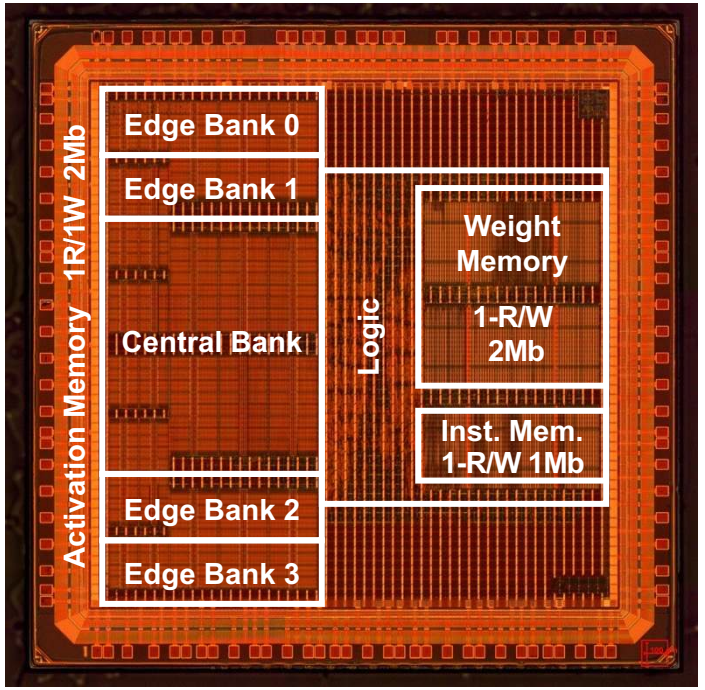
- Background
- Proposal:
  - Cartesian-product MAC array with activation aligners
- Model evaluation
- Chip evaluation
- Conclusion

Prototype chip

- Fabricated a prototype chip
  - 32x32 4-bit MAC array
  - 534 MHz, 3.03 TOPS/W **(dense)** at 1.1V
  - 234 MHz, 5.30 TOPS/W **(dense)** at 0.8V

  - Shift operations are not counted as OP: Achieves much higher "sparse TOPS/W" with highly sparse models
    - ✓ 5.30 TOPS/W (**dense**; measured) = 26.5 TOPS/W (**sparse**; effective) @0.8V assuming the model is pruned to 1/9

Technology	**TSMC 40nm CMOS (LP)**
Chip size	**3mm x 3mm**
Core area	**SRAM: 3.6mm^2 Logic: 0.9mm^2**
Core V_{DD}	**0.75 - 1.2V**
Frequency	**534MHz@1.1V 234MHz@0.8V**
Power Consumption	**352mW@1.1V 88mW@0.8V**
Gate Count	**746K Gate**
SRAM	**AMem: 16k x 32 x 4 WMem: 16k x 32 x 4b IMem: 16k x 64b -- Total 5Mb**



Outline

- Background
- Proposal:
  - Cartesian-product MAC array with activation aligners
- Model evaluation
- Chip evaluation
- Conclusion

Conclusion

- Cartesian-product MAC array with pipelined activation aligners
  - Processes **sparse** channel-shift and point-wise Conv densely
  - Any Conv can be processed as a superposition of Shift and PW
- Sparsity-aware model construction
  - Gradual pruning obtains a model with (arbitrary) **sparsity** bringing controllable efficiency-accuracy trade-off
- Fabricated chip
  - Achieved 5.30 TOPS/W (**dense**);
    equivalent to 26.5 TOPS/W (**sparse**) assuming a pruned-to-1/9model

Tokyo Tech

Thank You