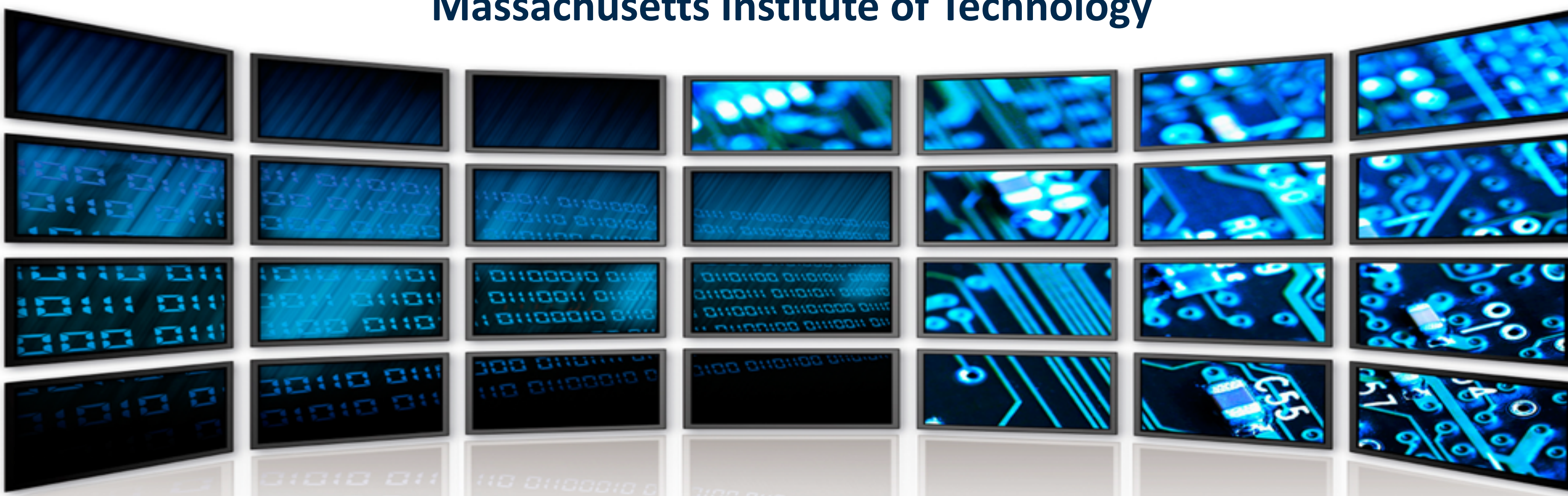
Balancing Efficiency and Flexibility for DNN Acceleration

Vivienne Sze (ft. Yu-Hsin Chen)

Massachusetts Institute of Technology

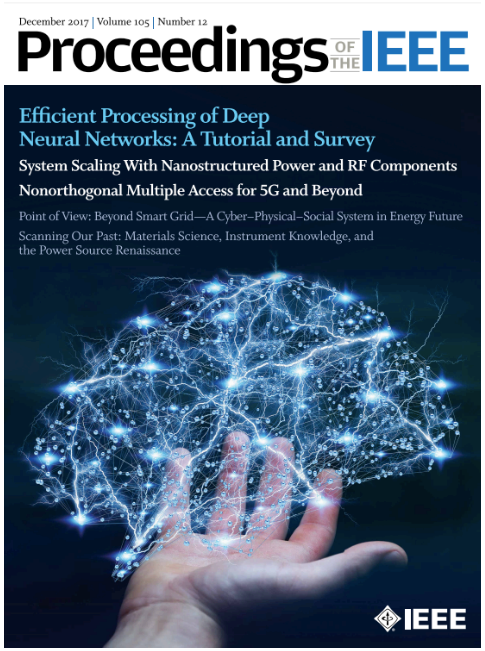


*In collaboration with **Joel Emer**,*
Sertac Karaman, Fangchang Ma, Diana Wofk, Tien-Ju Yang,
Google Mobile Vision Team

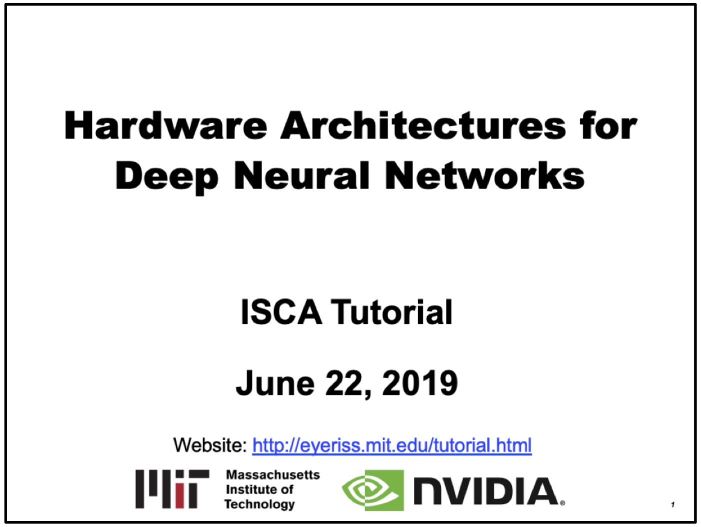
Contact Info
email: sze@mit.edu
website: www.rle.mit.edu/eems

Energy-Efficient Processing of DNNs

A significant amount of algorithm and hardware research on energy-efficient processing of DNNs



V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017



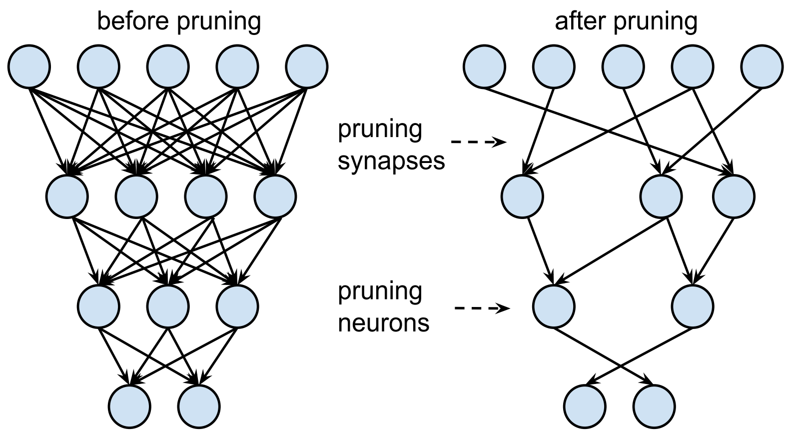
http://eyeriss.mit.edu/tutorial.html

We identified various challenges to existing approaches

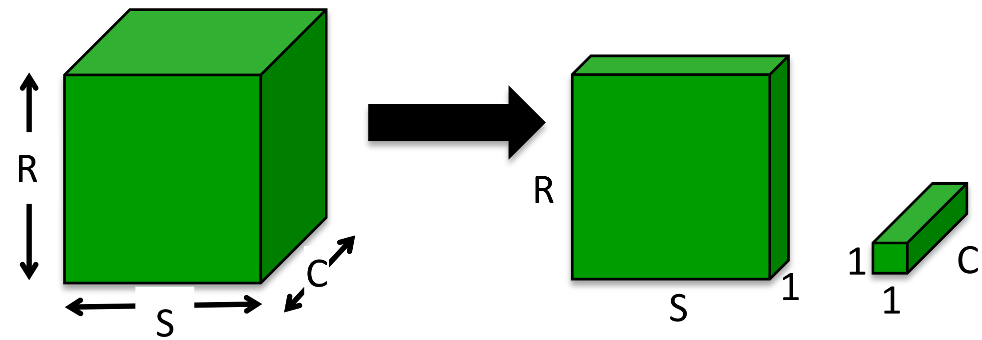
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



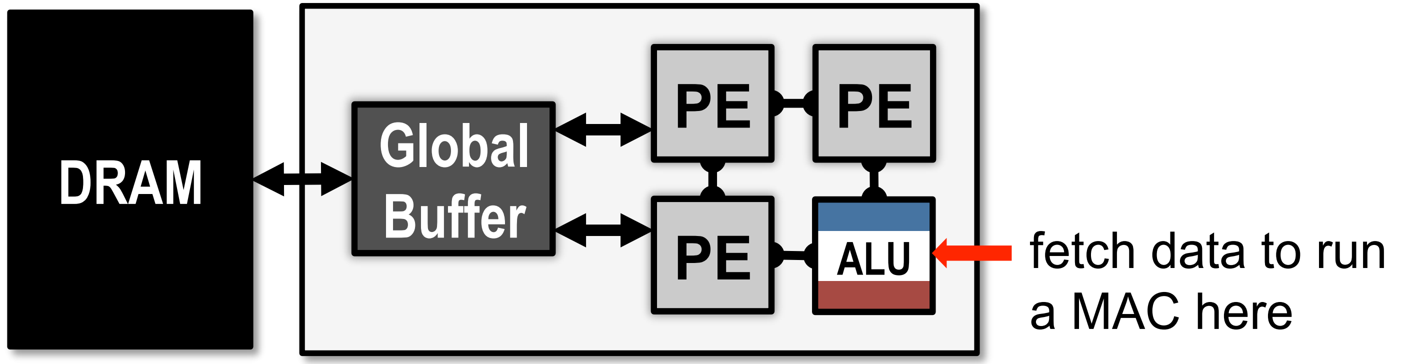
Compact Network Architectures

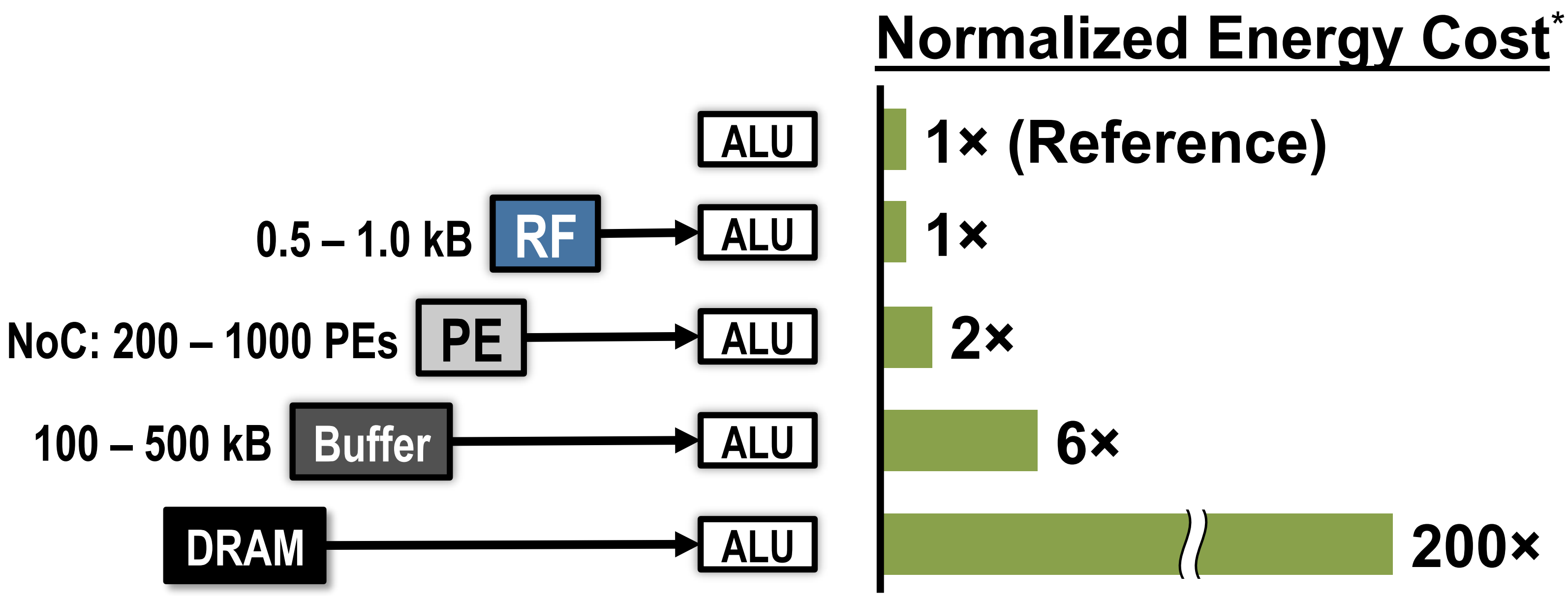


... also reduced precision

- Focus on reducing **number of MACs and weights**
- **Does it translate to energy savings and reduced latency?**

Data Movement is Expensive

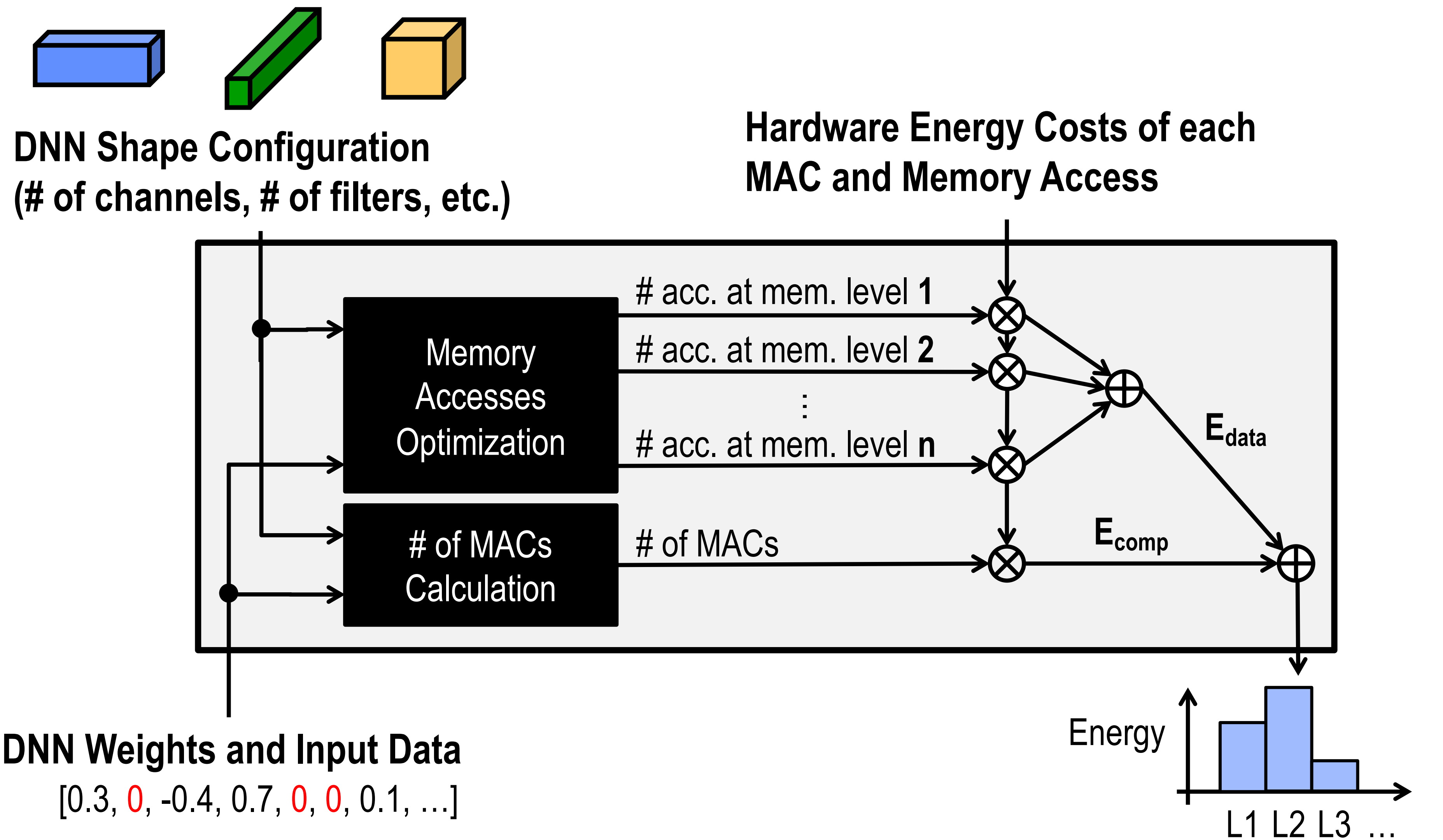




* measured from a commercial 65nm process

Energy of weight depends on **memory hierarchy** and **dataflow**

Energy-Evaluation Methodology



DNN Energy Consumption

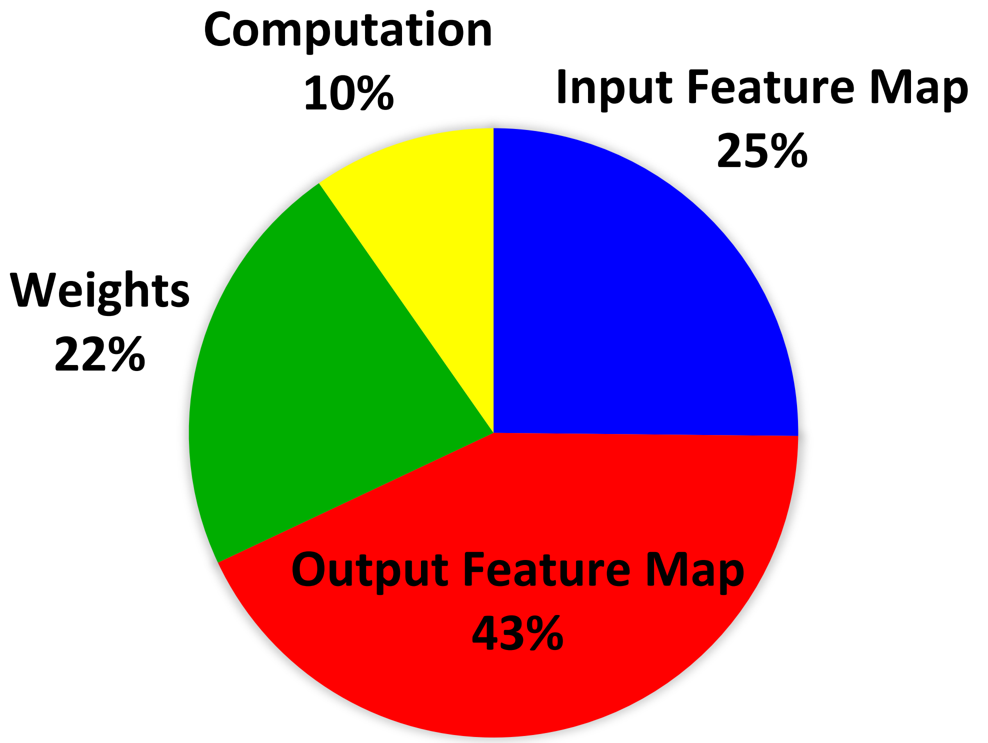
Tool available at: https://energyestimation.mit.edu/

[Yang et al., CVPR 2017]

Key Observations

- Number of weights ***alone*** is not a good metric for energy
- **All data types** should be considered

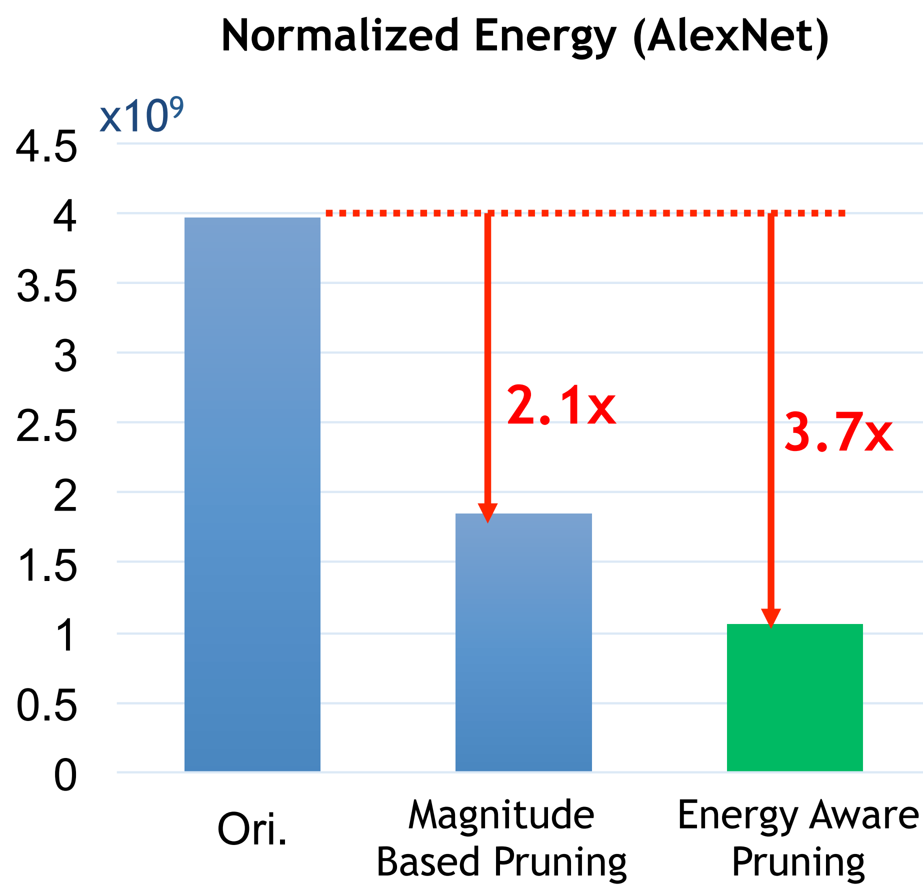
Energy Consumption of GoogLeNet



Energy-Aware Pruning

Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

- Sort layers based on energy and prune layers that consume most energy first
- EAP reduces AlexNet energy by **3.7x** and outperforms the previous work that uses magnitude-based pruning by **1.7x**

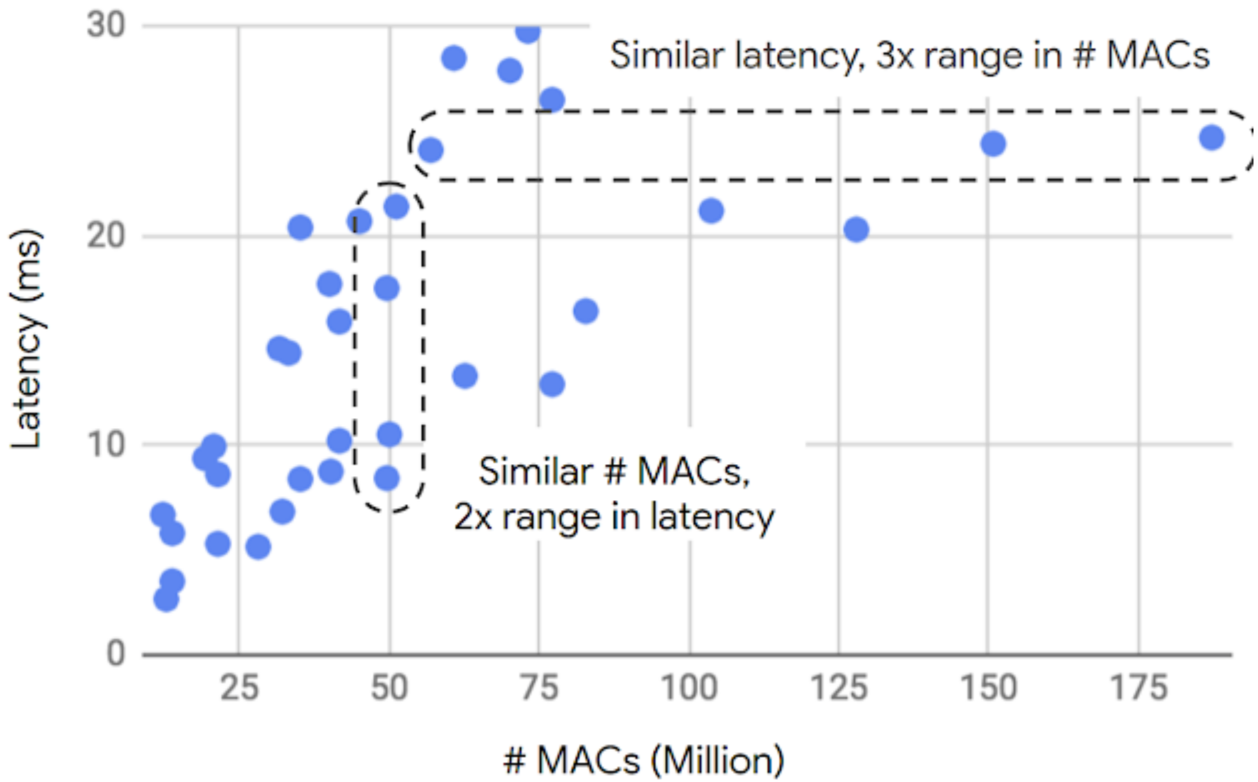


Pruned models available at
http://eyeriss.mit.edu/energy.html

[Yang et al., CVPR 2017]

of Operations vs. Latency

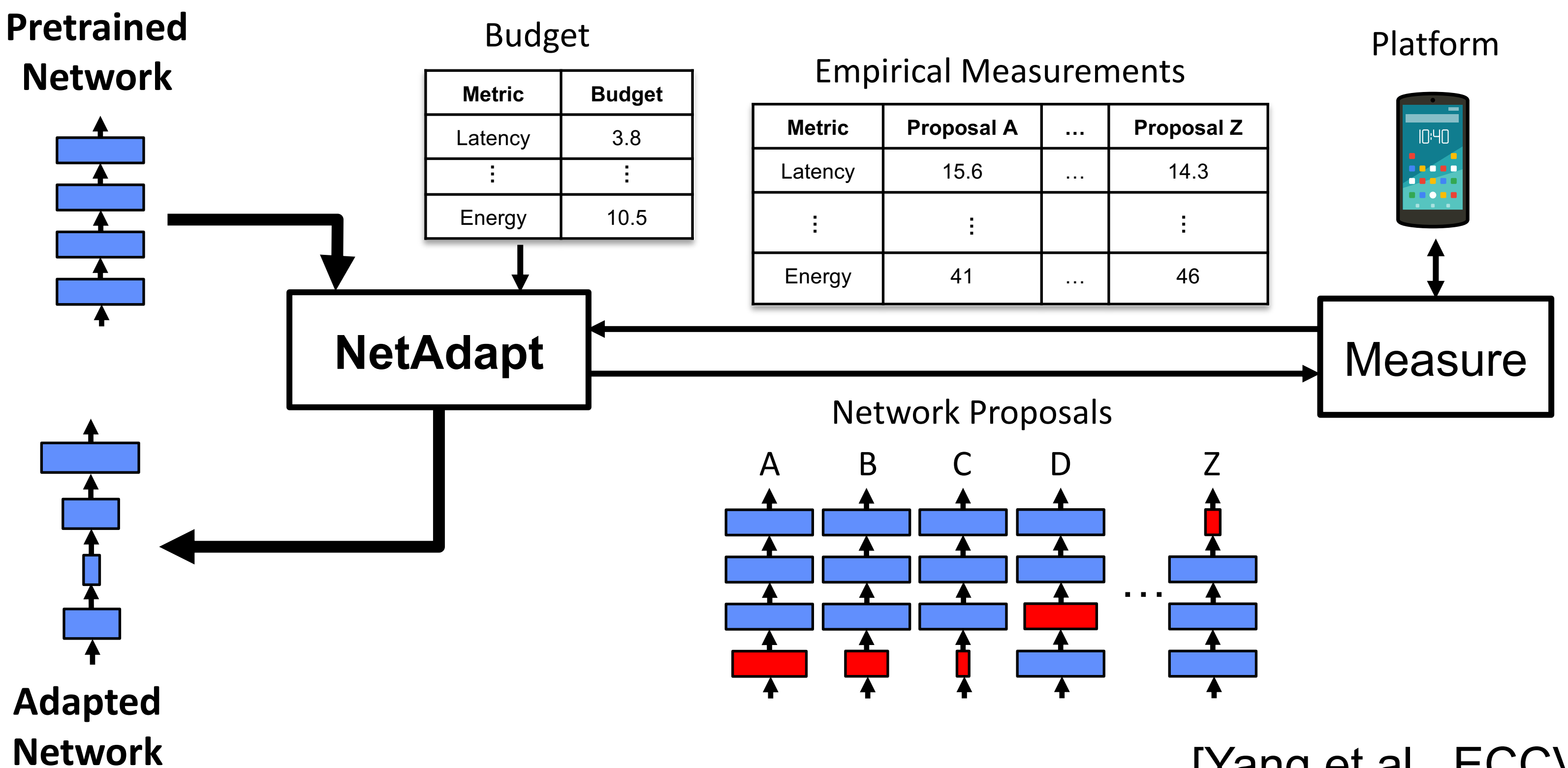
- # of operations (MACs) does not approximate latency well



Source: Google (https://ai.googleblog.com/2018/04/introducing-cvpr-2018-on-device-visual.html)

NetAdapt: Platform-Aware DNN Adaptation

- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)



Metric	Budget
Latency	3.8
⋮	⋮
Energy	10.5

Metric	Proposal A	...	Proposal Z
Latency	15.6	...	14.3
⋮	⋮		⋮
Energy	41	...	46

[Yang et al., ECCV 2018]

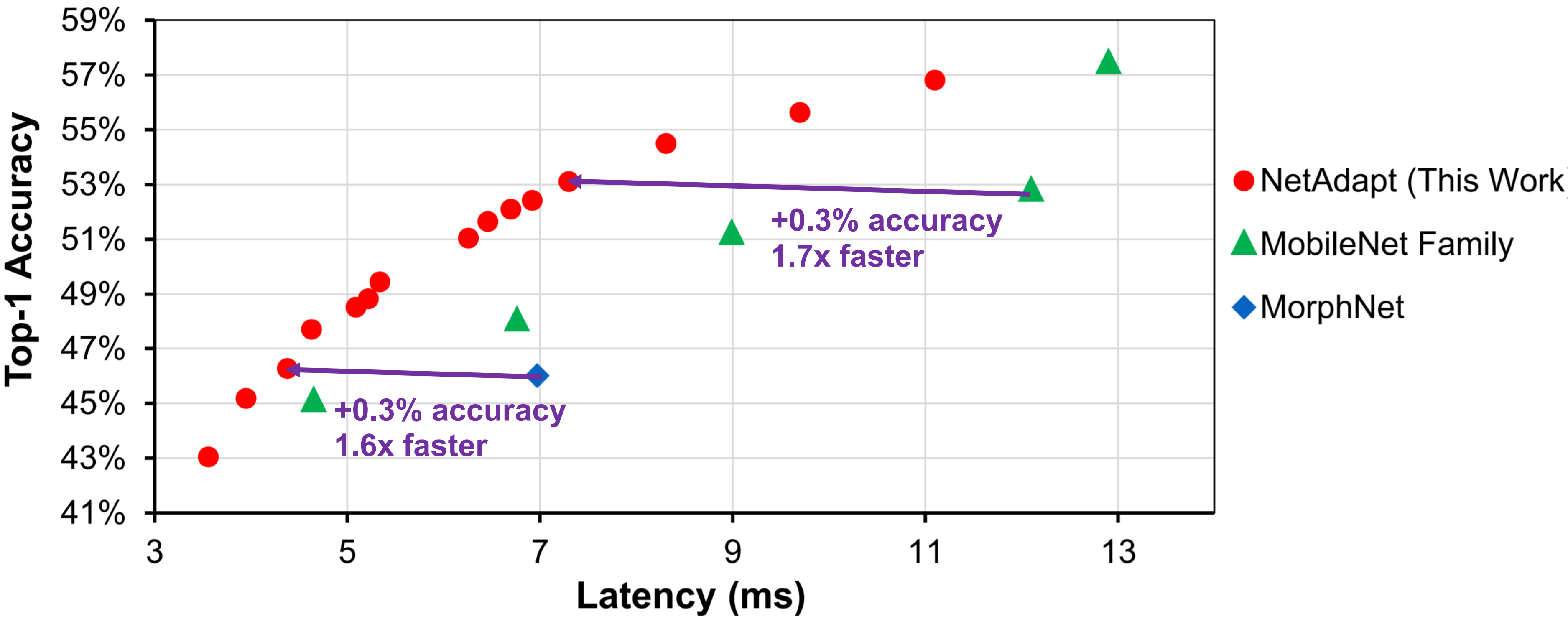
In collaboration with Google's Mobile Vision Team

RESEARCH LABORATORY OF ELECTRONICS AT MIT

MTL microsystems technology laboratories massachusetts institute of technology

Improved Latency vs. Accuracy Tradeoff

- NetAdapt boosts **the real inference speed** of MobileNet by up to 1.7x with higher accuracy



*Tested on the ImageNet dataset and a Google Pixel 1 CPU

Reference:
MobileNet: Howard et al, “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, arXiv 2017
MorphNet: Gordon et al., “Morphnet: Fast & simple resource-constrained structure learning of deep networks”, CVPR 2018

Problem Formulation

$$\max_{Net} Accuracy(Net)\ \ subject\ to\ Resource_j(Net) \leq Budget_j, j = 1, \cdots, m$$

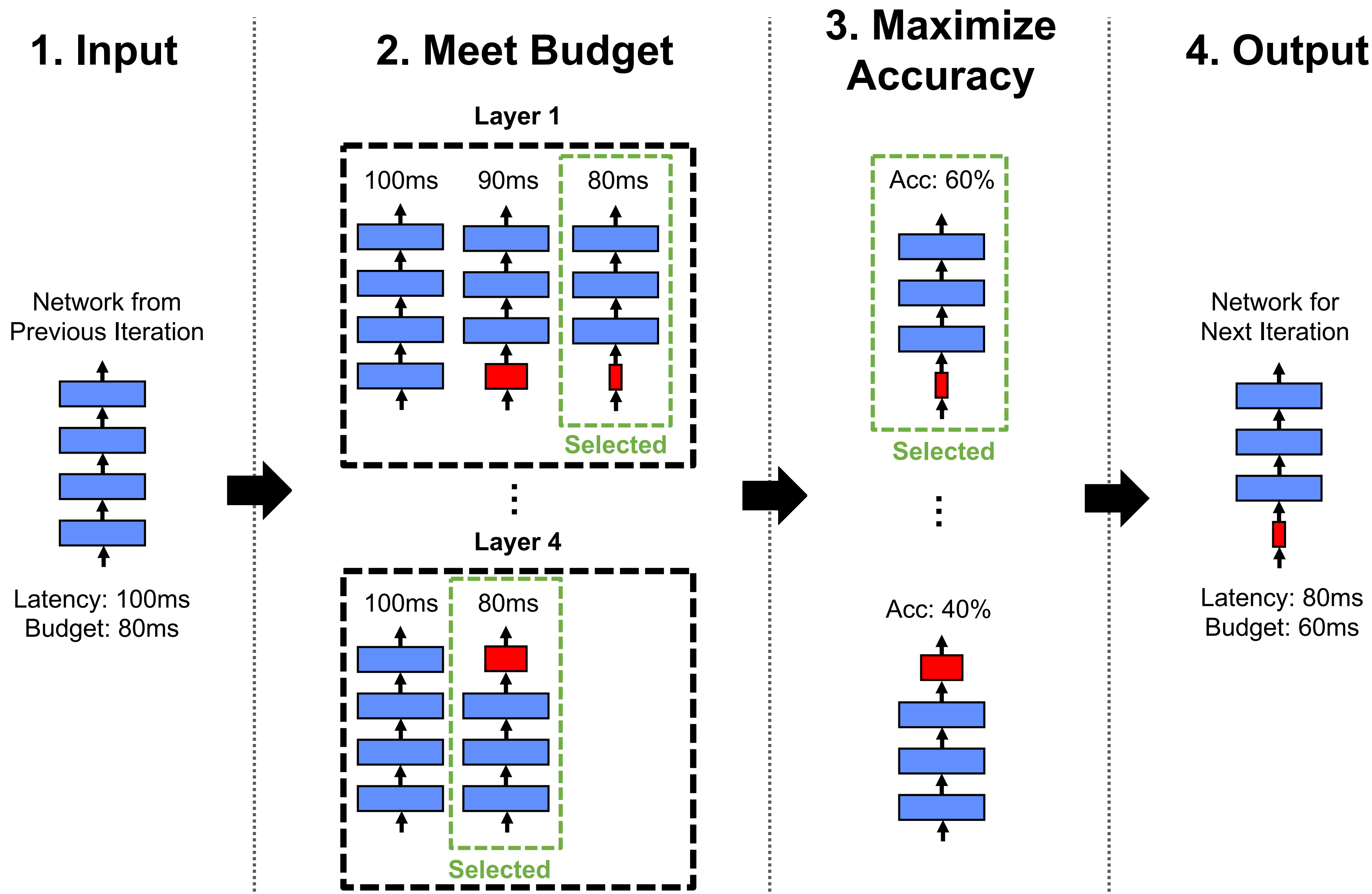
Break into a set of simpler problems and solve iteratively

$$\max_{Net_i} Acc(Net_i)\ \ subject\ to\ Res_j(Net_i) \leq Res_j(Net_{i-1}) - \Delta R_{i,j}, j = 1, \cdots, m$$

**Acc*: accuracy function, *Res*: resource evaluation function,
***ΔR*: resource reduction**, *Bud*: given budget
Budget incrementally tightens $\boldsymbol{Res_j(Net_{i-1}) - \Delta R_{i,j}}$

- **Advantages**
  - Supports multiple resource budgets at the same time
  - Guarantees that the budgets will be satisfied because the resource consumption decreases monotonically
  - Generates a family of networks (from each iteration) with different resource versus accuracy trade-offs
  - Intuitive and can easily set one additional hyperparameter ($\boldsymbol{\Delta R_{i,j}}$)

Simplified Example of One Iteration



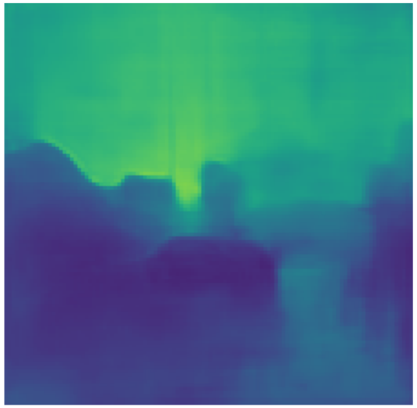
Code to be released at http://netadapt.mit.edu

FastDepth: Fast Monocular Depth Estimation

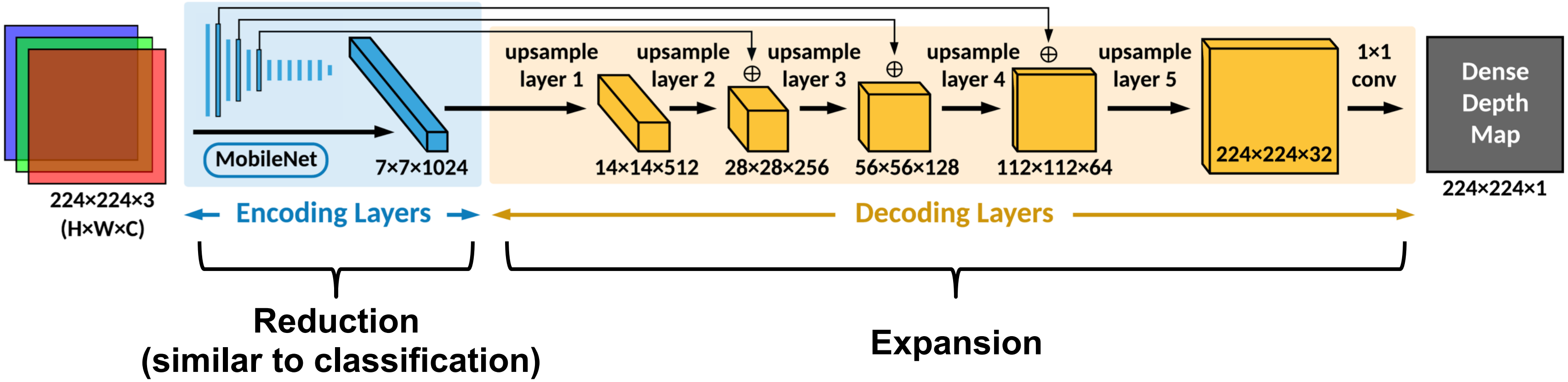
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction



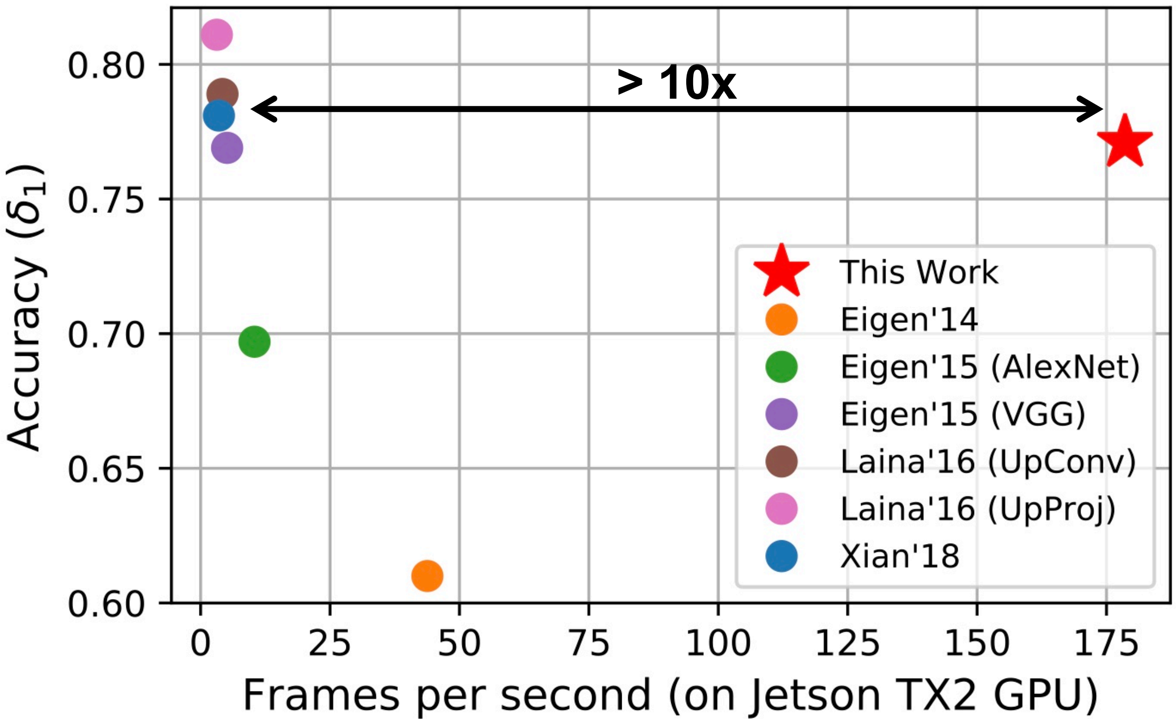
Auto Encoder DNN Architecture (Dense Output)



[Joint work with Sertac Karaman]

FastDepth: Fast Monocular Depth Estimation

Apply *NetAdapt*, *compact network design*, and *depth wise decomposition* to decoder layer to enable depth estimation at **high frame rates on an embedded platform** while still maintaining accuracy



Configuration: Batch size of one (32-bit float)

~40fps on an iPhone

Models available at *http://fastdepth.mit.edu*

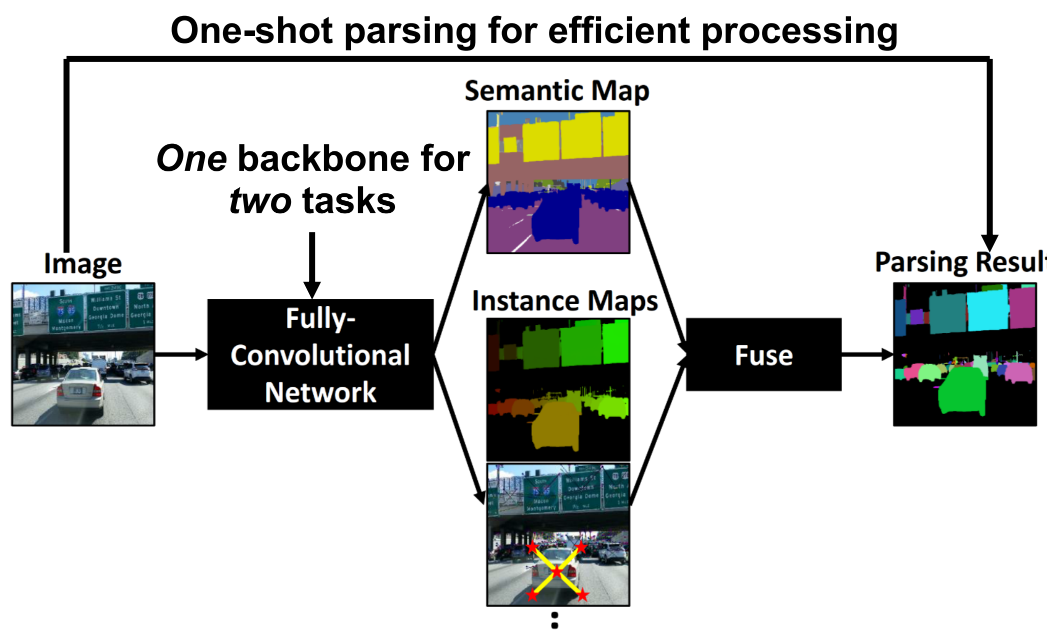
[Wofk*, Ma* et al., ICRA 2019]

DeeperLab: Single-Shot Image Parser

Results from Xception

Joint Semantic and Instance Segmentation (high resolution input image)

Fully convolutional, one-shot parsing
(bottom-up approach)



http://deeperlab.mit.edu/

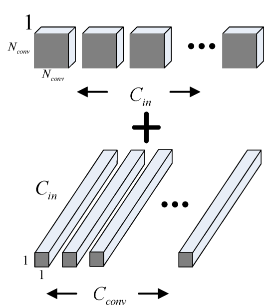
[Yang et al., arXiv 2019]

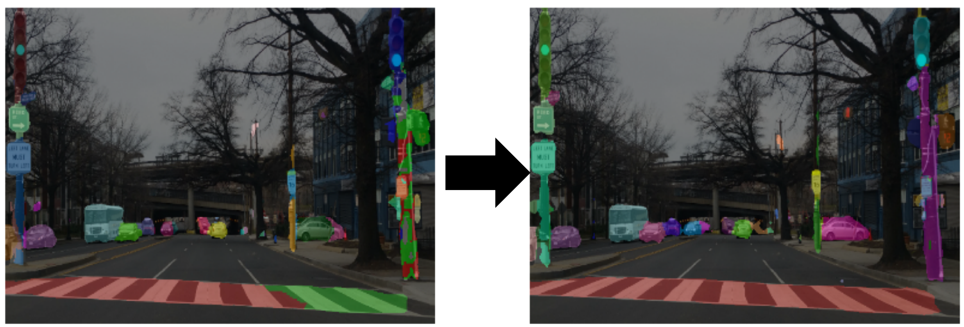
DeeperLab: Efficient Image Parsing

Address memory requirement for large feature map

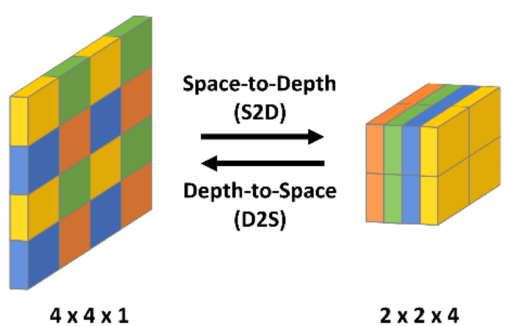
1 Wide MobileNet: Increase kernel size rather than depth

3x3 → 5x5





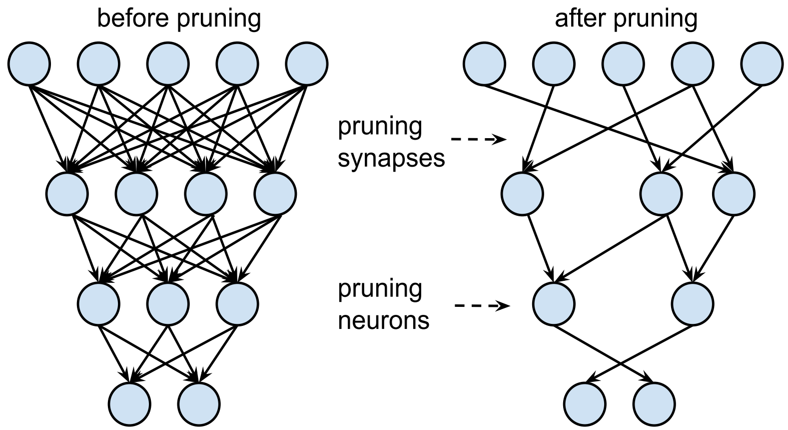
2 Space-to-depth/depth-to-space: Avoid upsampling



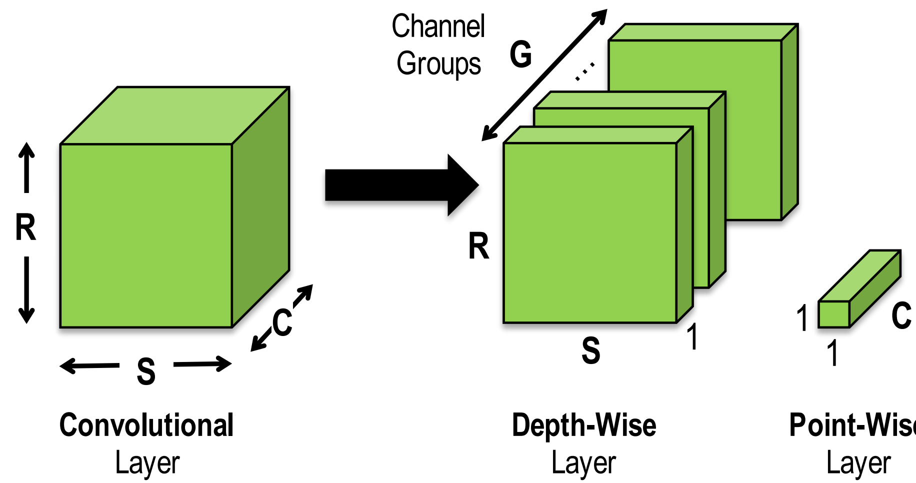
Achieves near real-time 6.19 fps on GPU (V100) with 25.2% PQ and 49.8% PC on Mapillary Vistas dataset

Many Efficient DNN Design Approaches

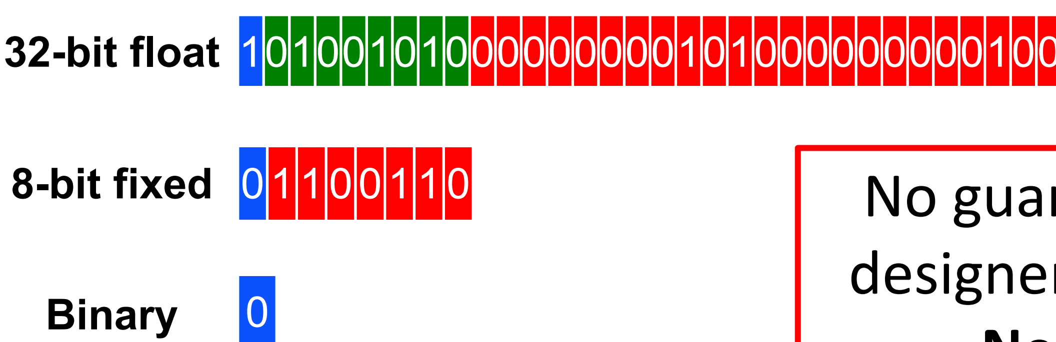
Network Pruning



Compact Network Architectures



Reduce Precision

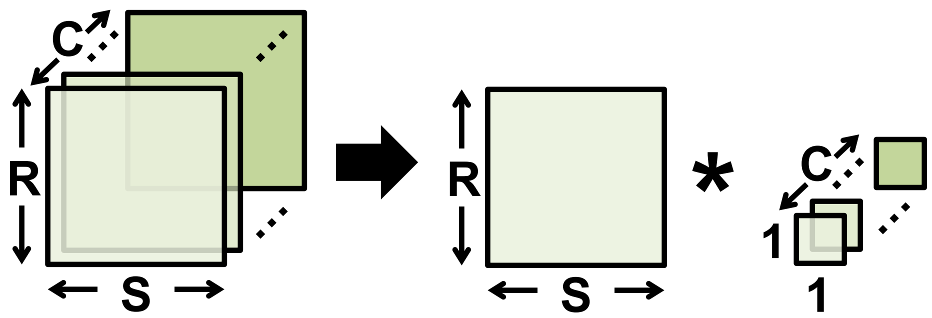


No guarantee that DNN algorithm designer will use a given approach.
Need flexible hardware!

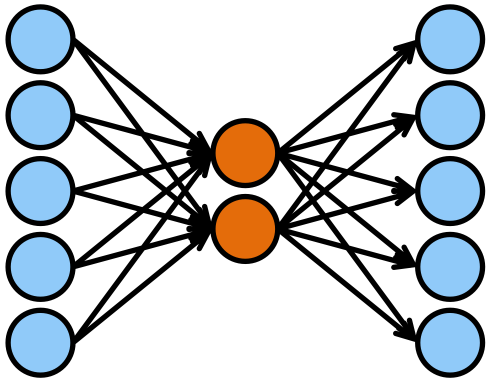
[Chen et al., SysML 2018]

DNNs are Becoming More Compact!

Filter Decomposition



Bottleneck Layer

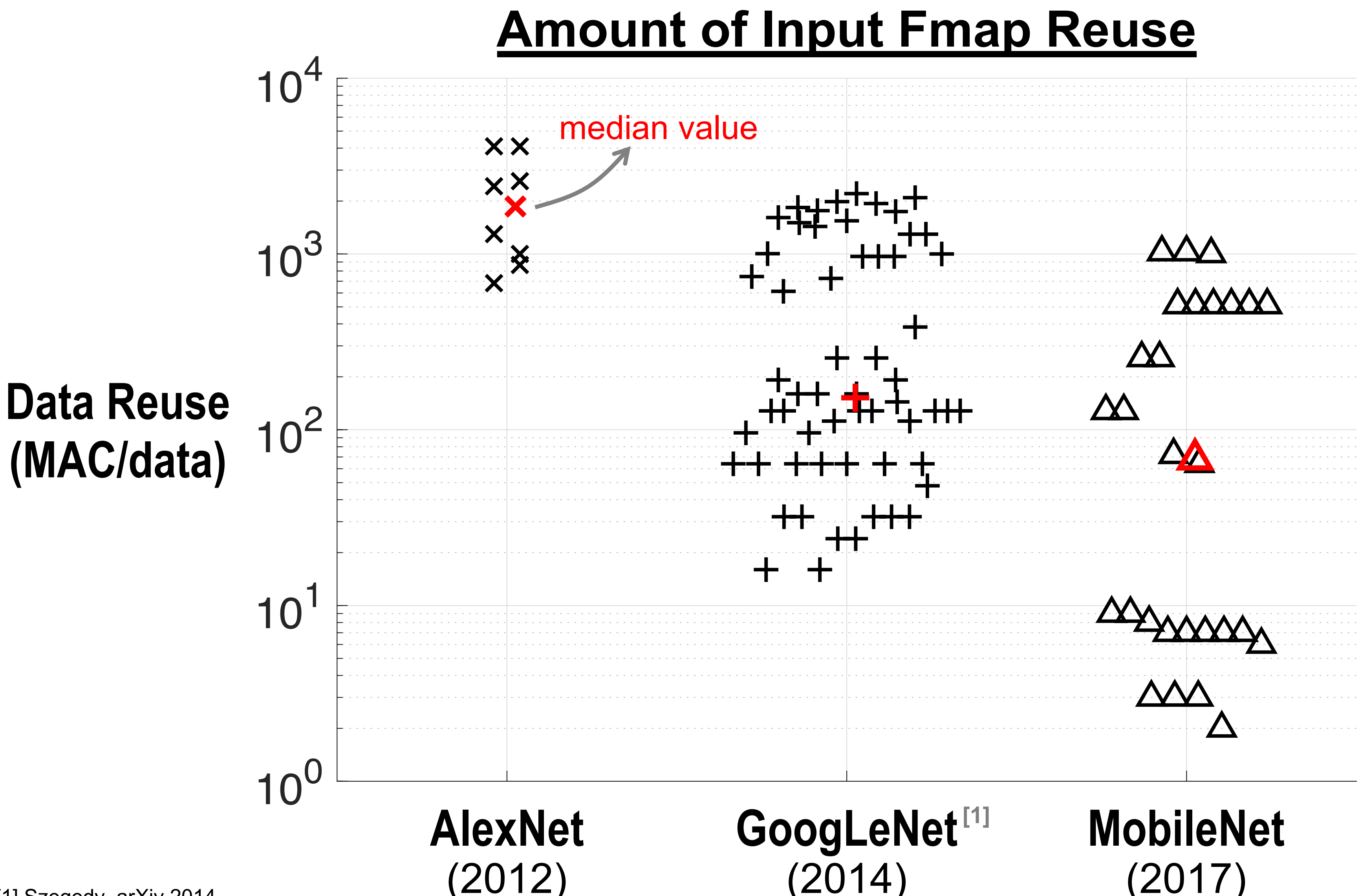


	Year	Accuracy*	# Layers	# Weights	# MACs
AlexNet	2012	80.4%	8	**61M**	**724M**
MobileNet [1]	2017	**89.5%**	28	**4M**	**569M**

* ImageNet Classification Top-5

[1] Howard, *arXiv* 2017

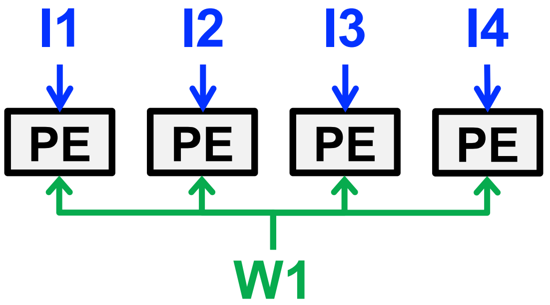
Data Reuse Going Against Our Favor



[1] Szegedy, arXiv 2014

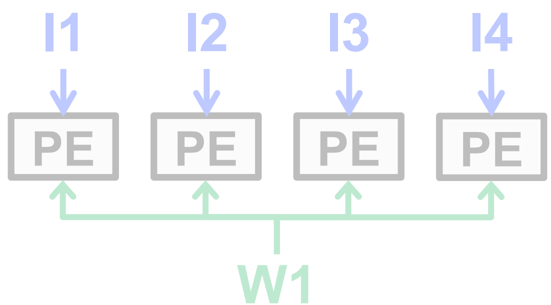
How Does Reuse Affect Performance?

Example: reuse the same **Weight** with different **Inputs**



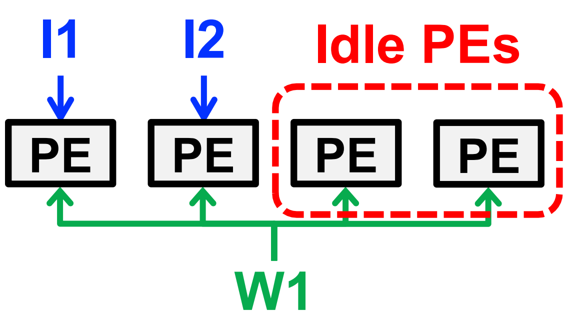
How Does Reuse Affect Performance?

Example: reuse the same **Weight** with different **Inputs**



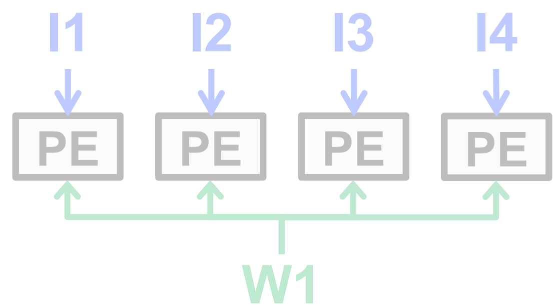
If weight reuse is low, **performance will go down!**

Case 1: Dataflow not flexible enough



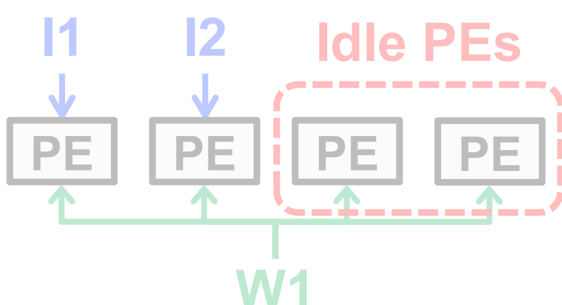
How Does Reuse Affect Performance?

Example: reuse the same **Weight** with different **Inputs**

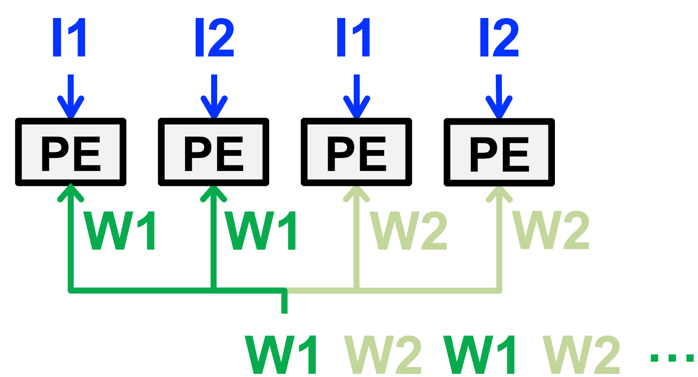


If weight reuse is low, performance will go down!

Case 1: Dataflow not flexible enough



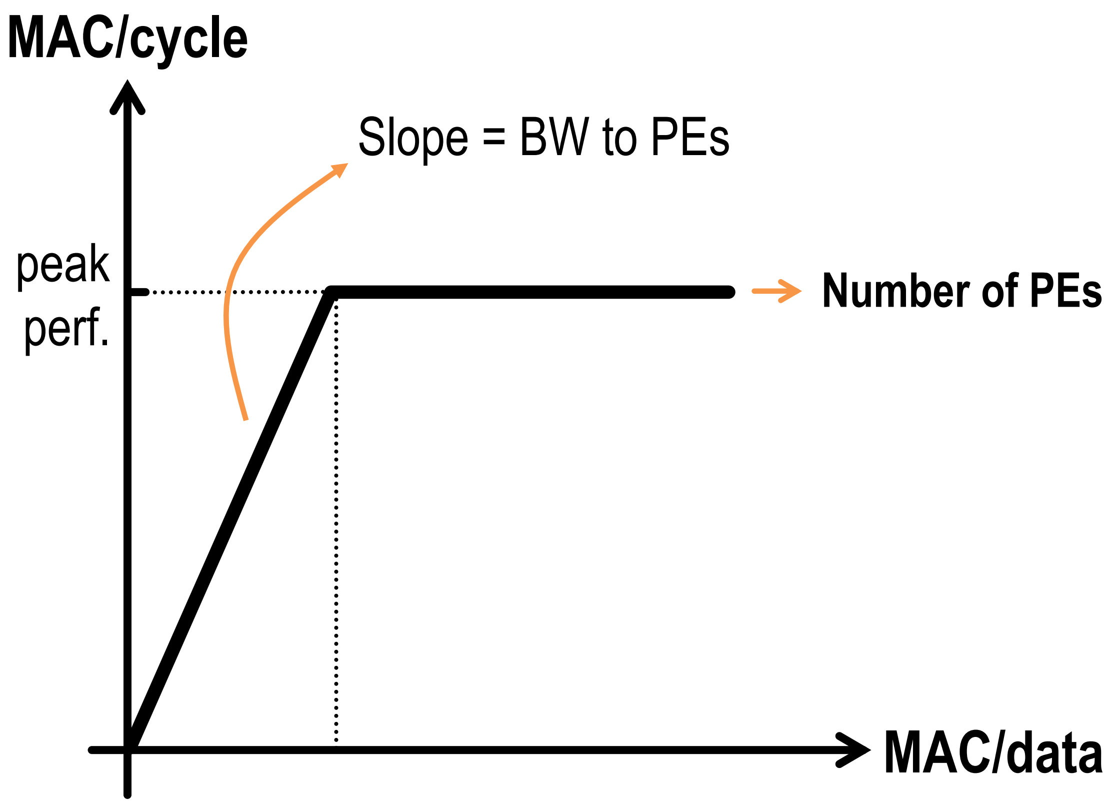
Case 2: Insufficient NoC bandwidth



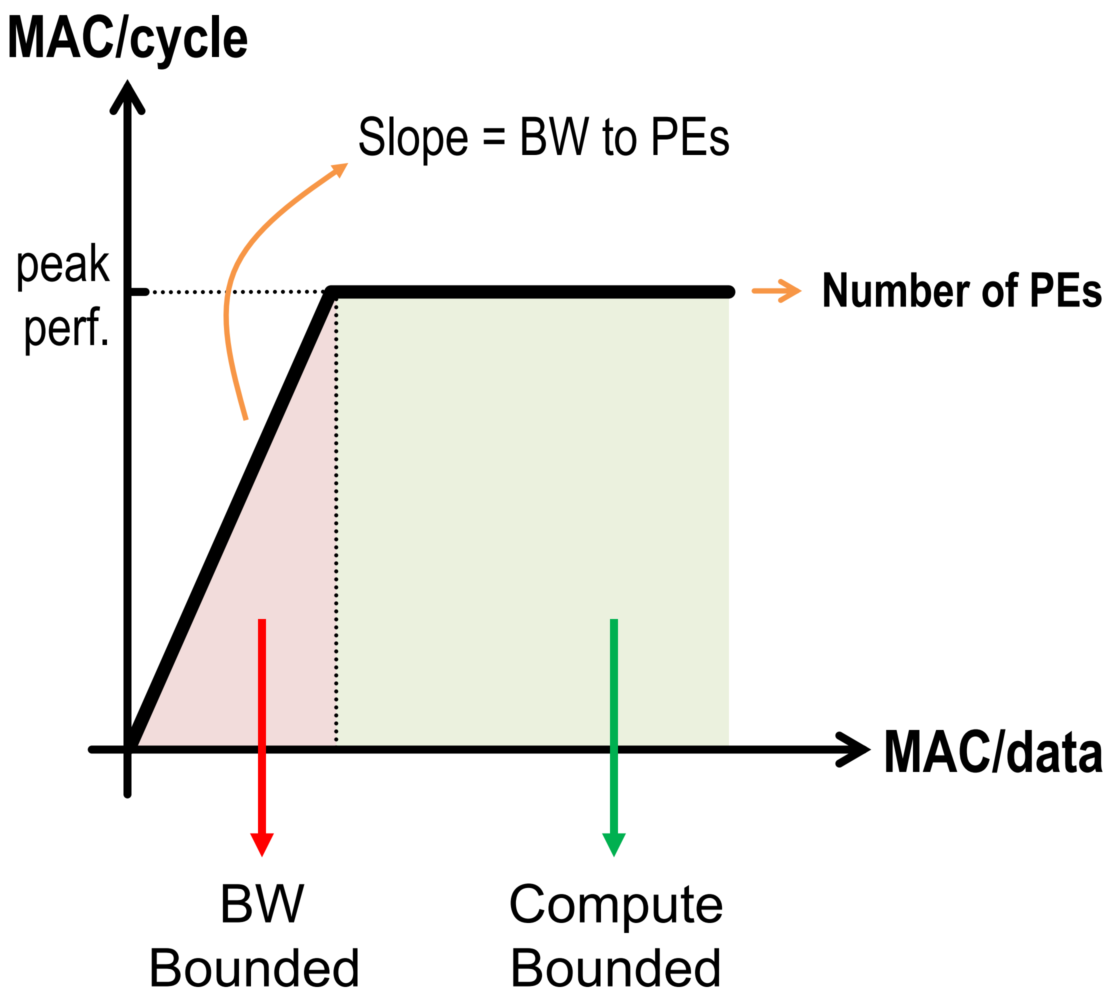
Eyexam
Performance Eval Framework

- A systematic way to quickly understand the performance limits of DNN accelerators in a step-by-step process

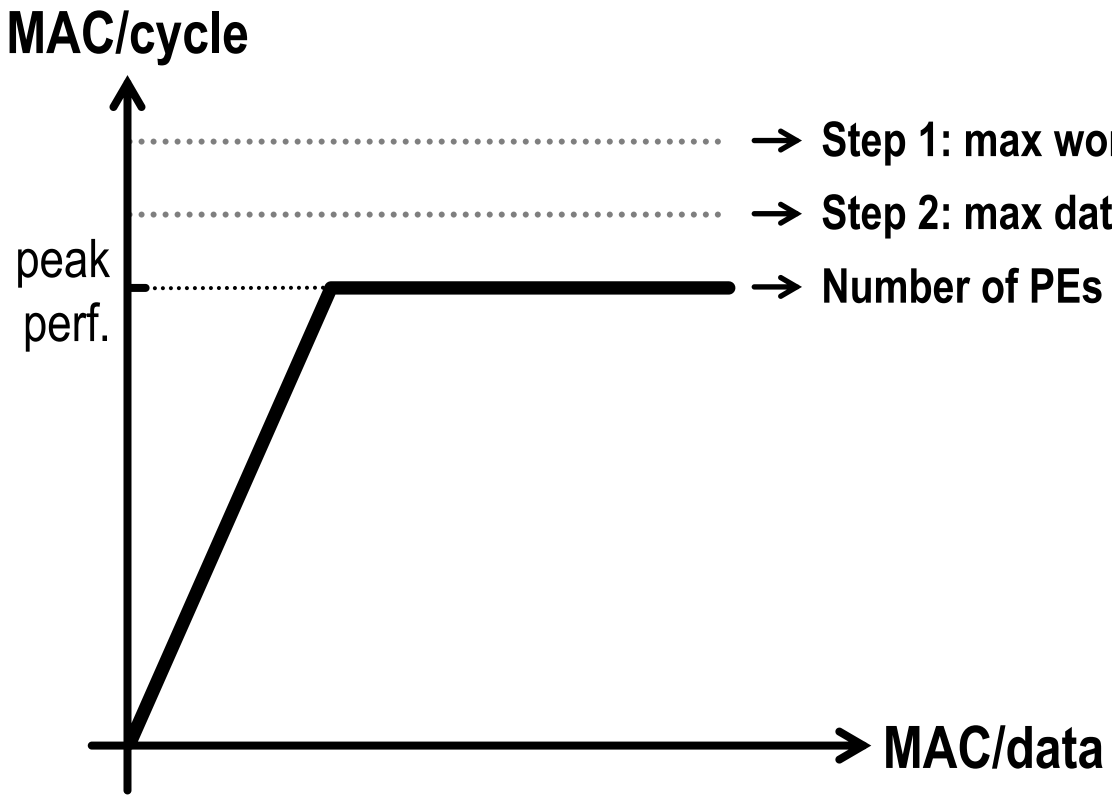
Eyexam: Performance Eval Framework



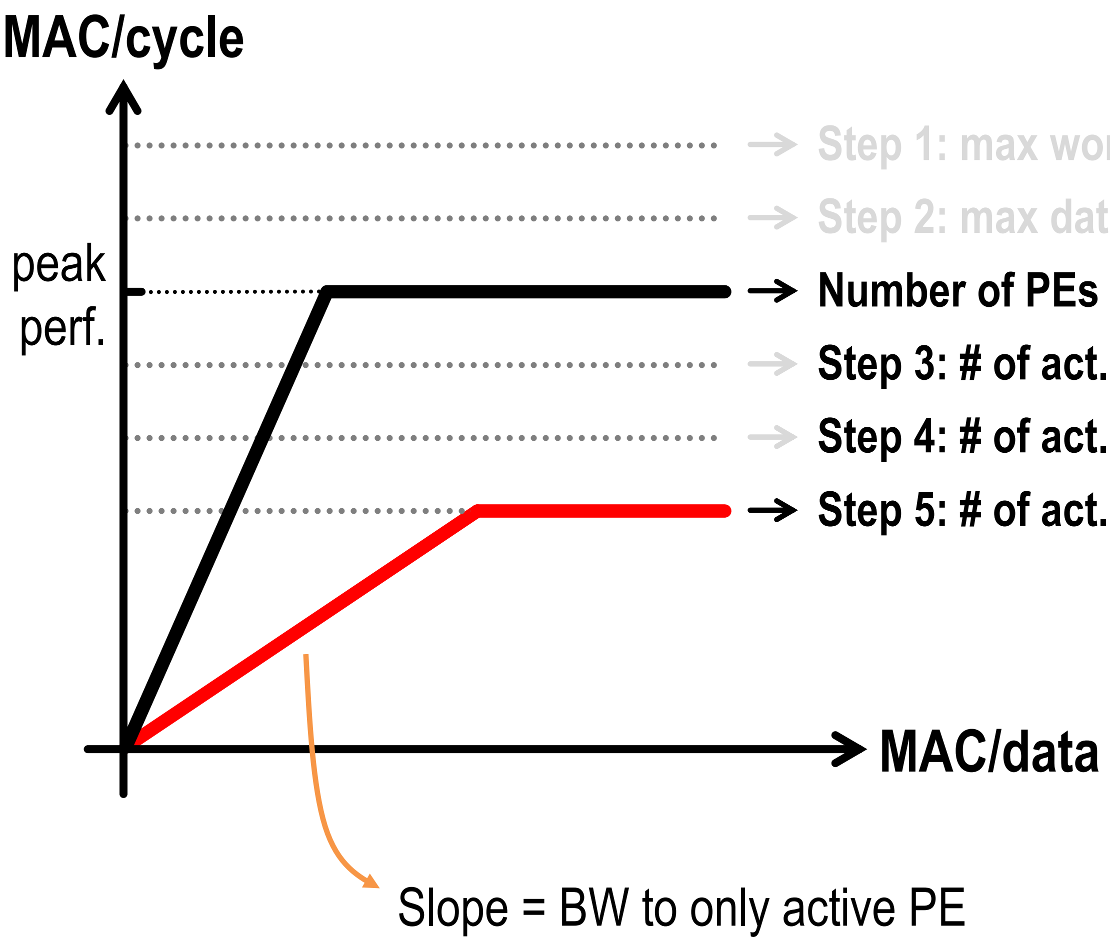
Eyexam: Performance Eval Framework

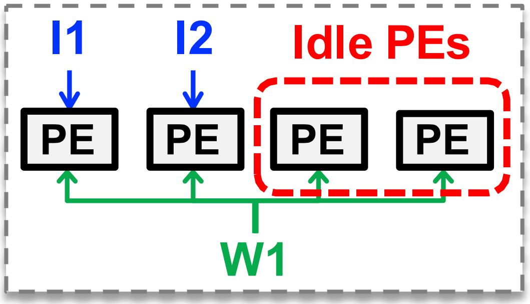


Eyexam: Performance Eval Framework

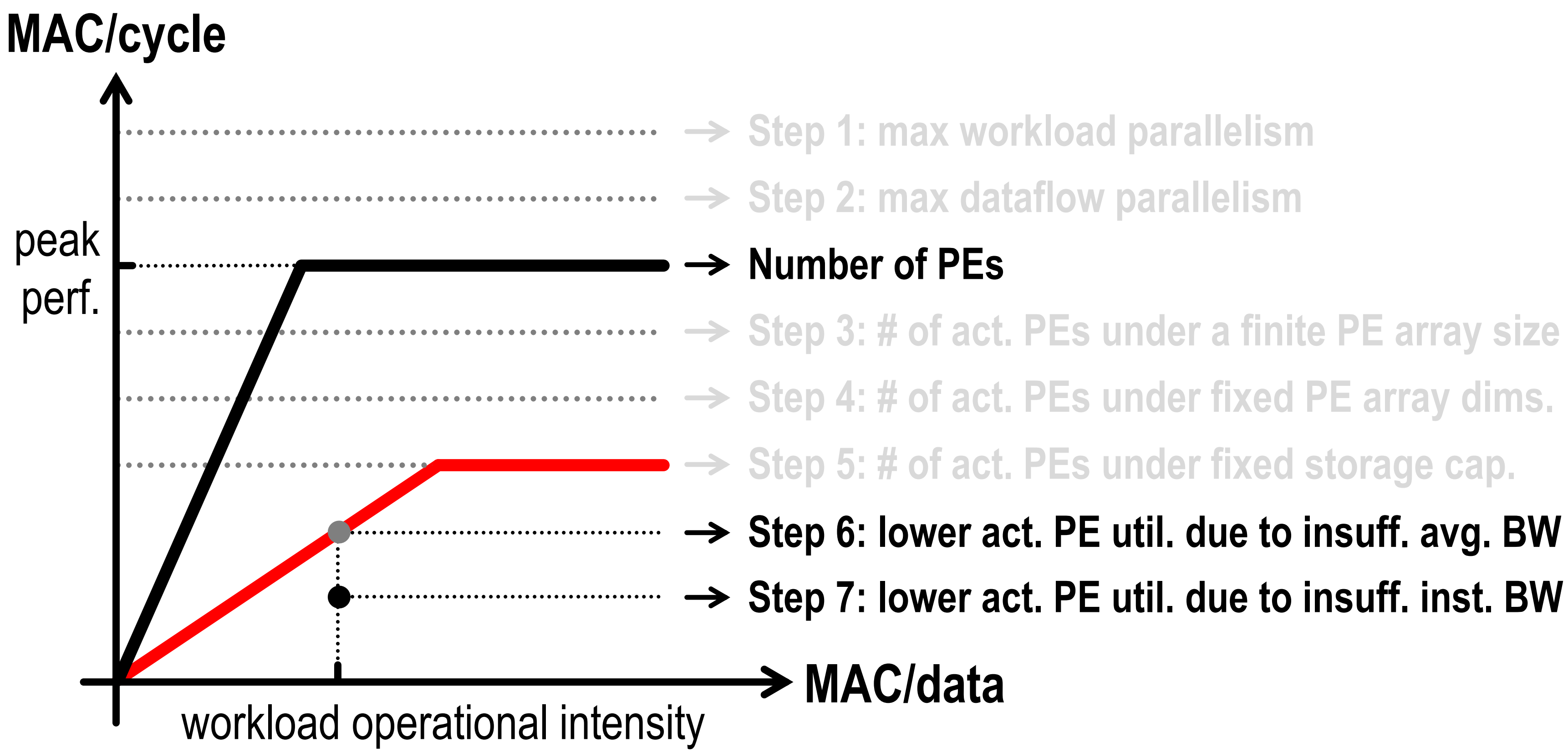


Eyexam: Performance Eval Framework

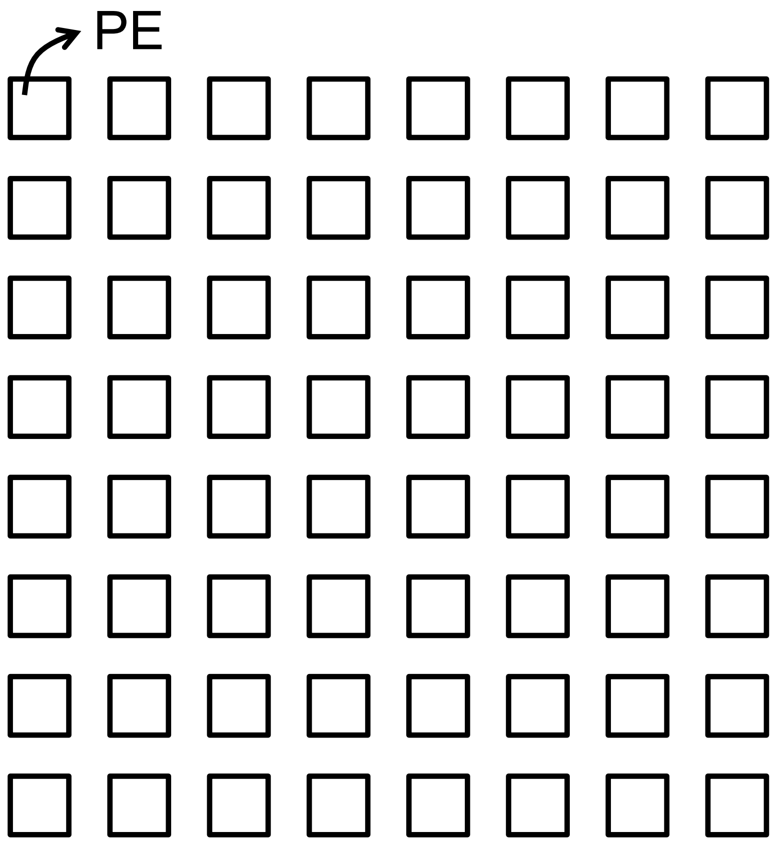




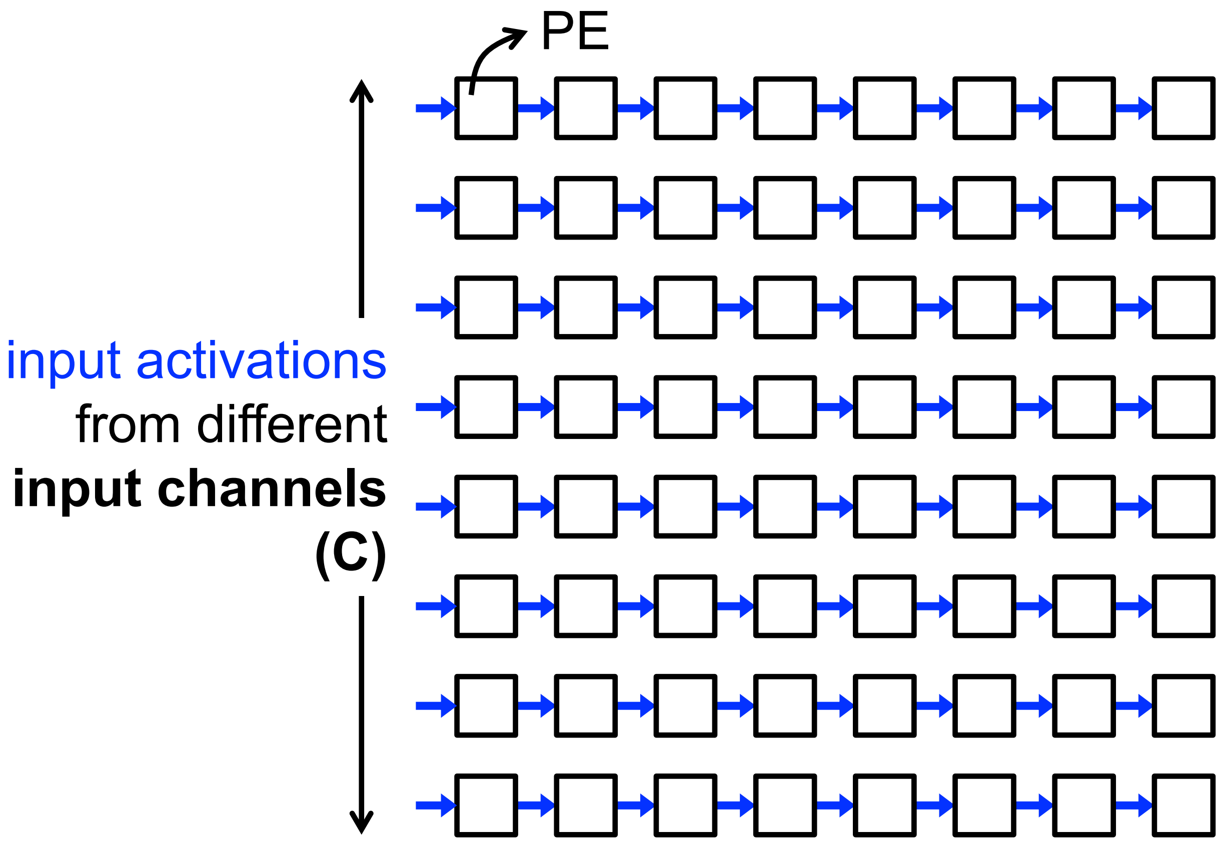
Eyexam: Performance Eval Framework



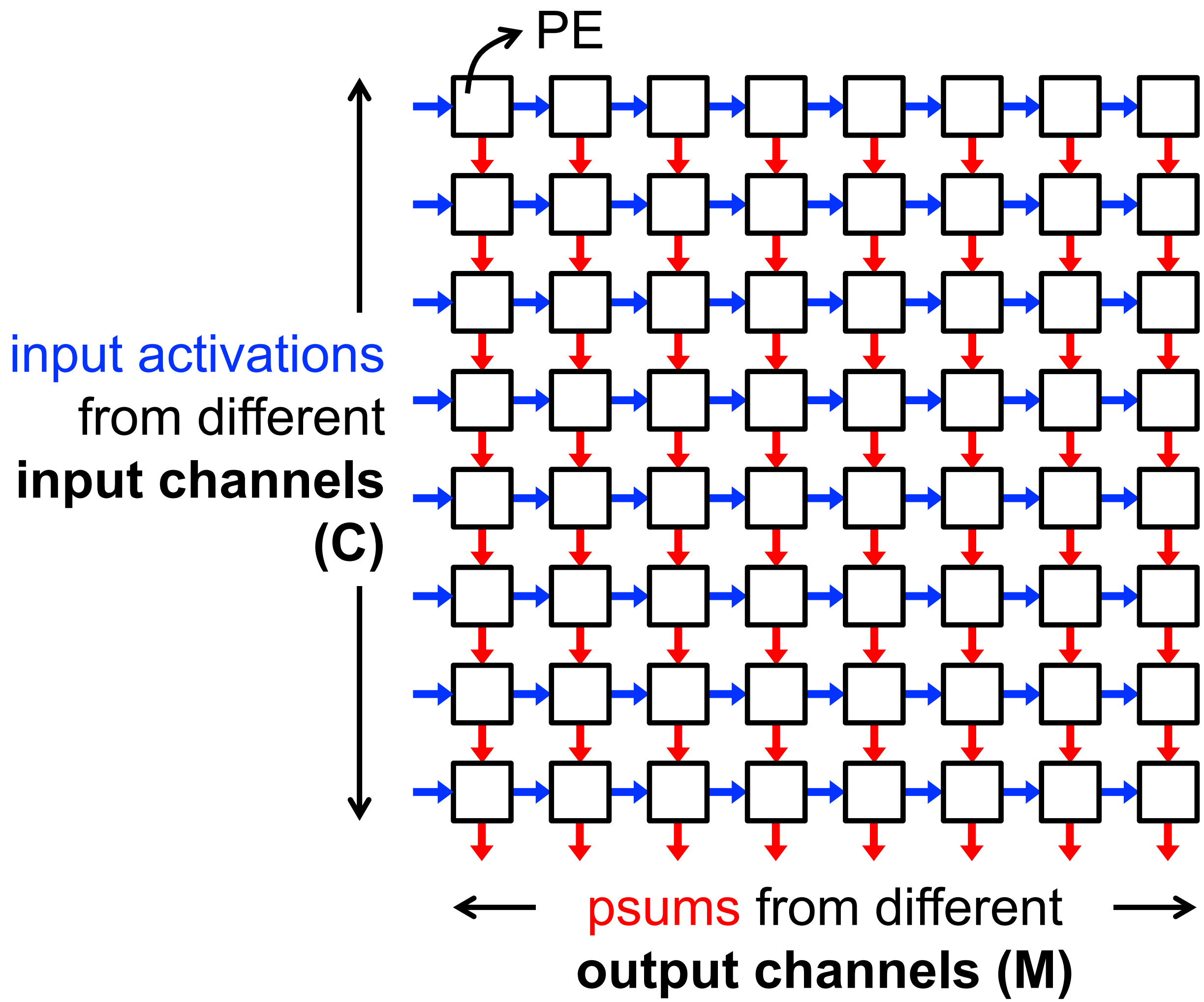
Example: A Common Design Pattern



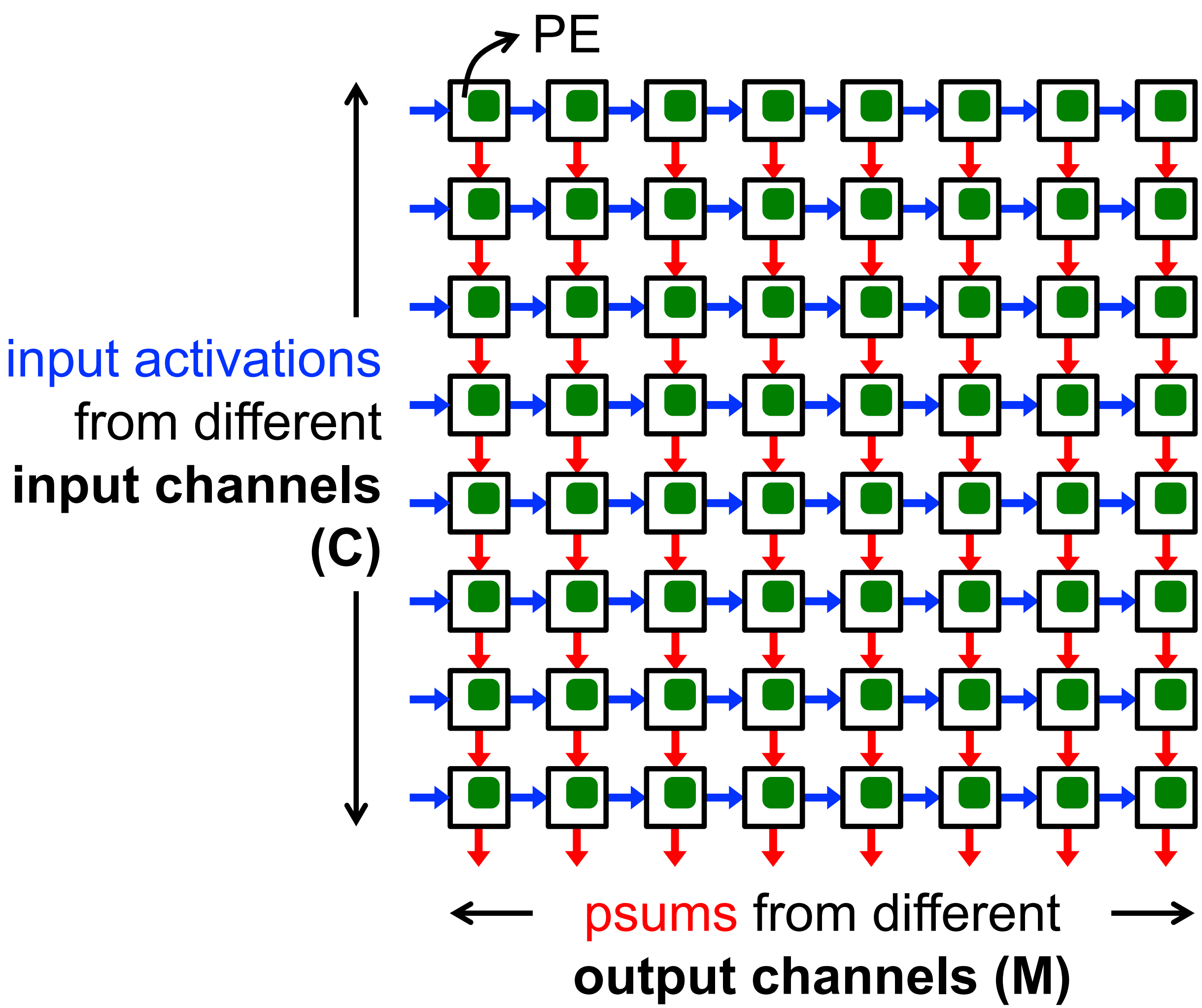
Example: A Common Design Pattern



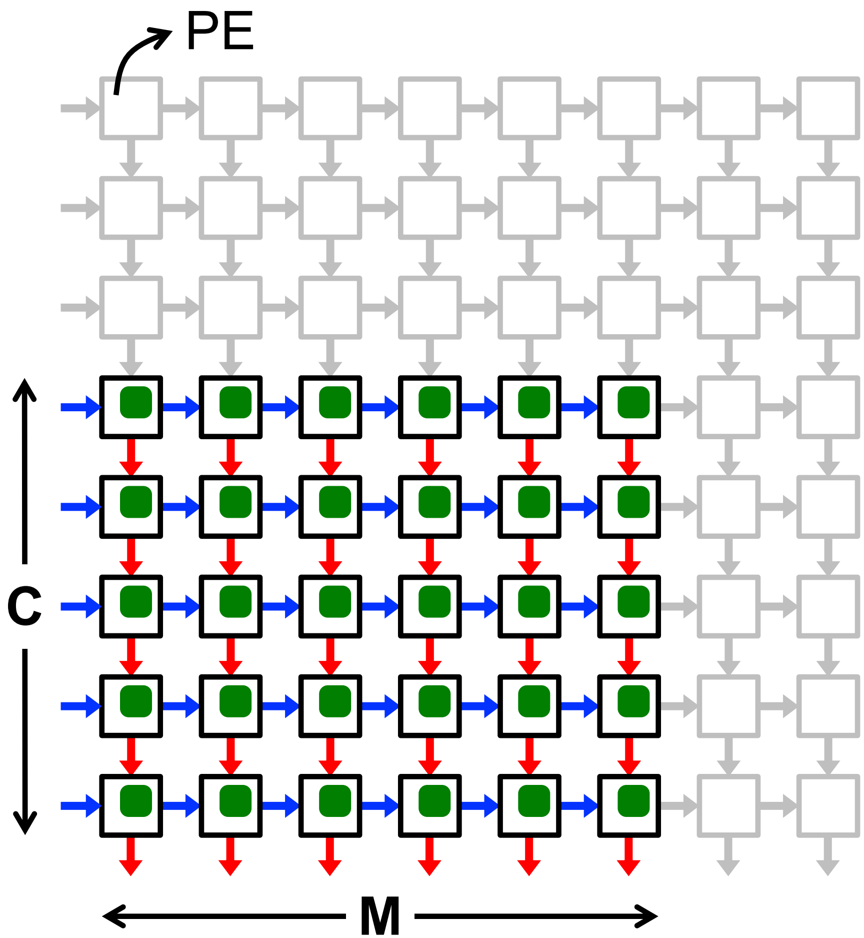
Example: A Common Design Pattern



Example: A Common Design Pattern



Example: A Common Design Pattern

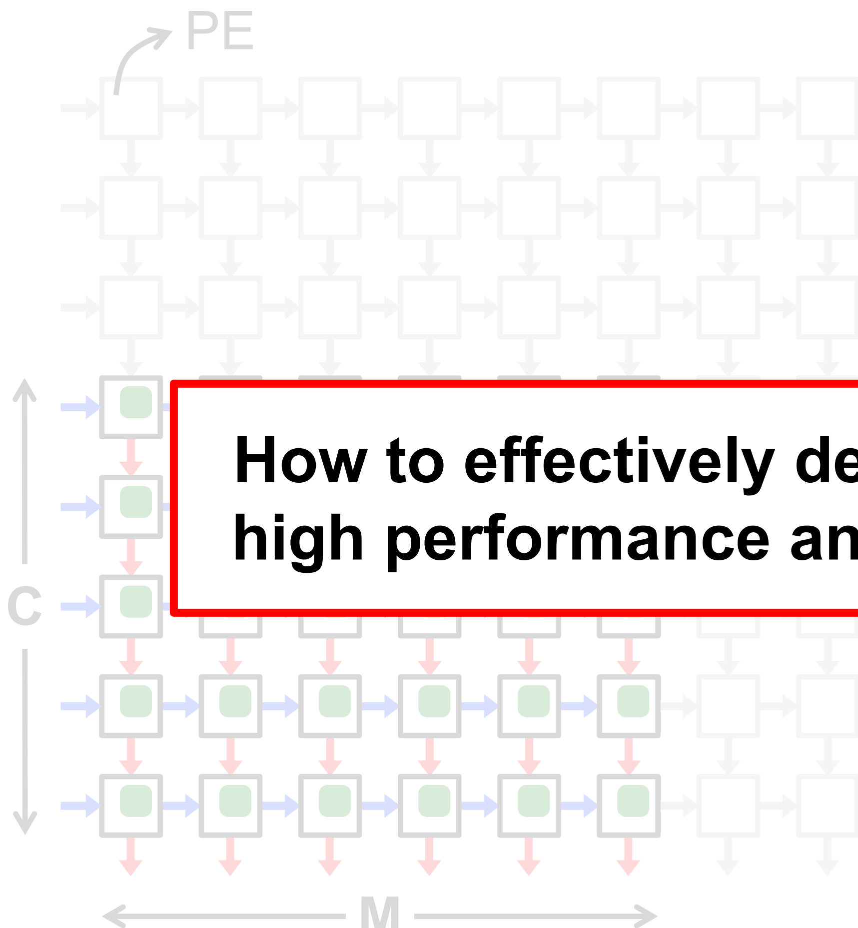


1. PE array **underutilized** If there are fewer input and/or output channels than the array dimensions

2. Effective data delivery BW also becomes lower → further impact performance

3. Not scalable →utilization will be worse at larger scales

Example: A Common Design Pattern



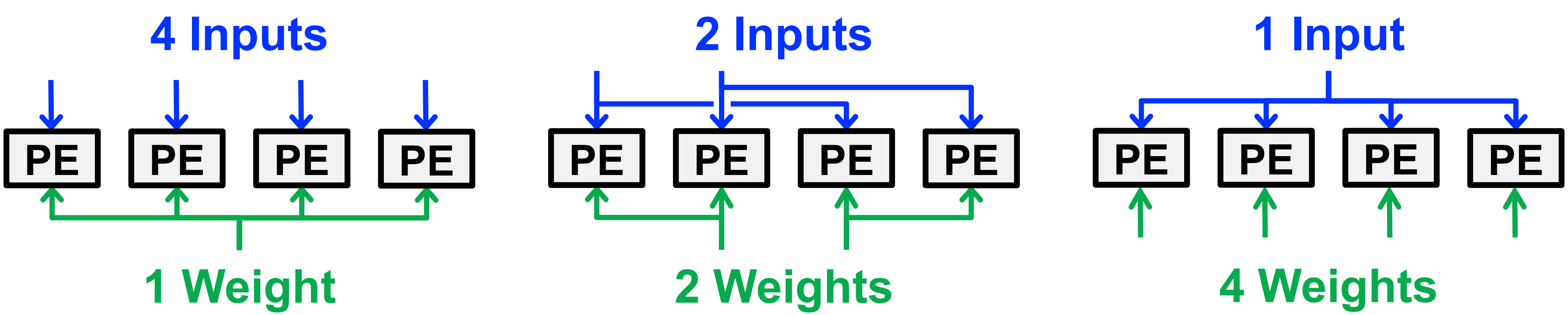
1. PE array **underutilized** if there are fewer input and/or output channels than the array dimensions

3. Not scalable → utilization will be worse at larger scales

How to effectively deliver data to achieve high performance and energy efficiency?

A More Flexible Data Delivery Strategy

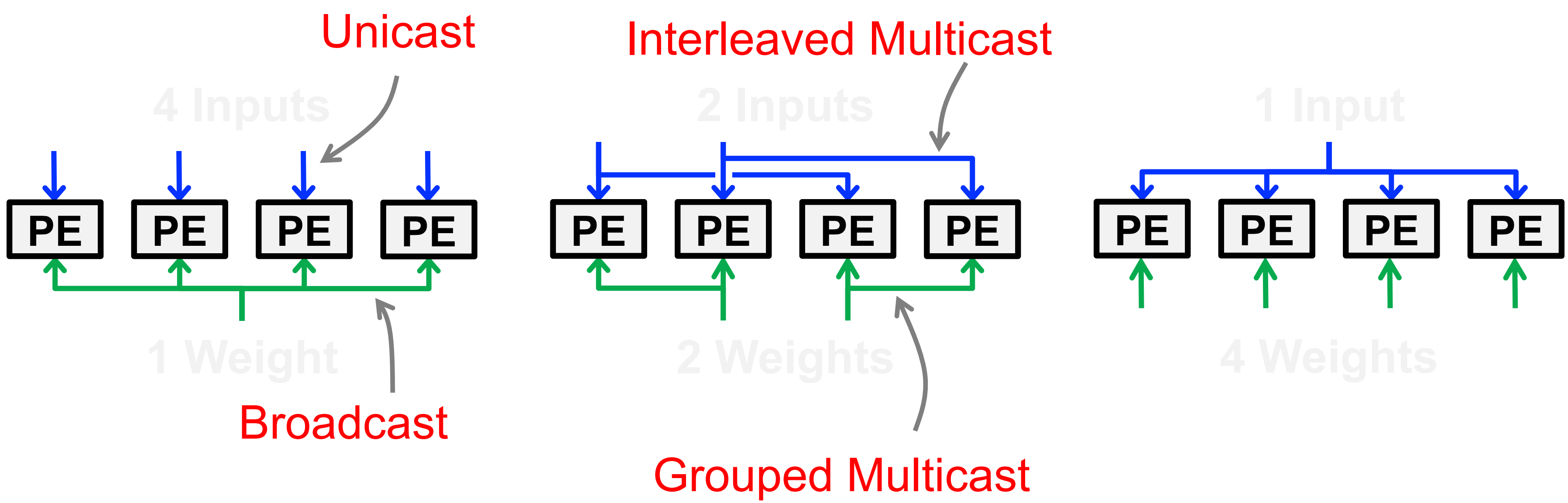
Adapt to the reuse and bandwidth requirements



A More Flexible Mapping Strategy

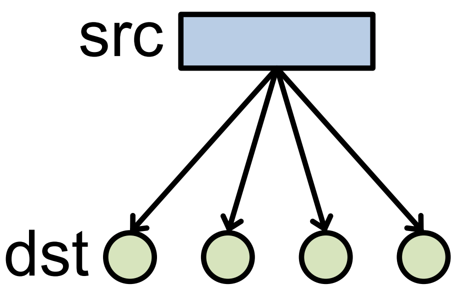
Adapt to the reuse and bandwidth requirements

4 Data Delivery Patterns



On-Chip Network (NoC) is the Bottleneck

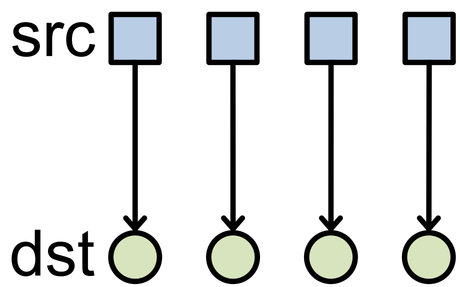
Broadcast Network
(Eyeriss v1 NoC)



High Reuse

Low Bandwidth

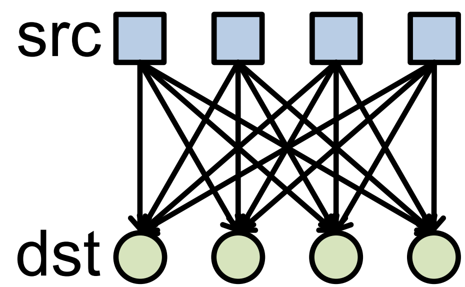
Unicast Networks



Low Reuse

High Bandwidth

All-to-All Networks

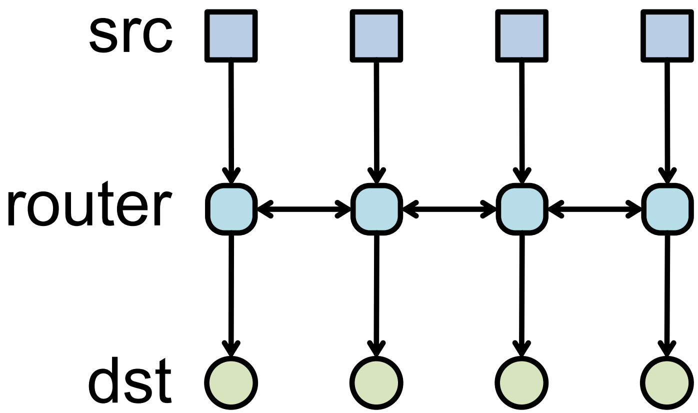


High Reuse

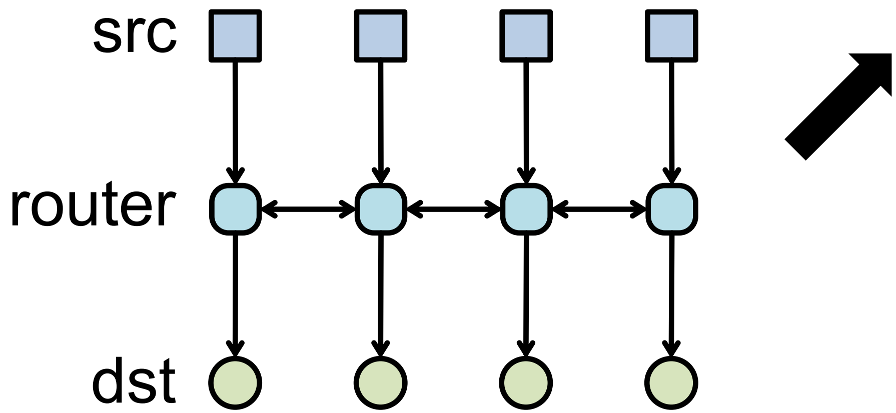
High Bandwidth

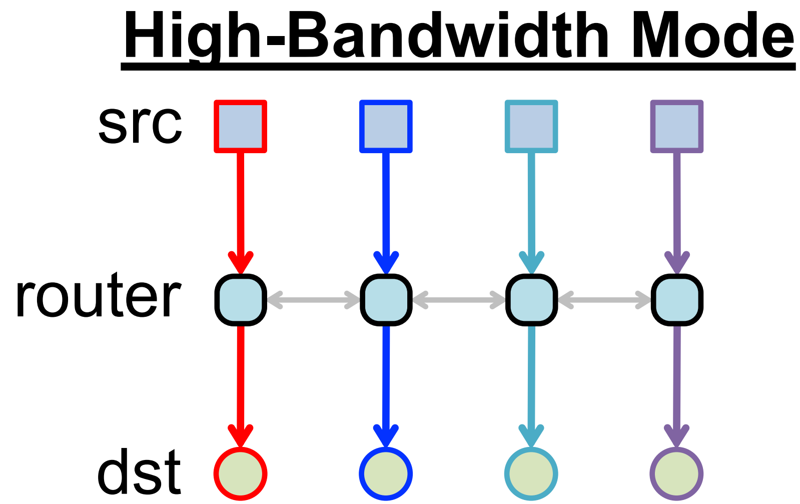
Hard to Scale

Mesh Network – Best of Both Worlds

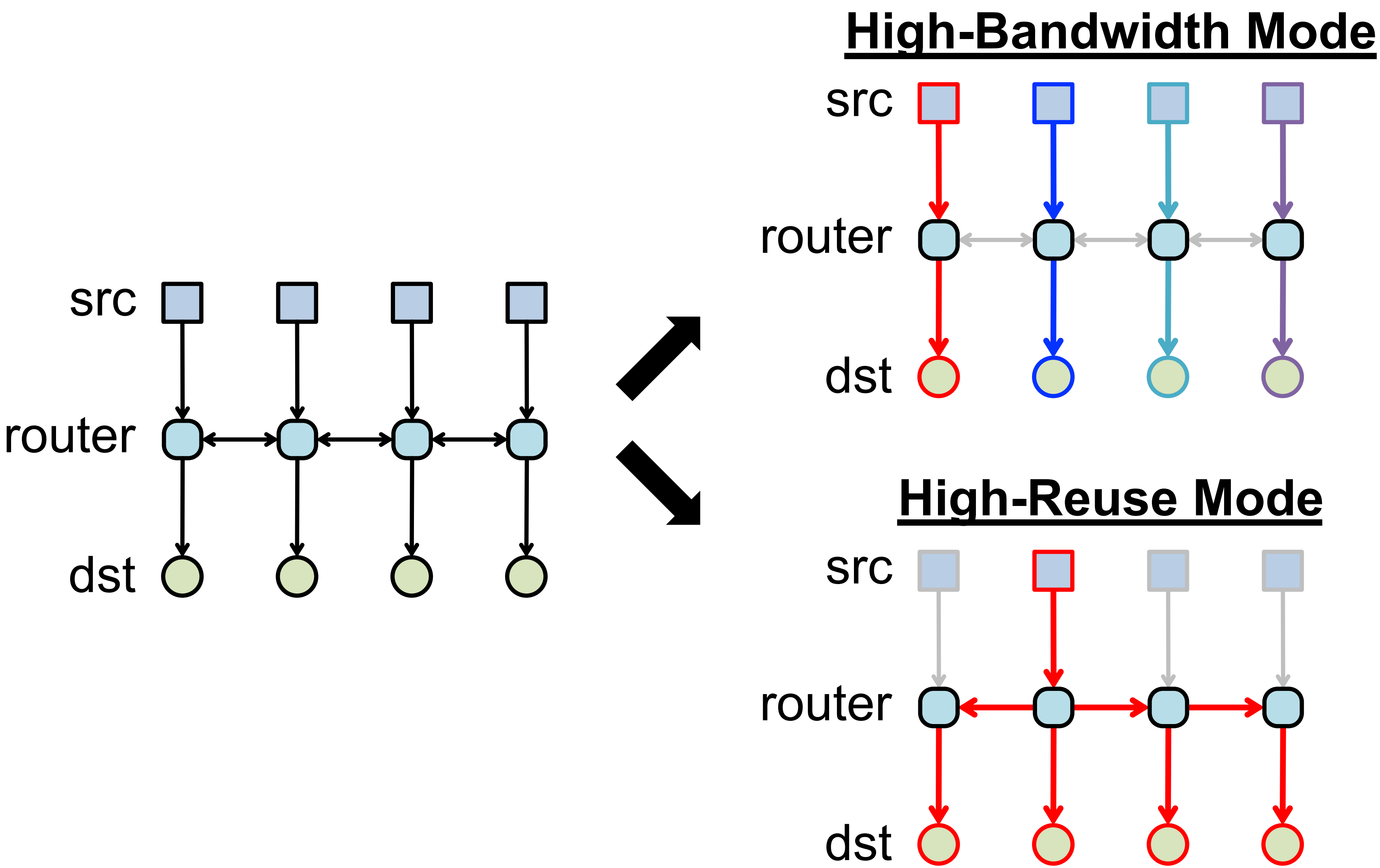


Mesh Network – Best of Both Worlds



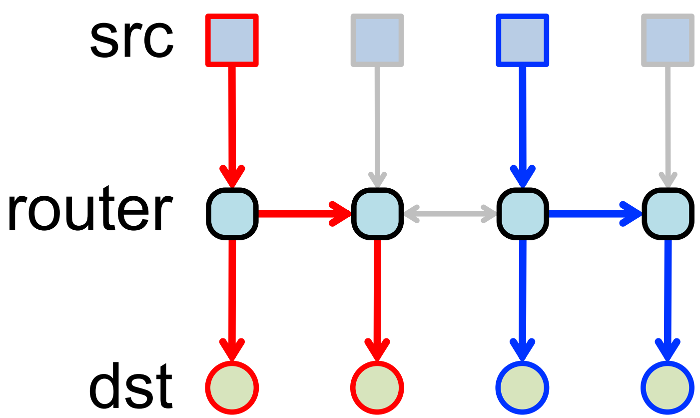


Mesh Network – Best of Both Worlds



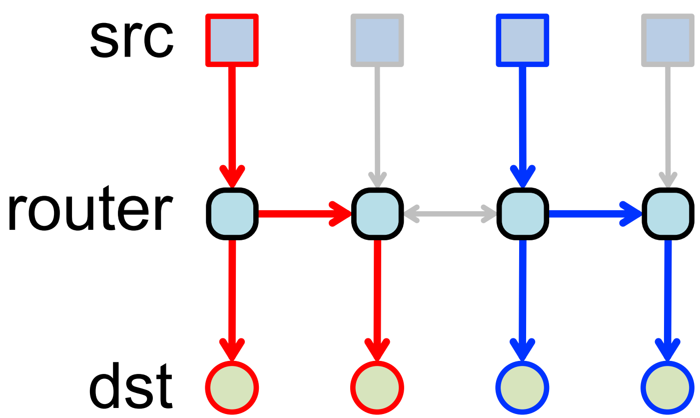
Mesh Network – More Complicated Cases

Grouped-Multicast Mode

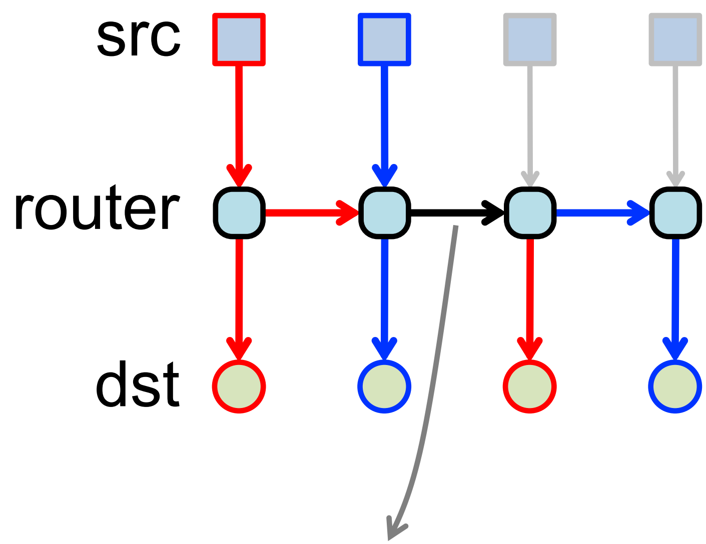


Mesh Network – More Complicated Cases

Grouped-Multicast Mode



Interleaved-Multicast Mode

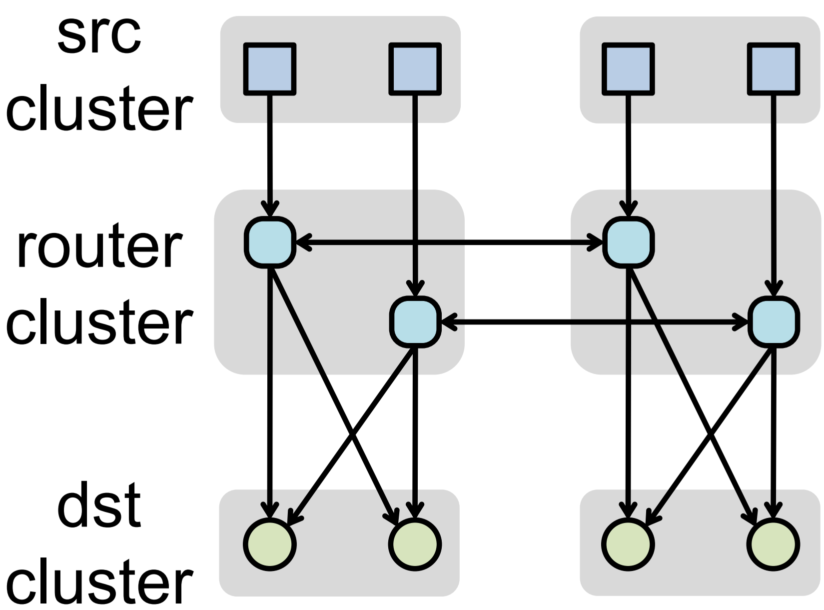


Bandwidth-limited route
(flow control required)

Hierarchical Mesh Network

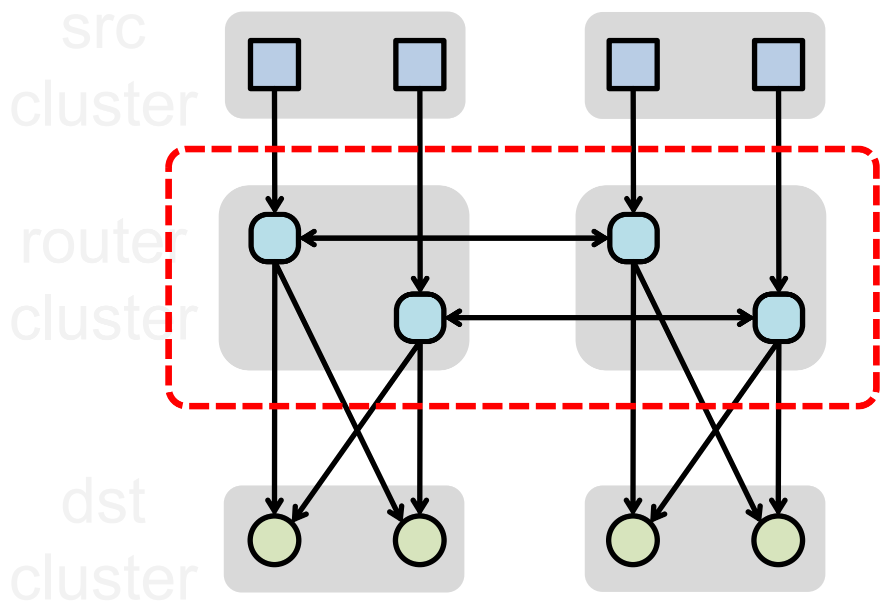
- Flexible to support patterns ranging from high reuse to high bandwidth scenarios
- Can be easily scaled at a low cost

Hierarchical Mesh Network

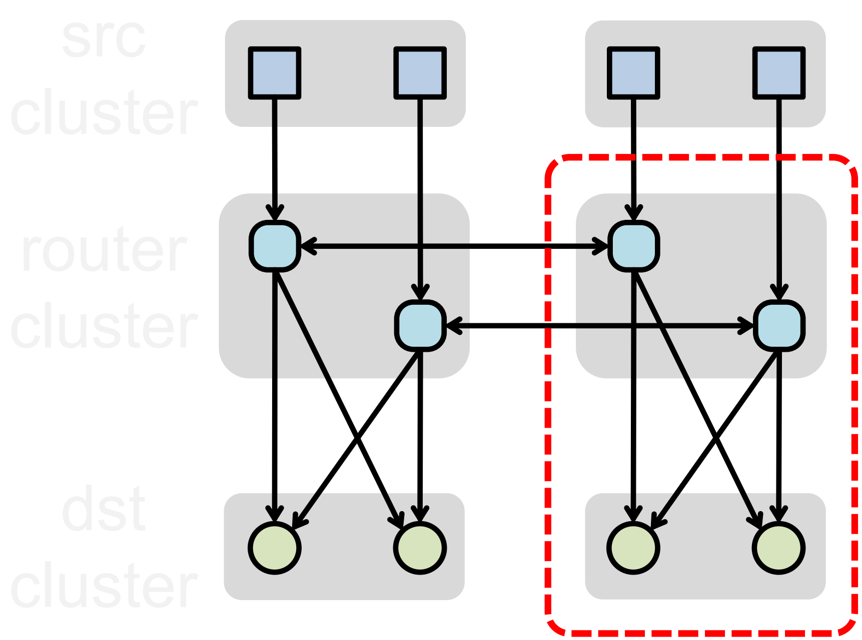


Hierarchical Mesh Network

Mesh Network for **inter-cluster** connections



Hierarchical Mesh Network

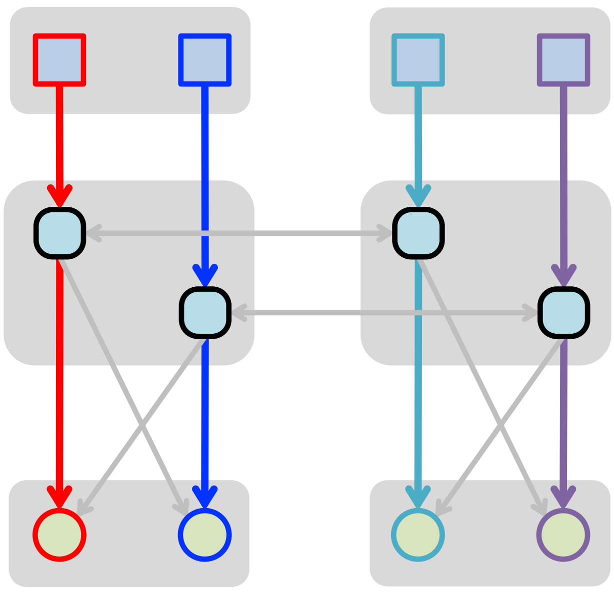


All-to-All Network for **intra-cluster** connections

Complexity is contained within a cluster

Hierarchical Mesh Network

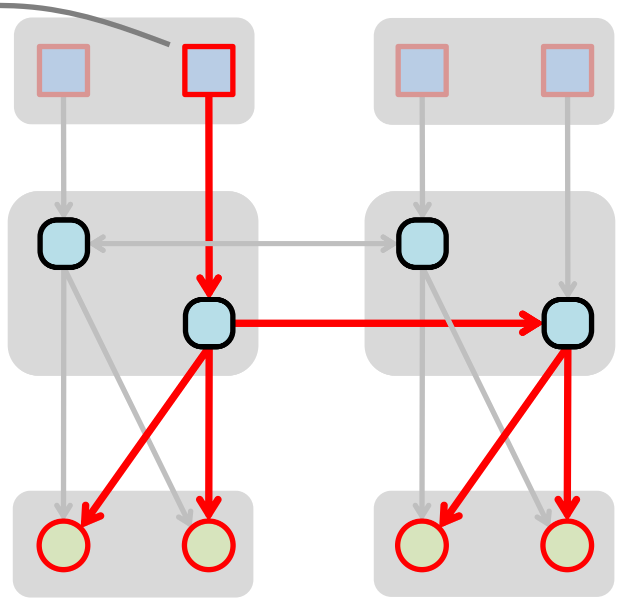
High-Bandwidth Mode



Hierarchical Mesh Network

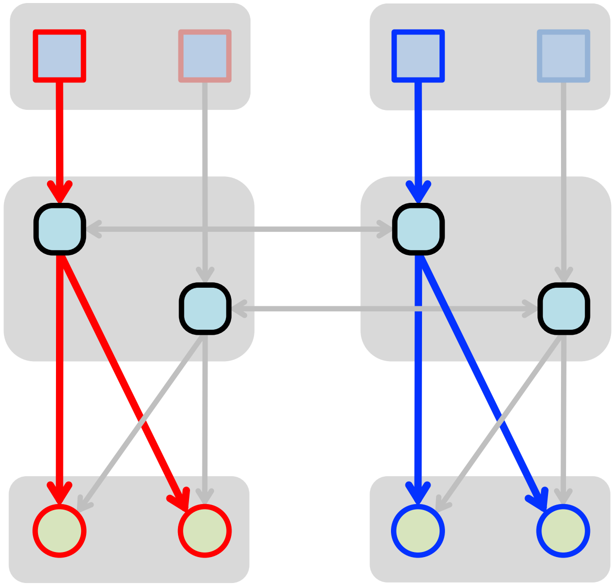
High-Reuse Mode

from any one src



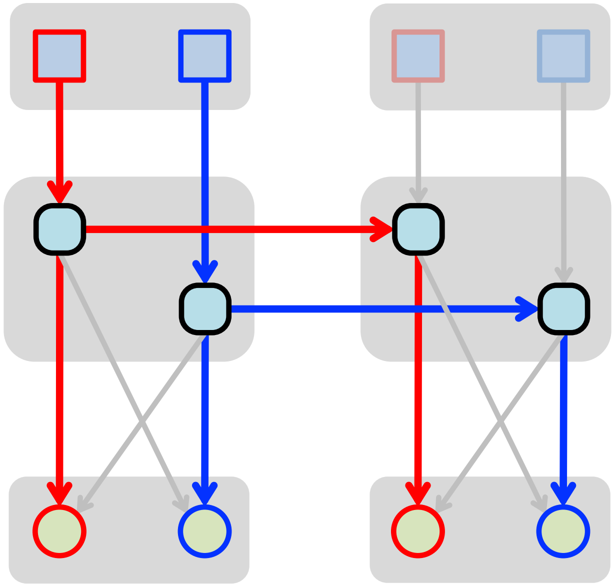
Hierarchical Mesh Network

Grouped-Multicast Mode



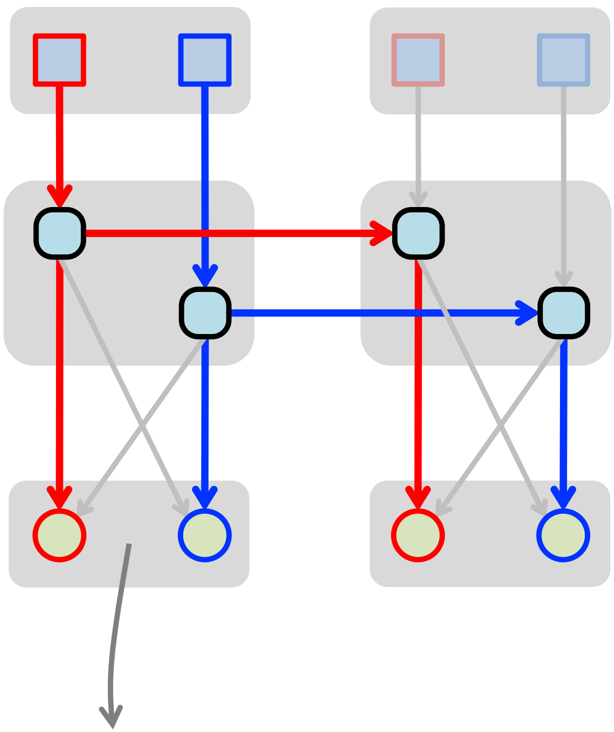
Hierarchical Mesh Network

Interleaved-Multicast Mode



Hierarchical Mesh Network

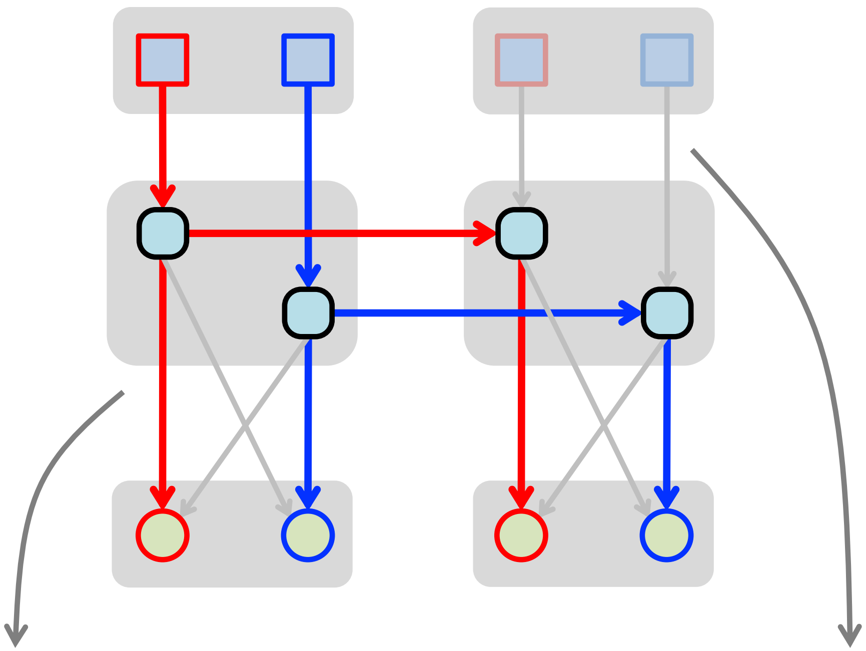
Interleaved-Multicast Mode



Can interleave more by scaling up the cluster size

Hierarchical Mesh Network

Interleaved-Multicast Mode

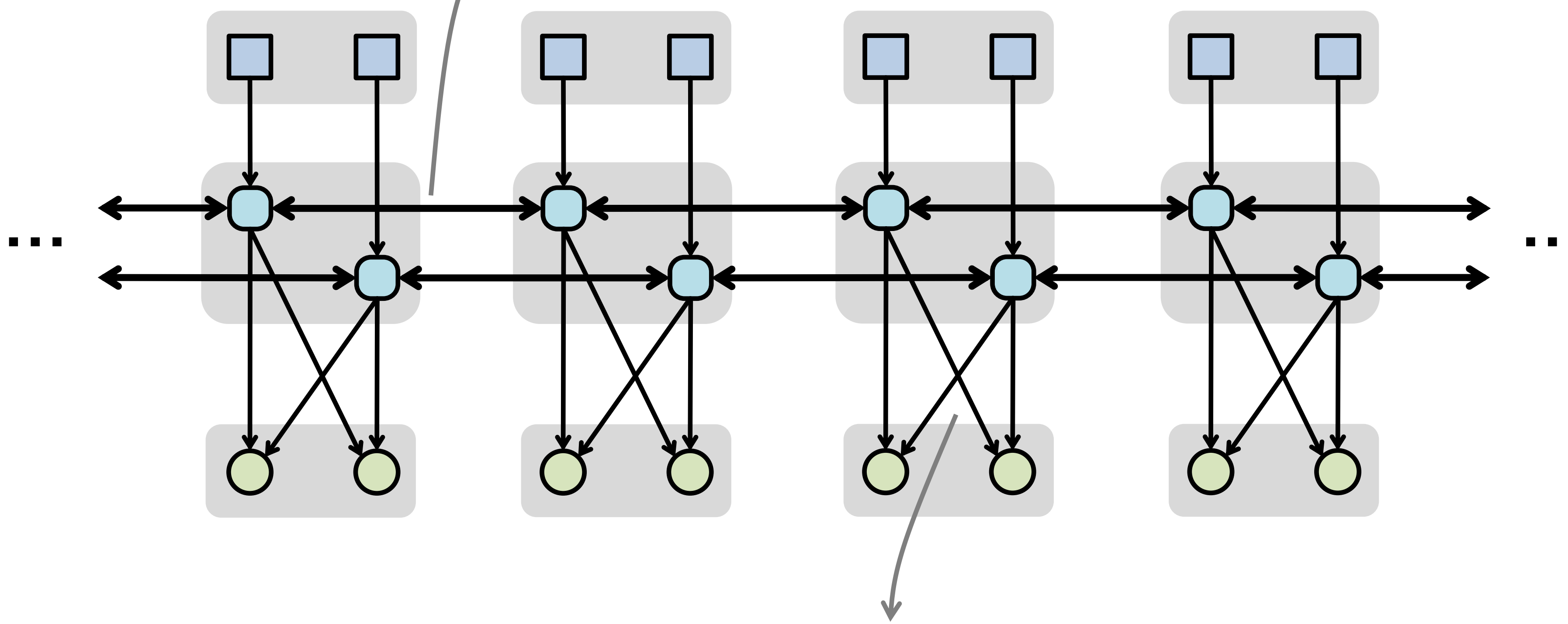


All routes are determined **at configuration time**

→ **Routers** are **circuit-switched** (only **MUXes**)

Scaling the Hierarchical Mesh Network

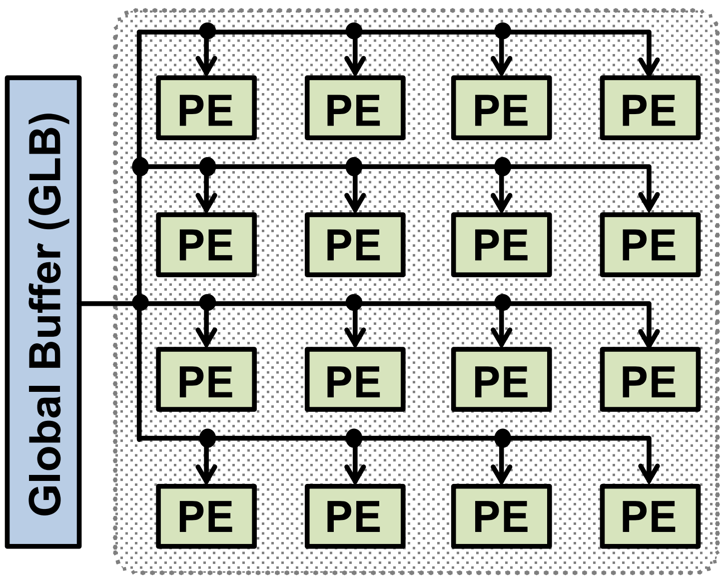
Cost of the mesh network scales **linearly**



Cluster size can scale up depending on tolerable cost of the all-to-all networks

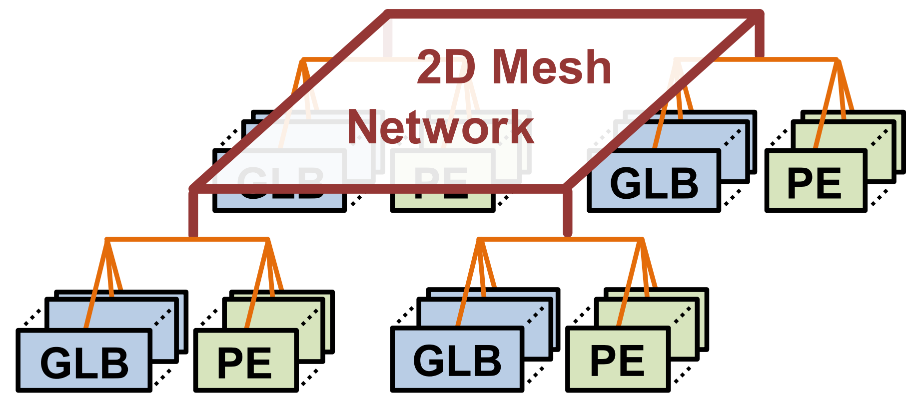
Eyeriss with Hierarchical Mesh Network

Eyeriss v1



- Multicast Network
- Centralized GLB

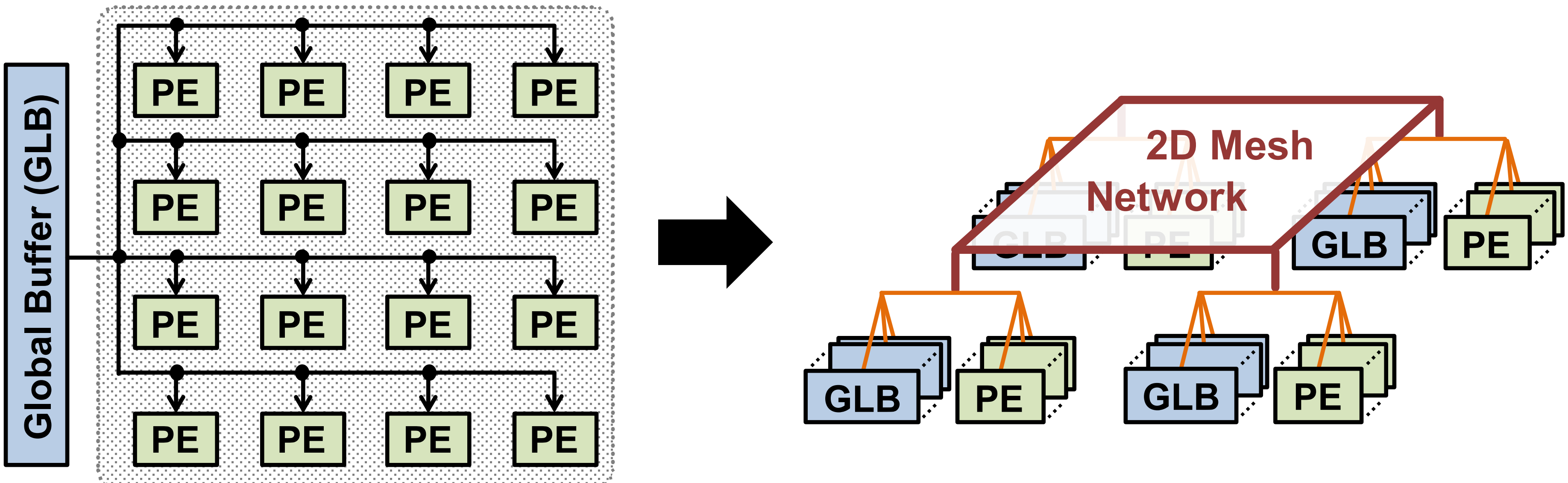
Eyeriss v2



- Hierarchical Mesh Network
- Distributed GLB

Eyeriss with Hierarchical Mesh Network

Eyeriss v1 / Eyeriss v2

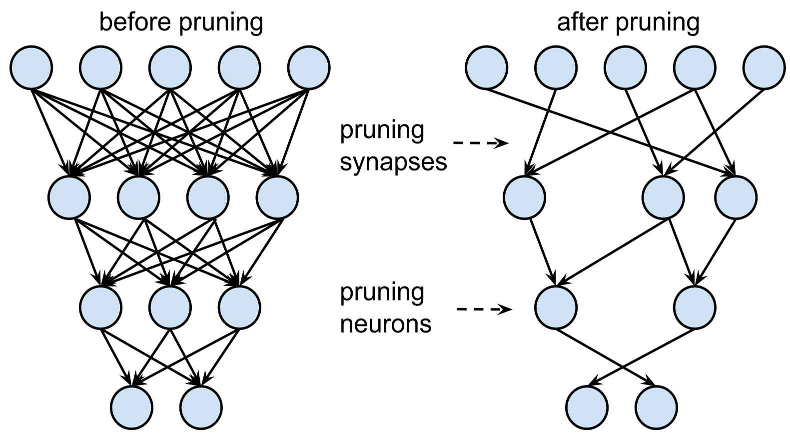


	Speedup	Energy Efficiency
AlexNet	6.9×	2.6×
MobileNet	5.6×	1.8×

192 PEs and 192 KB total GLB for both v1 and v2

DNNs are Becoming More Compact!

Network Pruning

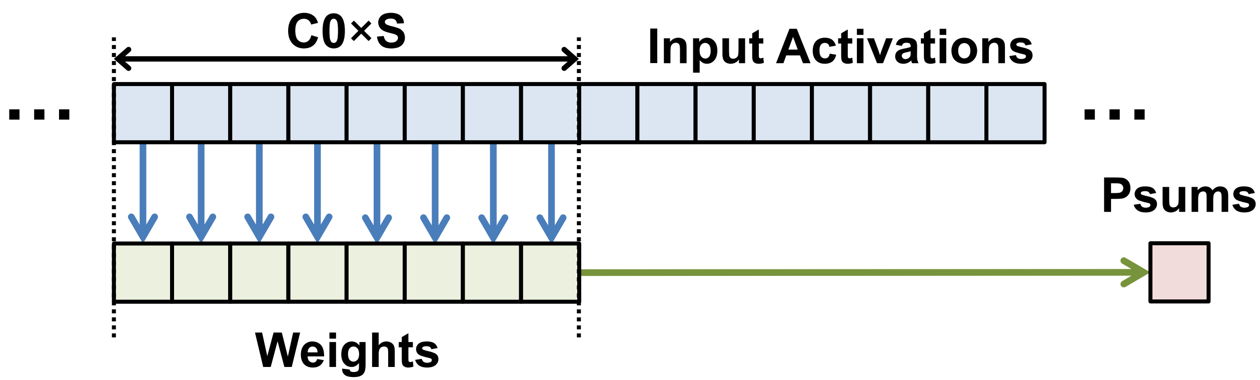


	Year	Accuracy*	# Layers	# Weights	# MACs
AlexNet	2012	80.4%	8	**61M**	**724M**
AlexNet [1] **(pruned)**	2017	79.6%	8	**5.7M** (non-zero)	**58M** (non-zero)

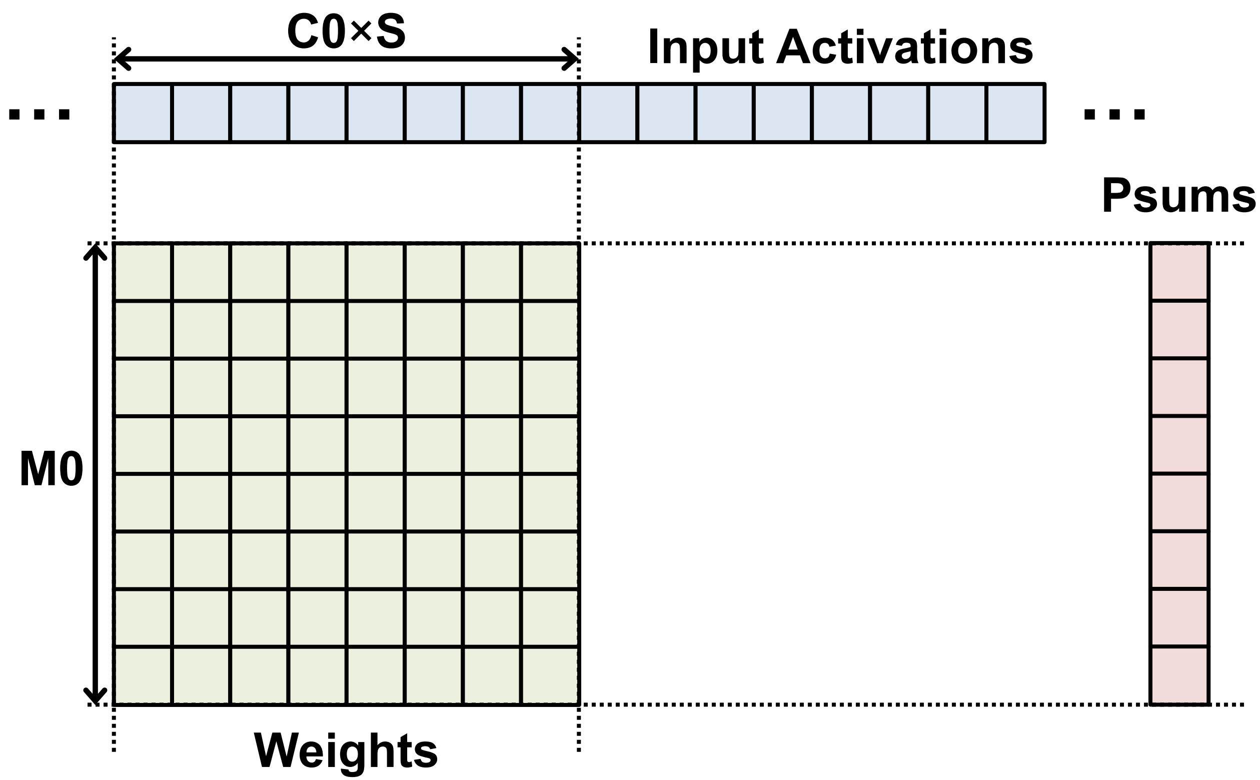
* ImageNet Classification Top-5

[1] Yang, *CVPR* 2017

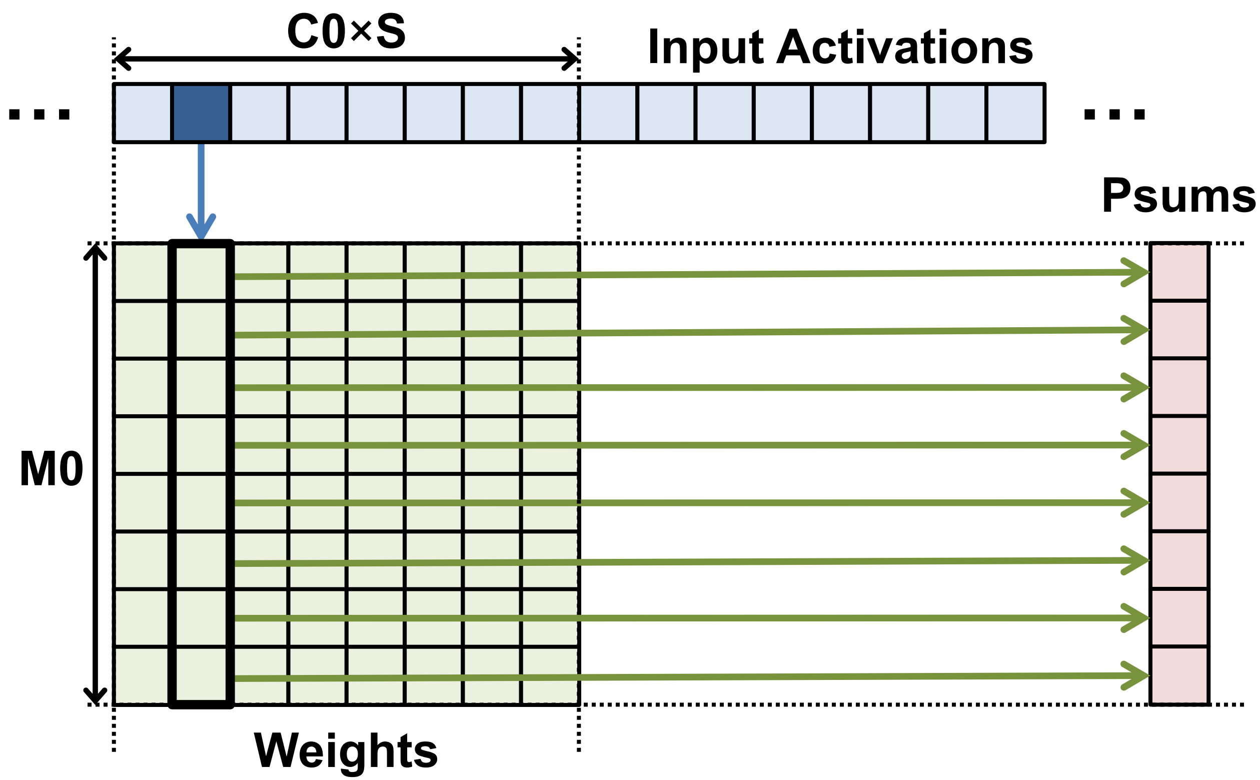
Processing in Eyeriss v1 PE



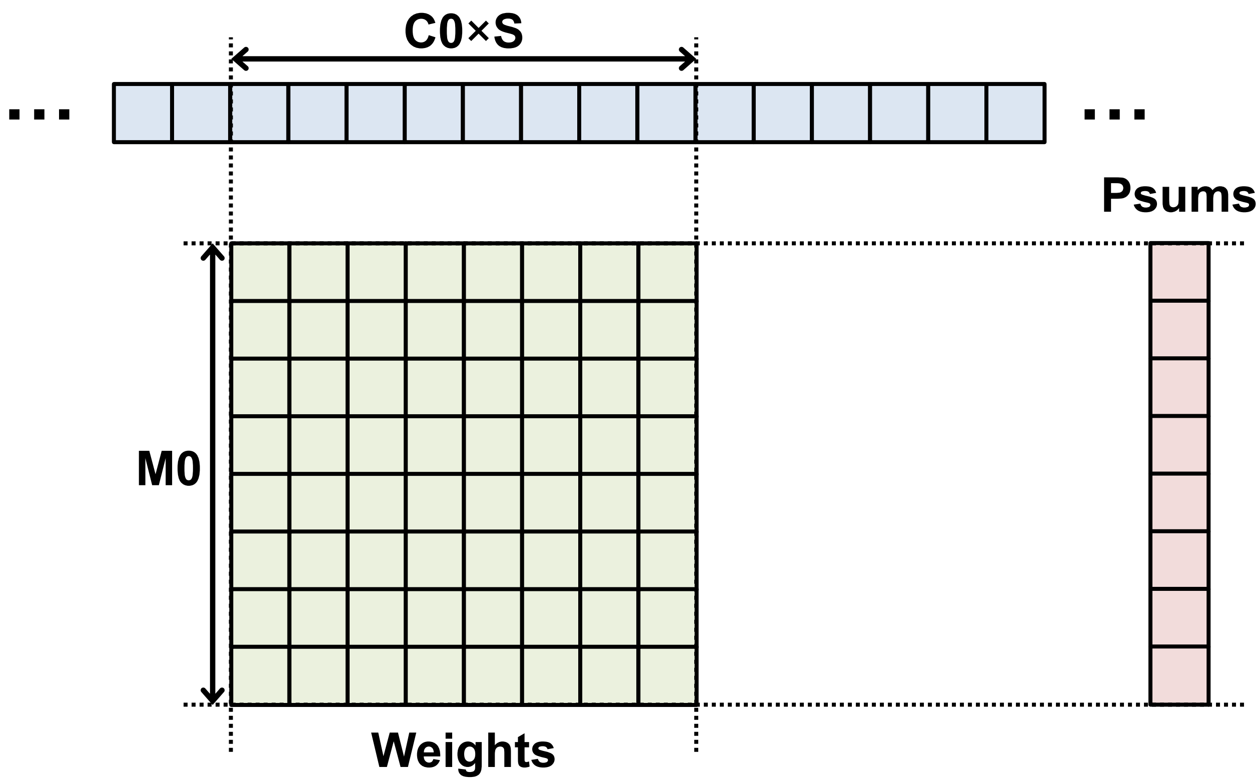
Processing in Eyeriss v1 PE



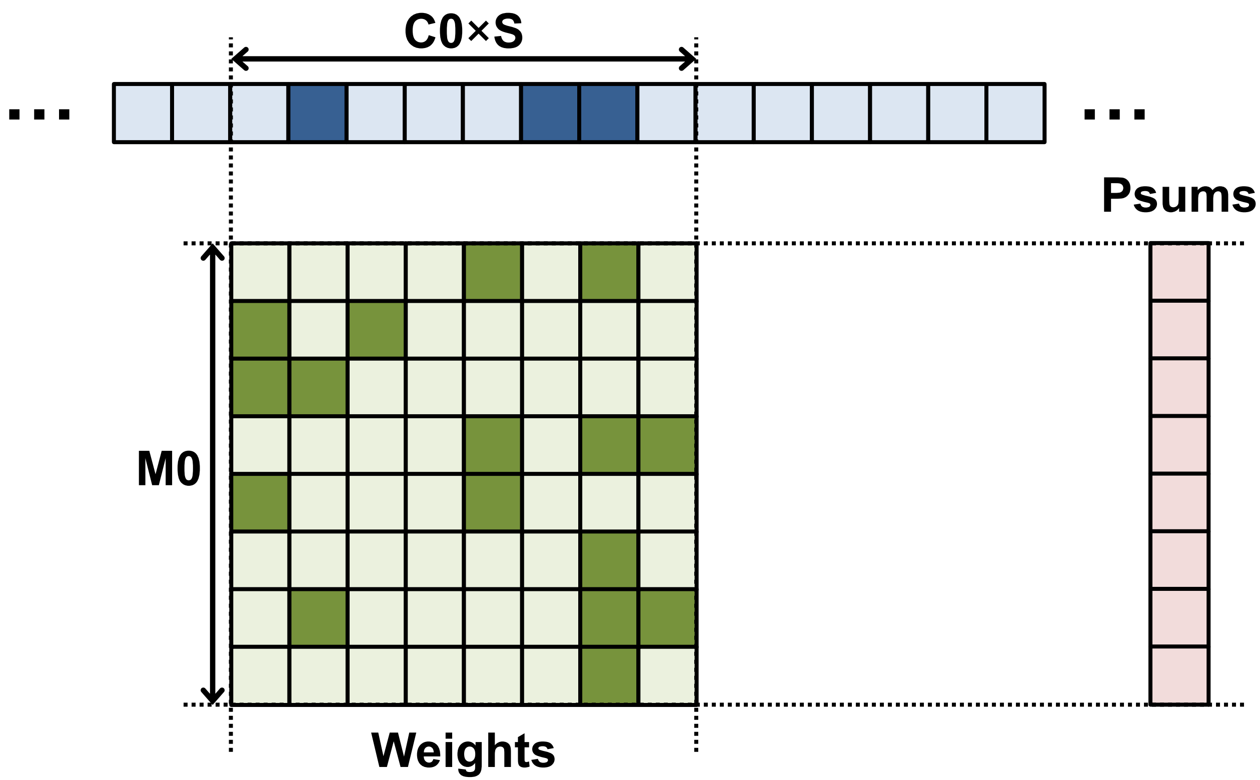
Processing in Eyeriss v1 PE



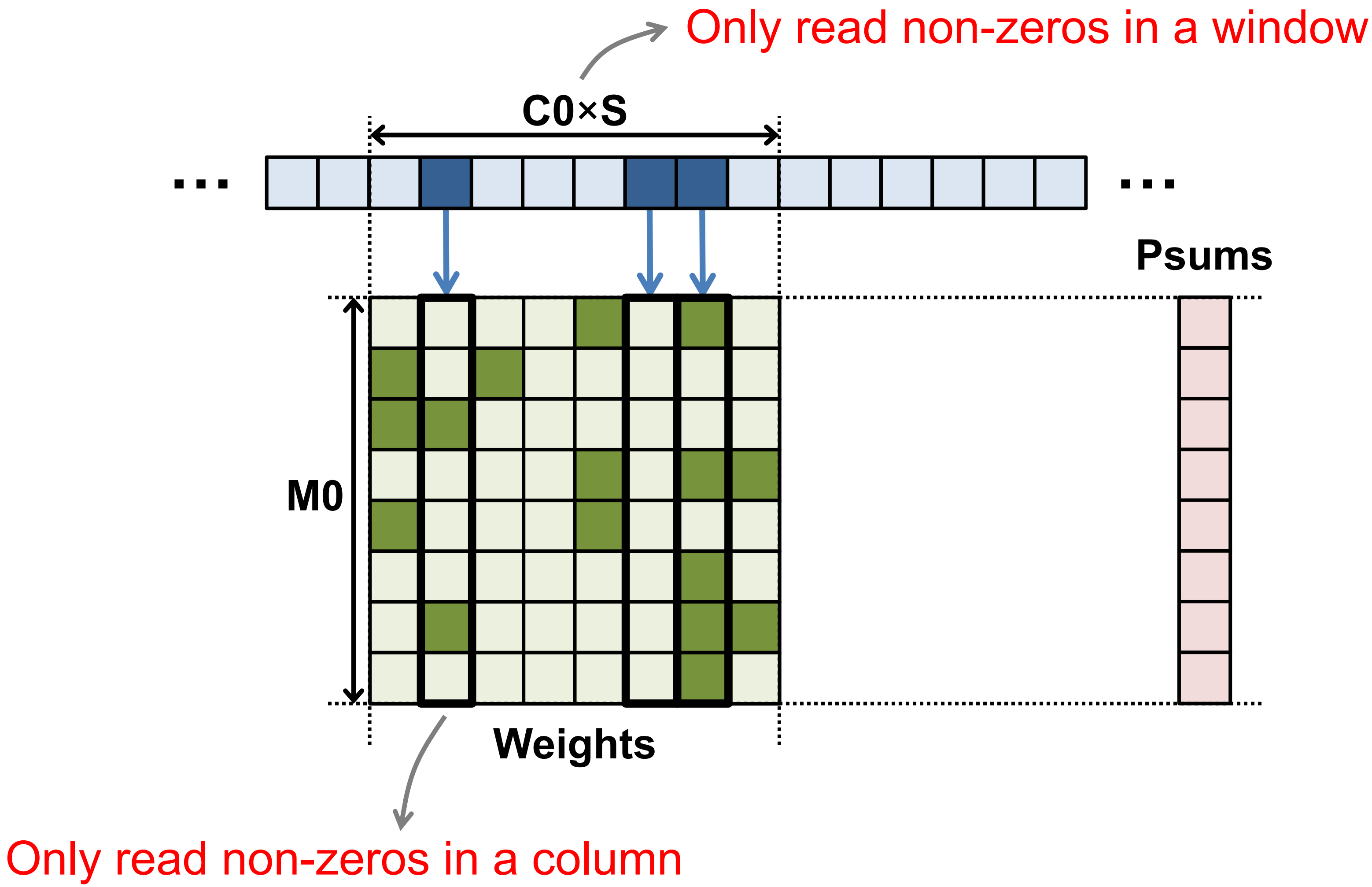
Processing in Eyeriss v1 PE



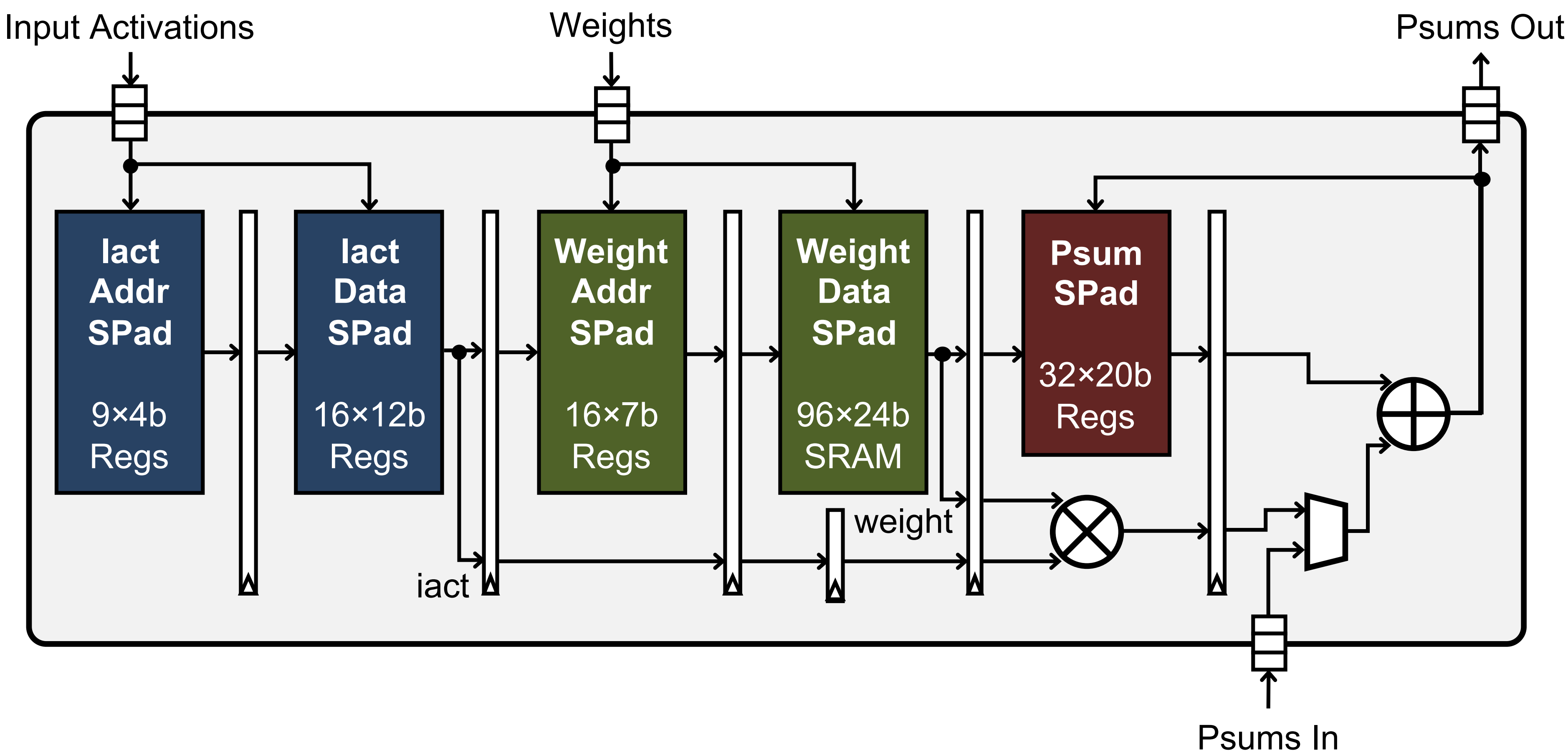
Processing in a PE with Sparsity



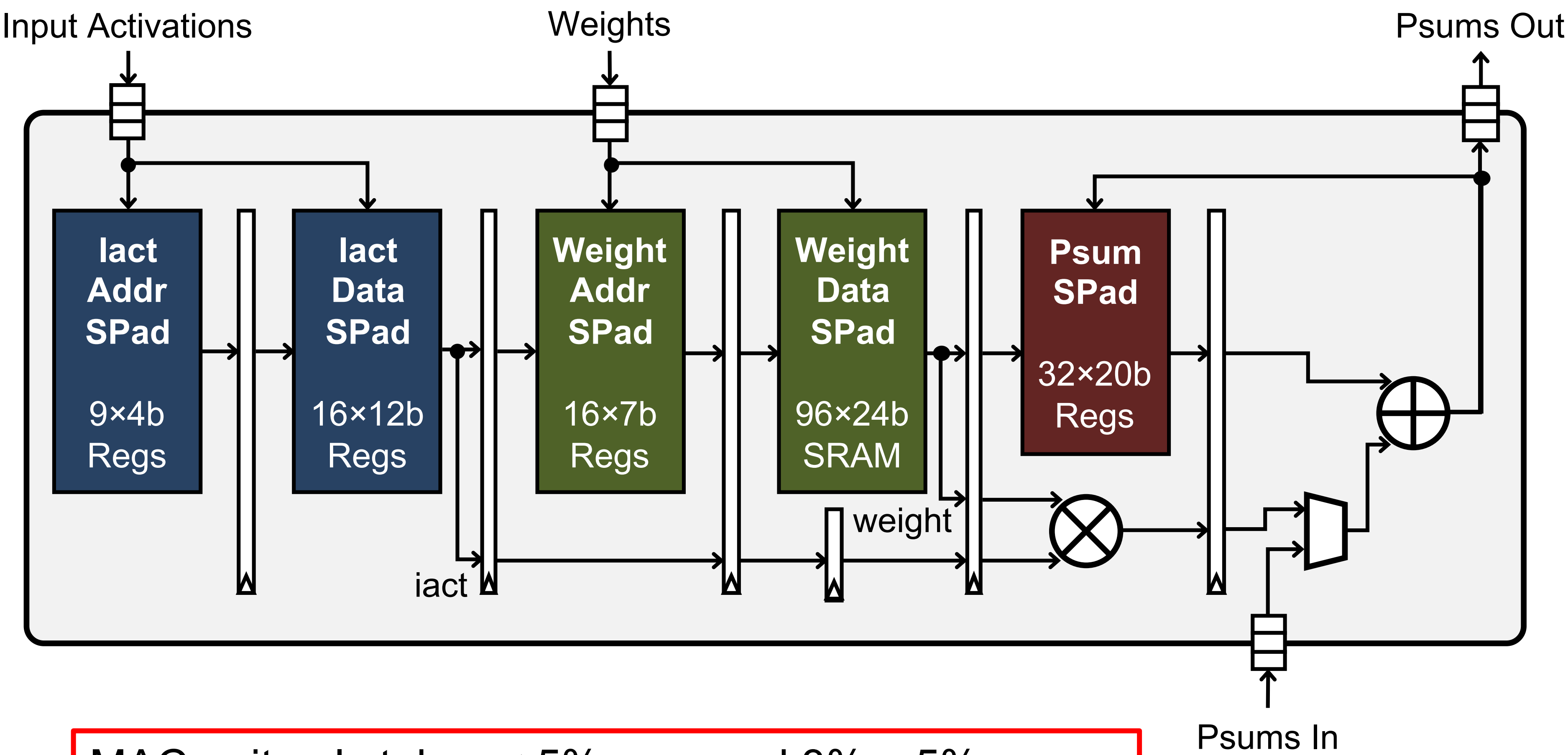
Processing in a PE with Sparsity



Sparse PE Architecture in Eyeriss v2

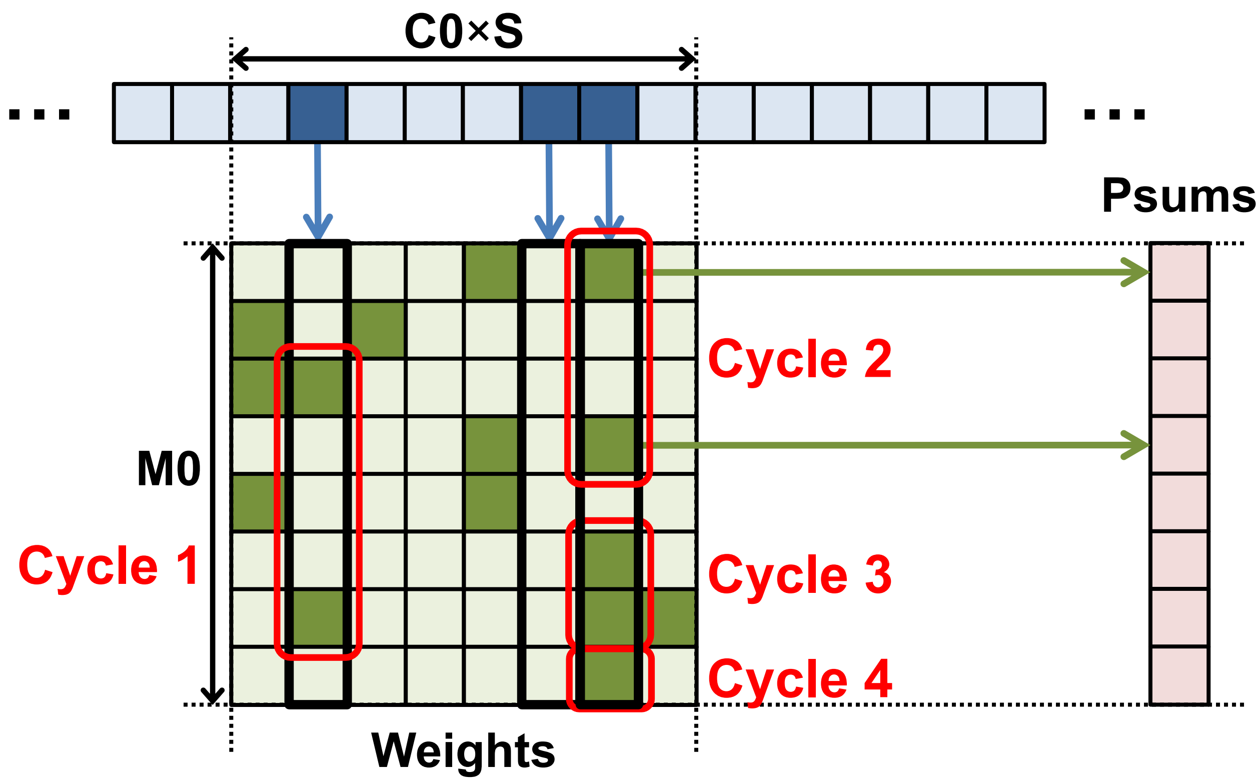


Sparse PE Architecture in Eyeriss v2

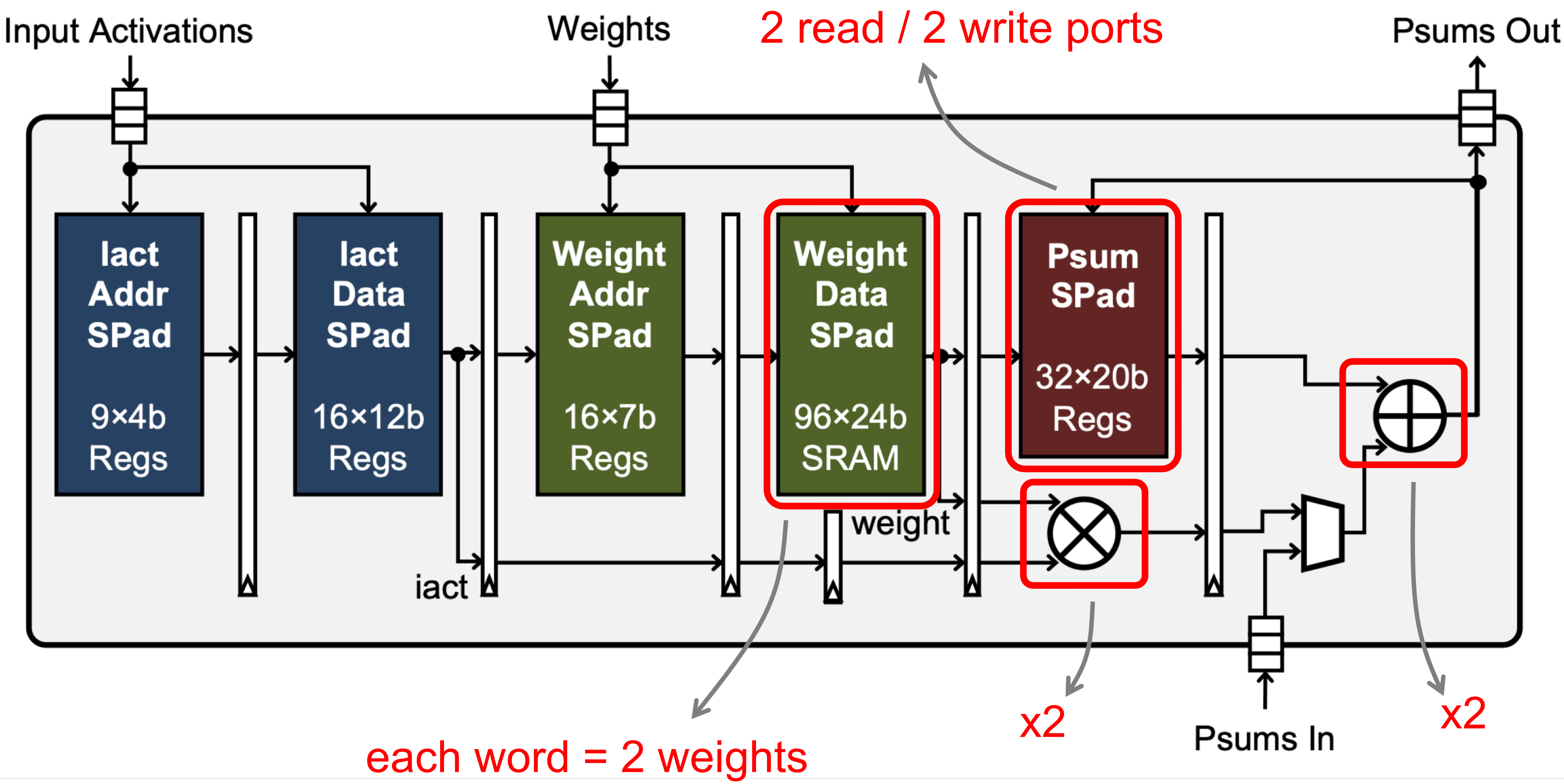


MAC unit only takes < 5% area and 3% – 5% power

SIMD Processing in a PE



Sparse SIMD PE Architecture in Eyeriss v2



Eyeriss v2 Performance Summary

Same Accuracy

	AlexNet	sparse-AlexNet	MobileNet
GOPS	148.3	**405.8**	**126.4**
fps	102.4	280.1	**1285.2**
Over v1	**15.5×**	**42.5×**	**10.9×**
GOPS/W	277.9	**1028.1**	**198.5**
Inferences/J	191.8	709.7	**2020.8**
Over v1	**3.0×**	**11.3×**	**1.9×**

Benchmarking Metrics for DNN Hardware

How can we compare designs?

V. Sze, Y.-H. Chen, T-J. Yang, J. Emer,
"***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***,"
Proceedings of the IEEE, Dec. 2017

Metrics for DNN Hardware

- **Accuracy**
  - Quality of result for a given task
- **Throughput**
  - Analytics on high volume data
  - Real-time performance (e.g., video at 30 fps)
- **Latency**
  - For interactive applications (e.g., autonomous navigation)
- **Energy and Power**
  - Edge and embedded devices have limited battery capacity
  - Data centers have stringent power ceilings due to cooling costs
- **Hardware Cost**
  - $$$

Specifications to Evaluate Metrics

- **Accuracy**
  - Difficulty of dataset and/or task should be considered
- **Throughput**
  - Number of cores (include utilization along with peak performance)
  - Runtime for running specific DNN models
- **Latency**
  - Include batch size used in evaluation
- **Energy and Power**
  - Power consumption for running specific DNN models
  - Include external memory access
- **Hardware Cost**
  - On-chip storage, number of cores, chip area + process technology

Comprehensive Coverage

- **All metrics** should be reported for fair evaluation of design tradeoffs
- Examples of what can happen if certain metric is omitted:
  - **Without the accuracy given for a specific dataset and task**, one could run a simple DNN and claim low power, high throughput, and low cost – however, the processor might not be usable for a meaningful task
  - **Without reporting the off-chip bandwidth**, one could build a processor with only multipliers and claim low cost, high throughput, high accuracy, and low chip power – however, when evaluating system power, the off-chip memory access would be substantial
- Are results measured or simulated? On what test data?

Evaluation Process

The evaluation process for whether a DNN system is a viable solution for a given application might go as follows:

1. **Accuracy** determines if it can perform the given task
2. **Latency and throughput** determine if it can run fast enough and in real-time
3. **Energy and power consumption** will primarily dictate the form factor of the device where the processing can operate
4. **Cost**, which is primarily dictated by the chip area, determines how much one would pay for this solution

Summary

- The number of weights and MACs are not sufficient for evaluating the energy consumption and latency of DNNs
  - **Designers of efficient DNN algorithms should directly target direct metrics such as energy and latency and incorporate into the design**
- Many of the existing DNN processors rely on certain properties of the DNN which cannot be guaranteed as the wide range techniques used for efficient DNN algorithm design has resulted in a more diverse set of DNNs
  - **DNN hardware used to process these DNNs should be sufficiently flexible to support a wide range of techniques efficiently**
- Evaluate DNN hardware on a comprehensive set of benchmarks and metrics

Acknowledgements

Joel Emer

Sertac Karaman

Thomas Heldt

Research conducted in the **MIT Energy-Efficient Multimedia Systems Group** would not be possible without the support of the following organizations:

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH · NSF · ANALOG DEVICES · BROADCOM · Google · intel · IBM · DARPA · SRC · 3M · NVIDIA · QUALCOMM · SAMSUNG · TEXAS INSTRUMENTS · tsmc

For updates on our research Follow @eems_mit

RESEARCH LABORATORY OF ELECTRONICS AT MIT

MTL microsystems technology laboratories massachusetts institute of technology

References

- **Limitations of Existing Efficient DNN Approaches**
  - *Y.-H. Chen*, T.-J. Yang*, J. Emer, V. Sze, "Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," SysML Conference, February 2018.*
  - *V. Sze, Y.-H. Chen, T.-J. Yang, J. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, December 2017.*
  - *Hardware Architecture for Deep Neural Networks:* http://eyeriss.mit.edu/tutorial.html
- **Co-Design of Algorithms and Hardware for Deep Neural Networks**
  - *T.-J. Yang, Y.-H. Chen, V. Sze, "Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.*
  - *Energy estimation tool:* http://eyeriss.mit.edu/energy.html
  - *T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, V. Sze, H. Adam, "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," European Conference on Computer Vision (ECCV), 2018.* http://netadapt.mit.edu/
  - *D. Wofk*, F. Ma*, T.-J. Yang, S. Karaman, V. Sze, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," IEEE International Conference on Robotics and Automation (ICRA), May 2019.* http://fastdepth.mit.edu/
  - *T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, L.-C. Chen, "DeeperLab: Single-Shot Image Parser," arXiv, February 2019.* http://deeperlab.mit.edu/

References

- **Energy-Efficient Hardware for Deep Neural Networks**
  - **Project website:** http://eyeriss.mit.edu
  - *Y.-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid State Circuits (JSSC), ISSCC Special Issue, Vol. 52, No. 1, pp. 127-138, January 2017.*
  - *Y.-H. Chen, J. Emer, V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," International Symposium on Computer Architecture (ISCA), pp. 367-379, June 2016.*
  - *Y.-H. Chen, T.-J. Yang, J. Emer, V. Sze, "Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices," IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), June 2019.*
  - *Eyexam:* https://arxiv.org/abs/1807.07928

- ***Looking beyond DNN Accelerator for Acceleration***
  - *Z. Zhang, V. Sze, "FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos," CVPR Workshop on New Trends in Image Restoration and Enhancement, July 2017.* www.rle.mit.edu/eems/fast