Efficient Computing for AI and Robotics: From Hardware Accelerators to Algorithm Design

Vivienne Sze (🐦@eems_mit)

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Sertac Karaman, Tushar Krishna, Peter Li, Yi-Lun Liao, Fangchang Ma, Angshu Parashar, Amr Suleiman, Po-An Tsai, Diana Wofk, Nellie Wu, Tien-Ju Yang, Zhengdong Zhang

Slides available at
https://tinyurl.com/ISCAS2021Sze

Processing at "Edge" instead of the "Cloud"

Communication

Privacy

Latency
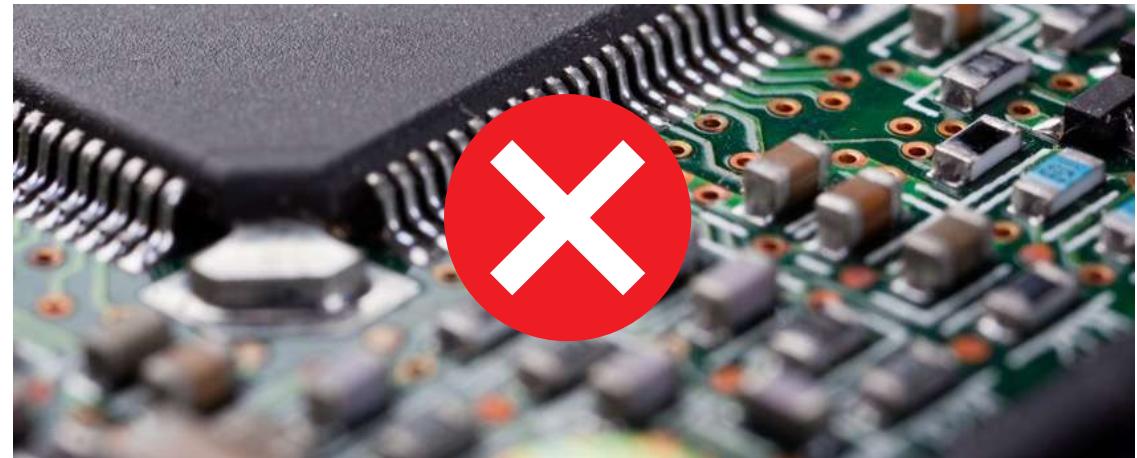
Existing Processors Consume Too Much Power


< 1 Watt
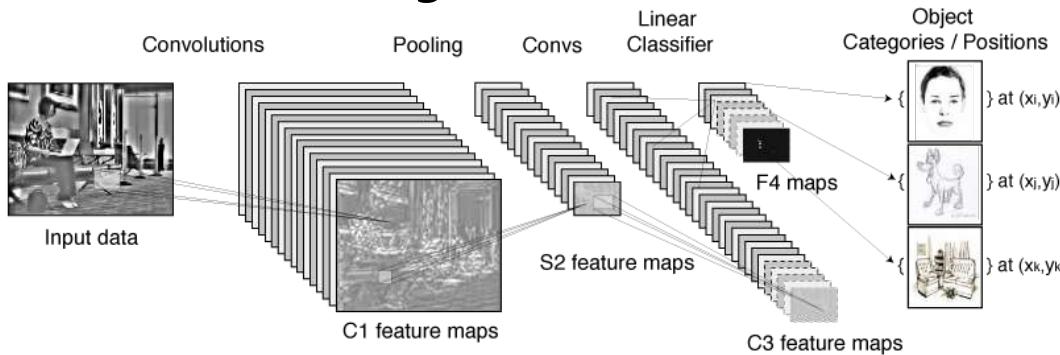
**> 10 Watts**

Efficient Computing with Cross-Layer Design
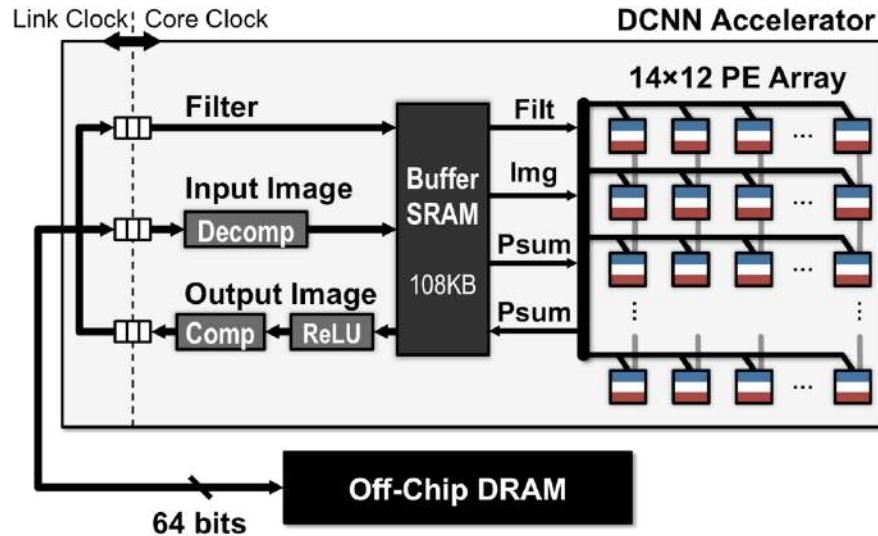
Algorithms
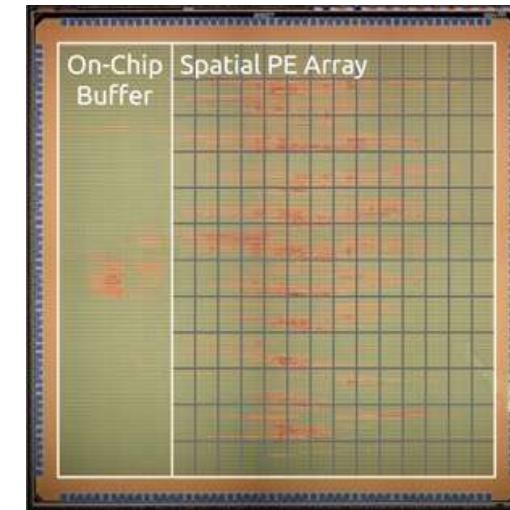
Systems
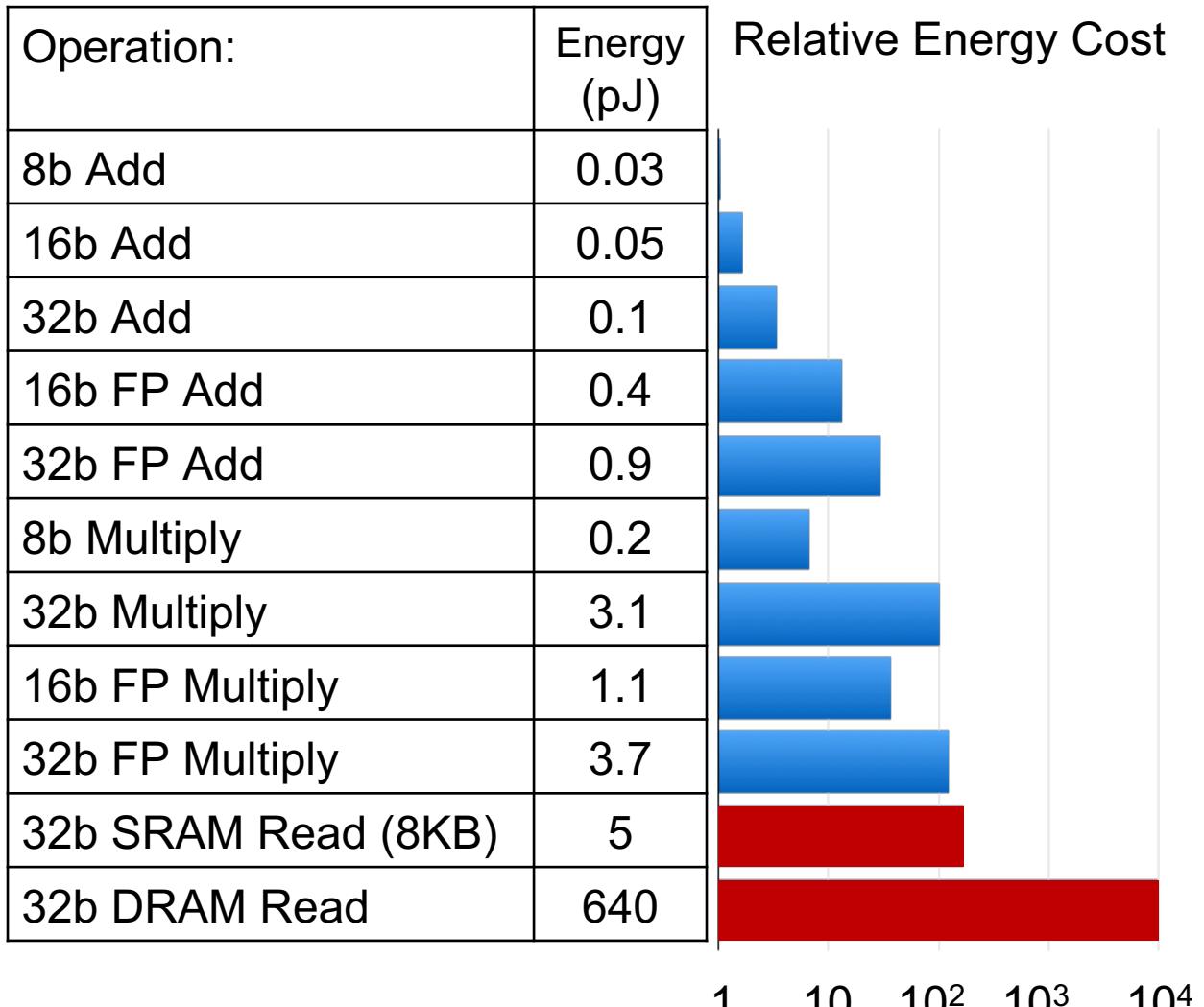
Architectures

Circuits

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

Energy Dominated by Data Movement

Operation:	Energy (pJ)	Relative Energy Cost
8b Add	0.03	
16b Add	0.05	
32b Add	0.1	
16b FP Add	0.4	
32b FP Add	0.9	
8b Multiply	0.2	
32b Multiply	3.1	
16b FP Multiply	1.1	
32b FP Multiply	3.7	
32b SRAM Read (8KB)	5	
32b DRAM Read	640	

$1 \quad 10 \quad 10^2 \quad 10^3 \quad 10^4$

Memory access is **orders of magnitude** higher energy than compute
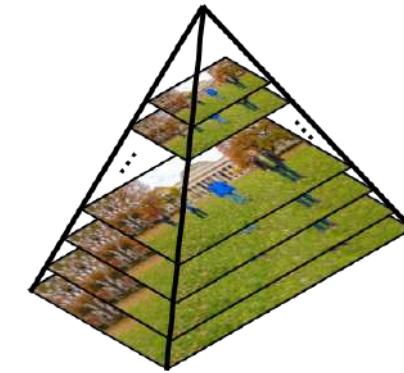
Autonomous Navigation Uses a Lot of Data

Semantic Understanding

- High frame rate
- Large resolutions
- Data expansion

Geometric Understanding

- Growing map size

2 million pixels

10x-100x more pixels

[**Pire**, *RAS* 2017]

Visual-Inertial Localization

Determines location/orientation of robot from images and IMU
(also used by headset in Augmented Reality and Virtual Reality)

Image sequence

IMU
Inertial Measurement Unit

Visual-Inertial Odometry (VIO)*

Localization

Mapping

*Subset of SLAM algorithm
(Simultaneous Localization And Mapping)

Localization at Under 25 mW

First chip that performs
complete Visual-Inertial Odometry

Front-End for camera
(Feature detection, tracking, and outlier elimination)

Front-End for IMU
*(pre-integration of accelerometer and gyroscope data)*
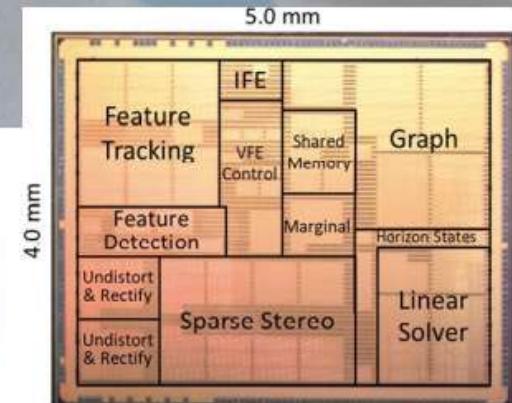
Back-End Optimization of Pose Graph

Consumes **684× and 1582×** less energy than mobile and desktop CPUs, respectively

Navion
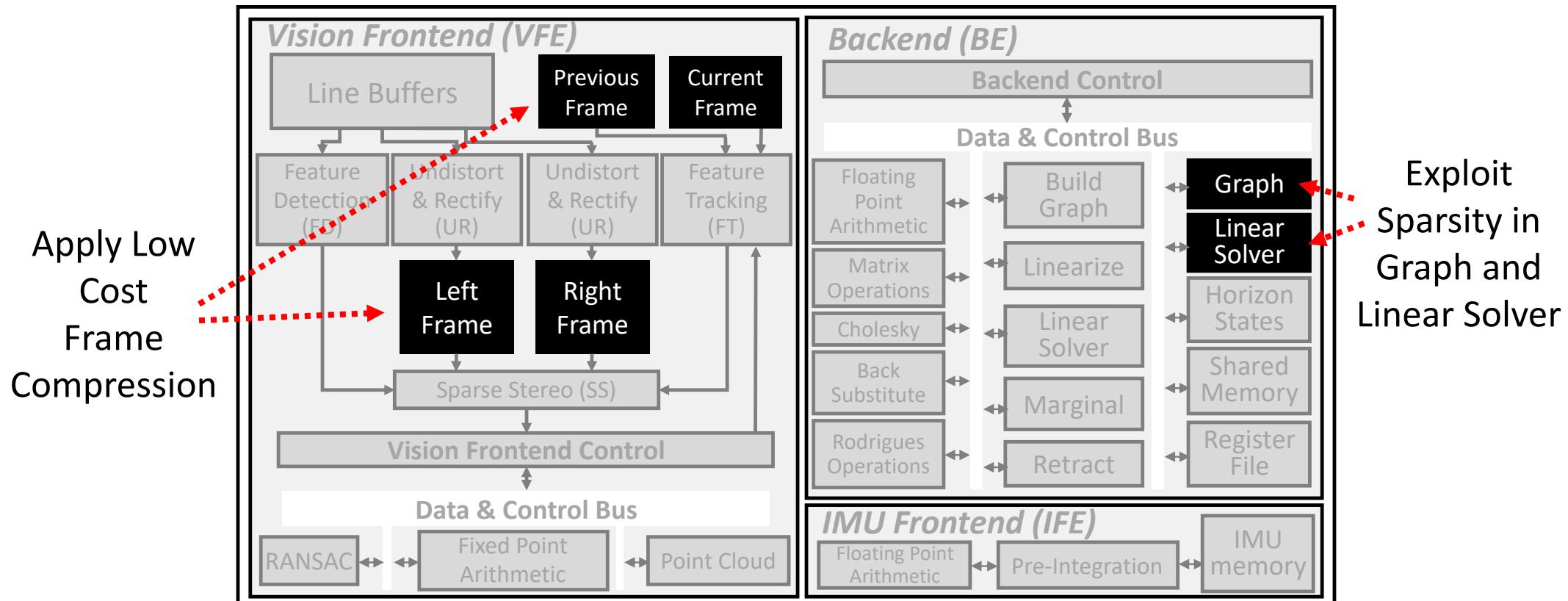
Technology	65nm CMOS	Supply	1 V
Chip area (mm²)	4.0 x 5.0	Resolution	752x480
Core area (mm²)	3.54 x 4.54	Camera rate	28 - 171 fps
Logic gates	2,043 kgates	Keyframe rate	16 - 90 fps
SRAM	854KB	Average Power	24 mW
VFE Frequency	62.5 MHz	GOPS	10.5 – 59.1
BE Frequency	83.3 MHz	GFLOPS	1 – 5.7

[Joint work with Sertac Karaman (AeroAstro)]

[Zhang, *RSS* 2017], [**Suleiman**, *VLSI-C* 2018]

Key Methods to Reduce Data Size

Navion: Fully integrated system – no off-chip processing or storage

Apply Low Cost Frame Compression

Exploit Sparsity in Graph and Linear Solver

Vision Frontend (VFE)
- Line Buffers
- Previous Frame
- Current Frame
- Feature Detection (FD)
- Undistort & Rectify (UR)
- Undistort & Rectify (UR)
- Feature Tracking (FT)
- Left Frame
- Right Frame
- Sparse Stereo (SS)
- Vision Frontend Control
- Data & Control Bus
- RANSAC
- Fixed Point Arithmetic
- Point Cloud

Backend (BE)
- Backend Control
- Data & Control Bus
- Floating Point Arithmetic
- Matrix Operations
- Cholesky
- Back Substitute
- Rodrigues Operations
- Build Graph
- Linearize
- Linear Solver
- Marginal
- Retract
- Graph
- Linear Solver
- Horizon States
- Shared Memory
- Register File

IMU Frontend (IFE)
- Floating Point Arithmetic
- Pre-Integration
- IMU memory

Use **compression** and **exploit sparsity** to reduce memory down to 854kB
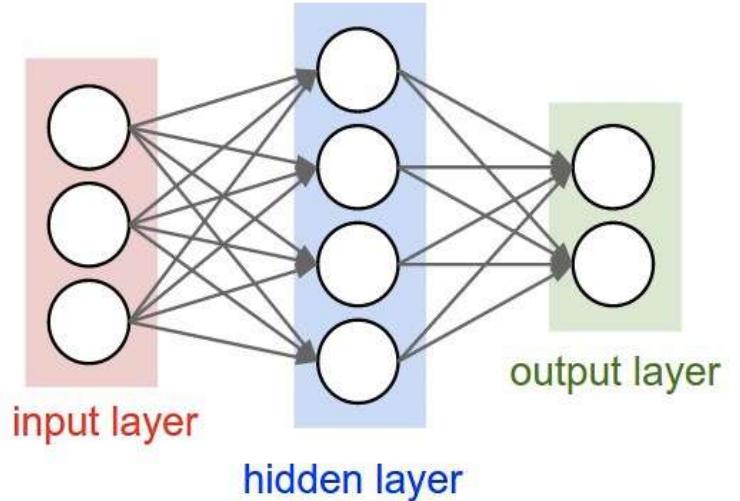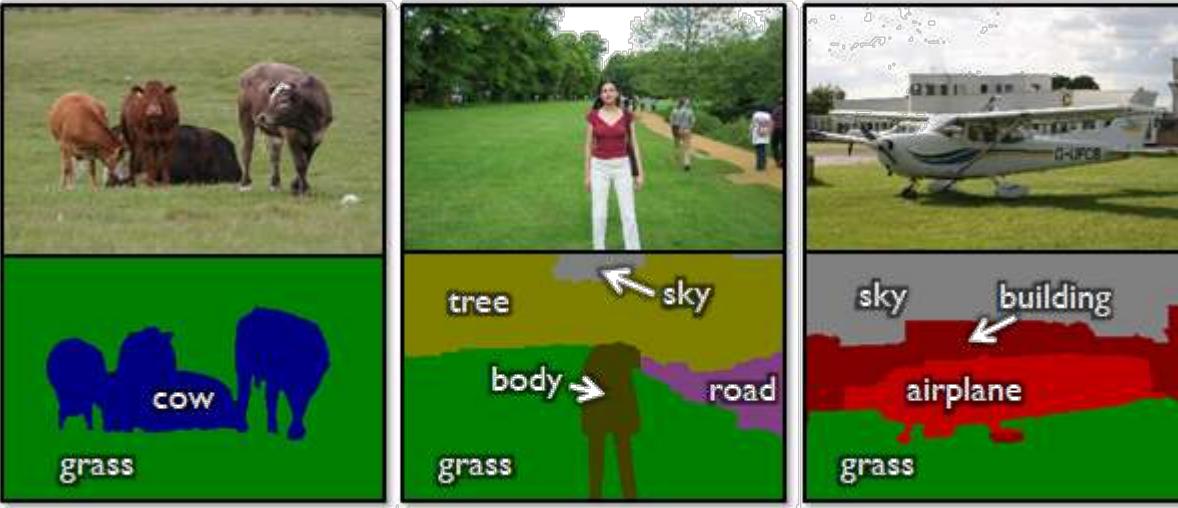
Navion Project Website: http://navion.mit.edu

Understanding the Environment

Depth Estimation

Semantic Segmentation

cow
grass

tree — sky
body → road
grass

sky — building
airplane
grass
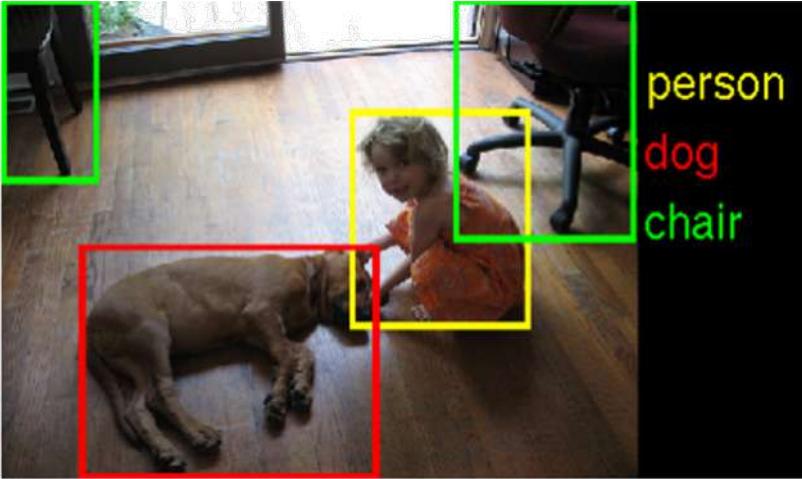
State-of-the-art approaches use **Deep Neural Networks,** which require **up to several hundred millions of operations and weights to compute!**
>100x more complex than video compression

Deep Neural Networks

*Deep Neural Networks (DNNs) have become a **cornerstone of AI***

Computer Vision
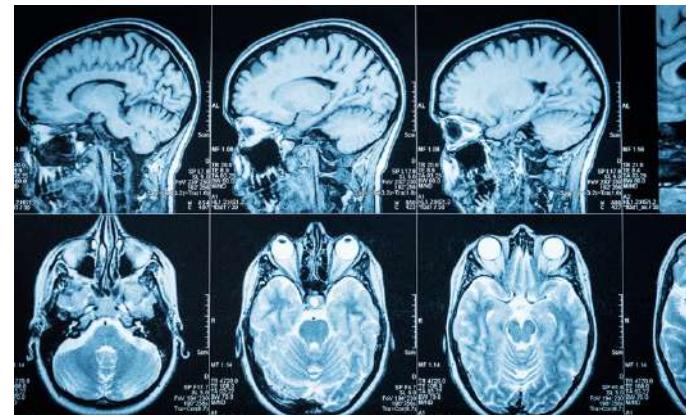
Speech Recognition
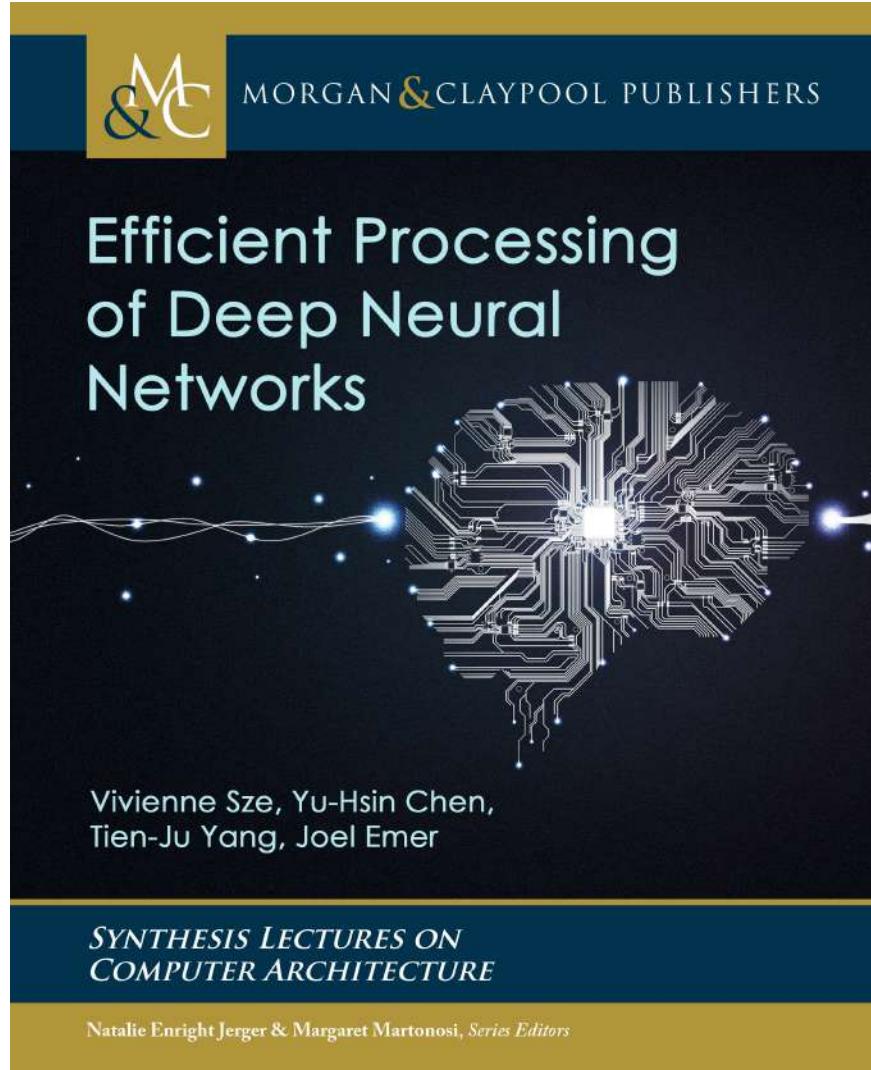
Game Play

Medical

Book on Efficient Processing of DNNs

Efficient Processing of Deep Neural Networks

MORGAN & CLAYPOOL PUBLISHERS

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, Joel Emer

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Natalie Enright Jerger & Margaret Martonosi, *Series Editors*

Part I Understanding Deep Neural Networks
Introduction
Overview of Deep Neural Networks

Part II Design of Hardware for Processing DNNs
Key Metrics and Design Objectives
Kernel Computation
Designing DNN Accelerators
Operation Mapping on Specialized Hardware

Part III Co-Design of DNN Hardware and Algorithms
Reducing Precision
Exploiting Sparsity
Designing Efficient DNN Models
Advanced Technologies
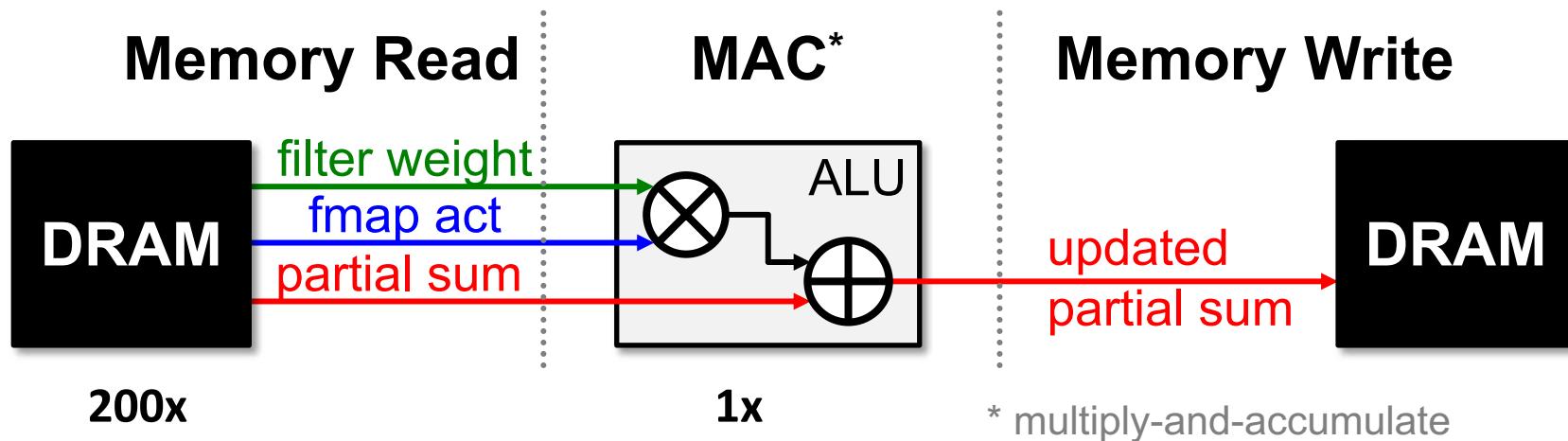
https://tinyurl.com/EfficientDNNBook

Properties We Can Leverage

- Operations exhibit **high parallelism**

  → **high throughput** possible

- Memory Access is the Bottleneck

Memory Read **MAC*** **Memory Write**

DRAM filter weight → ⊗ ALU → **DRAM**
fmap act →
partial sum → ⊕ → updated partial sum

200x 1x * multiply-and-accumulate
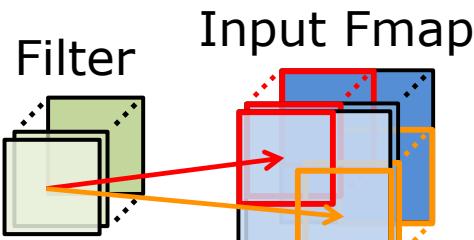
Worst Case: all memory R/W are **DRAM** accesses
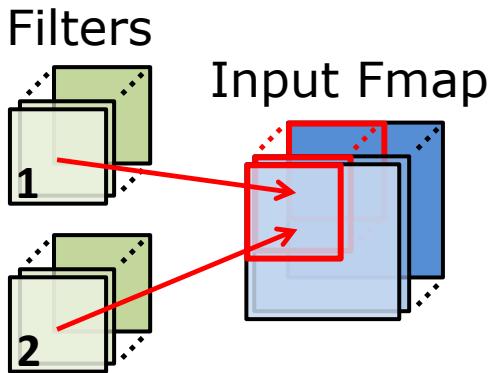
- Example: AlexNet has **724M** MACs

  → **2896M** DRAM accesses required

Properties We Can Leverage
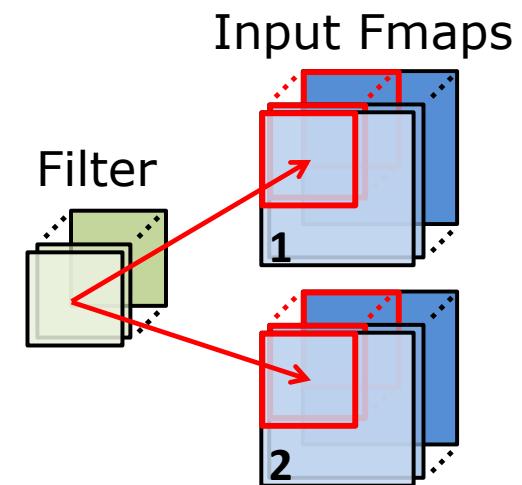
- Operations exhibit **high parallelism**
  - → **high throughput** possible

- **Input data reuse** opportunities (**up to 500x**)

Convolutional Reuse
(Activations, Weights)
CONV layers only
(sliding window)
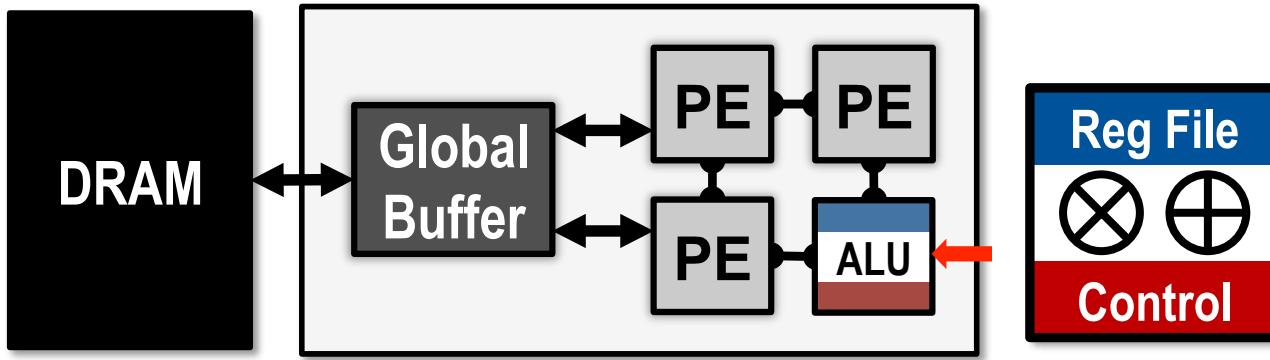
Fmap Reuse
(Activations)
CONV and FC layers
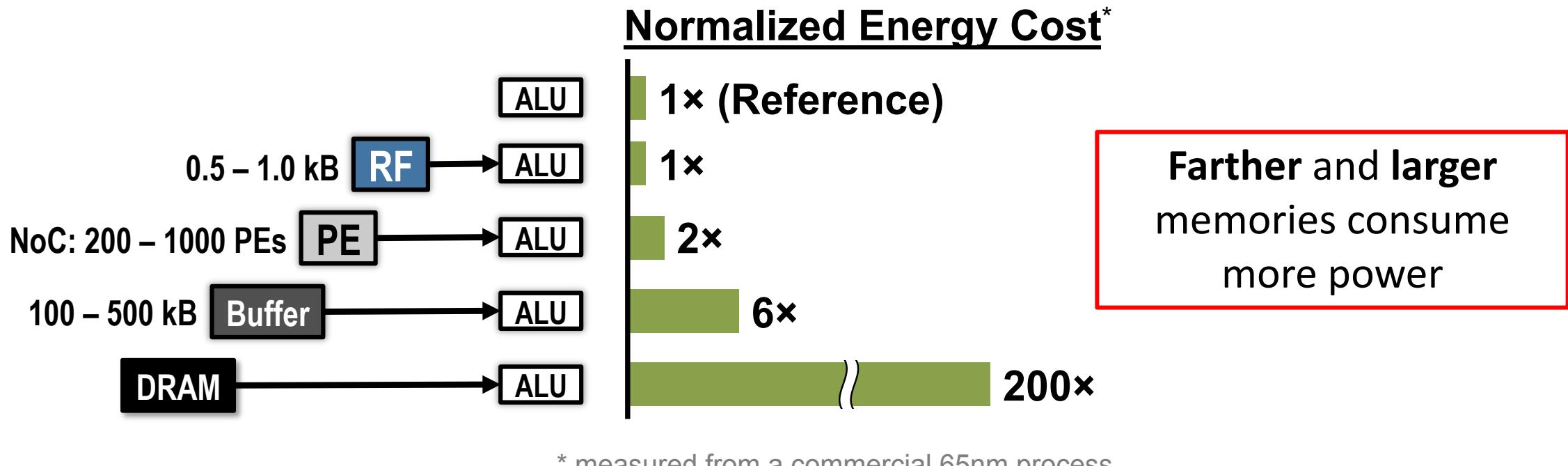
Filter Reuse
(Weights)
CONV and FC layers
(batch size > 1)

# Exploit Data Reuse at Low-Cost Memories

DRAM

Global Buffer

PE PE

PE ALU

Reg File
⊗ ⊕
Control
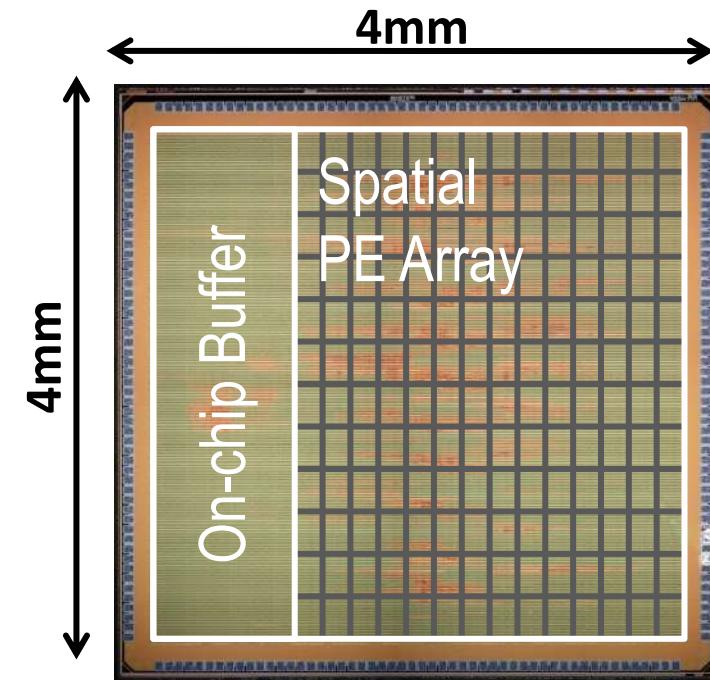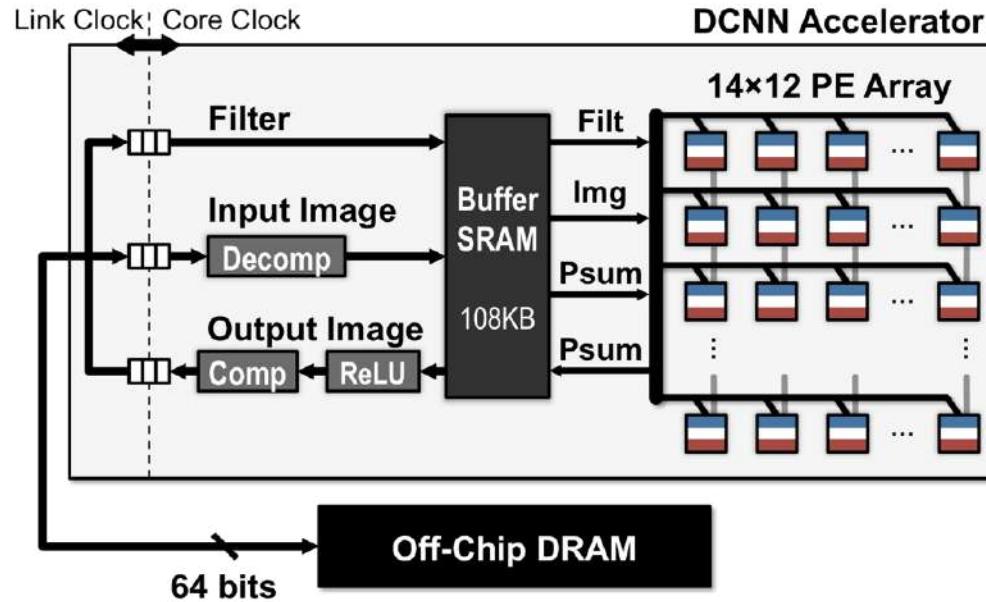
Specialized hardware with small (< 1kB) low cost memory near compute

Normalized Energy Cost*

	Memory		ALU	Cost
			ALU	1× (Reference)
0.5 – 1.0 kB	RF	→	ALU	1×
NoC: 200 – 1000 PEs	PE	→	ALU	2×
100 – 500 kB	Buffer	→	ALU	6×
	DRAM	→	ALU	200×

Farther and **larger** memories consume more power

* measured from a commercial 65nm process

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

MIT

Energy-Efficient Dataflow for Deep Neural Networks

Eyeriss: Row-Stationary Dataflow

[**Chen**, *ISSCC* 2016]
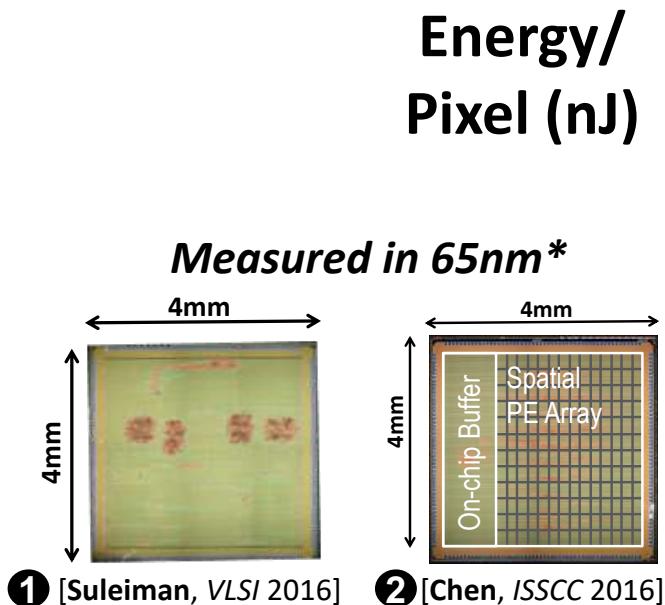
Exploits data reuse for **100x** reduction in memory accesses from global buffer and **1400x** reduction in memory accesses from off-chip DRAM

Overall **>10x energy reduction** compared to a mobile GPU (Nvidia TK1)

Results for AlexNet
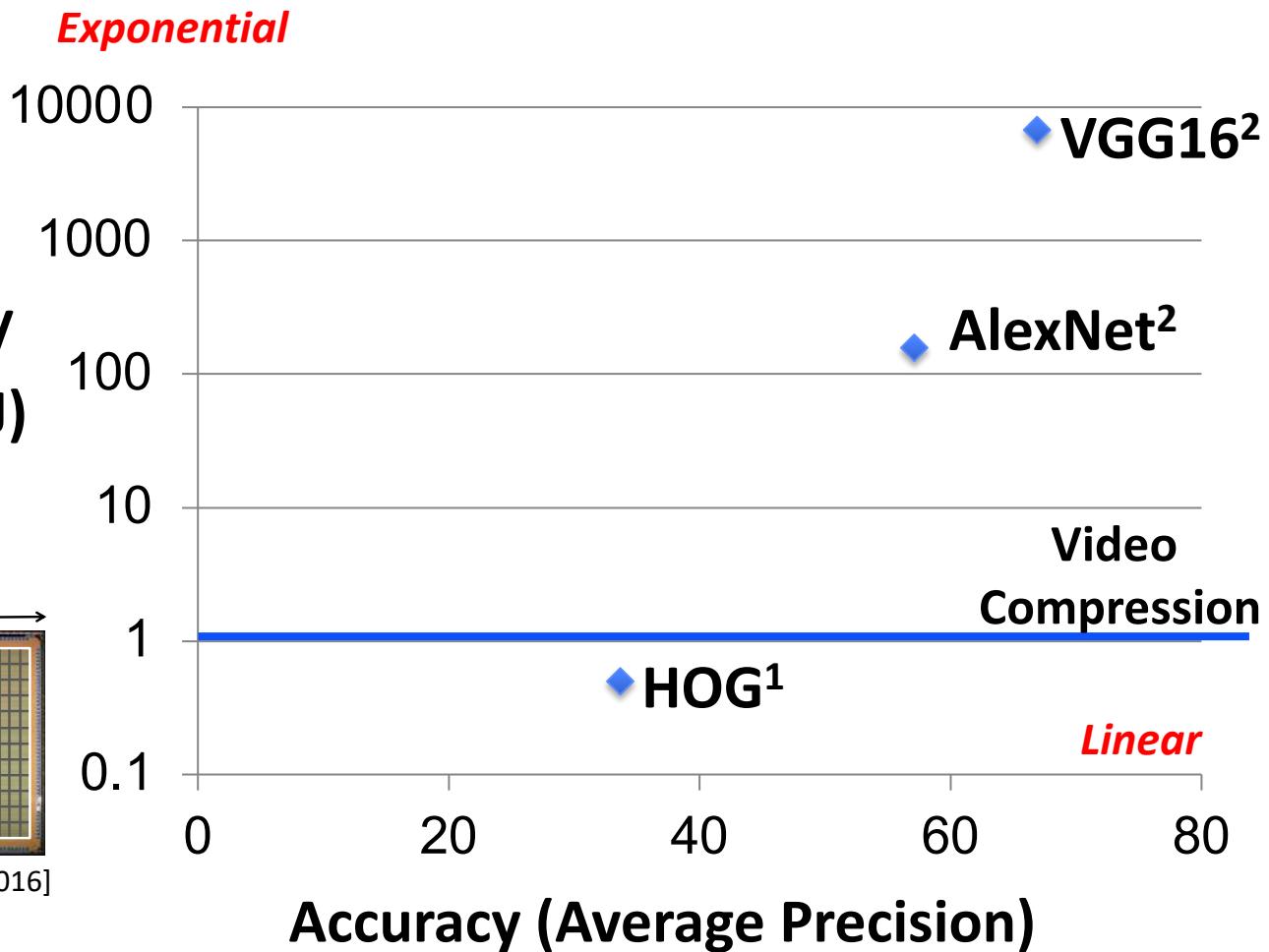
Eyeriss Project Website: http://eyeriss.mit.edu

Features: Energy vs. Accuracy

Exponential

Energy/Pixel (nJ) vs. Accuracy (Average Precision)

- VGG16[2] — ~6000 nJ, ~67 accuracy
- AlexNet[2] — ~150 nJ, ~57 accuracy
- Video Compression — ~1 nJ
- HOG[1] — ~0.5 nJ, ~34 accuracy

Linear

Measured in 65nm*

4mm × 4mm

On-chip Buffer | Spatial PE Array

❶ [**Suleiman**, *VLSI* 2016] ❷ [**Chen**, *ISSCC* 2016]

* Only feature extraction. Does not include data, classification energy, augmentation and ensemble, etc.
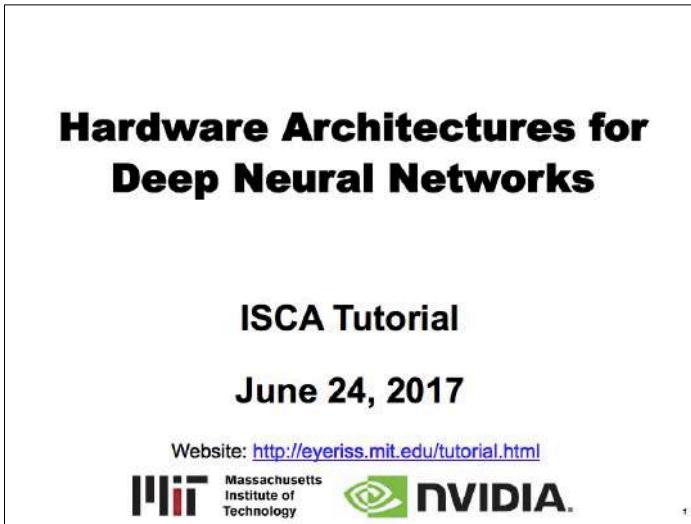
Measured in on VOC 2007 Dataset
1. DPM v5 [Girshick, 2012]
2. Fast R-CNN [Girshick, CVPR 2015]
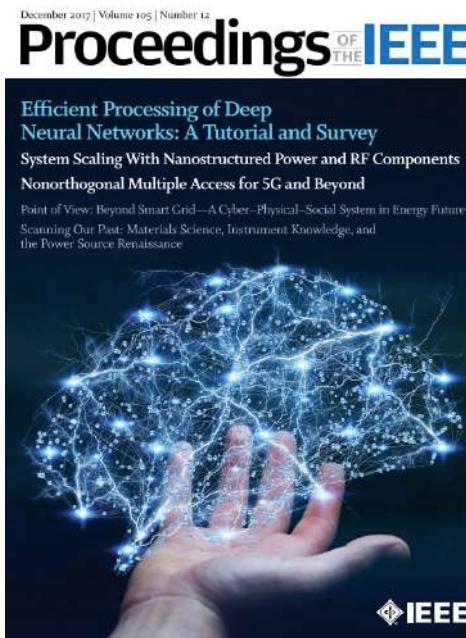
[**Suleiman**, *ISCAS* 2017]

Energy-Efficient Processing of DNNs

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

Hardware Architectures for Deep Neural Networks

ISCA Tutorial

June 24, 2017

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

Massachusetts Institute of Technology · NVIDIA.

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

Proceedings of the IEEE

December 2017 | Volume 105 | Number 12

Efficient Processing of Deep Neural Networks: A Tutorial and Survey
System Scaling With Nanostructured Power and RF Components
Nonorthogonal Multiple Access for 5G and Beyond
Point of View: Beyond Smart Grid—A Cyber–Physical–Social System in Energy Future
Scanning Our Past: Materials Science, Instrument Knowledge, and the Power Source Renaissance

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

We identified various limitations to existing approaches
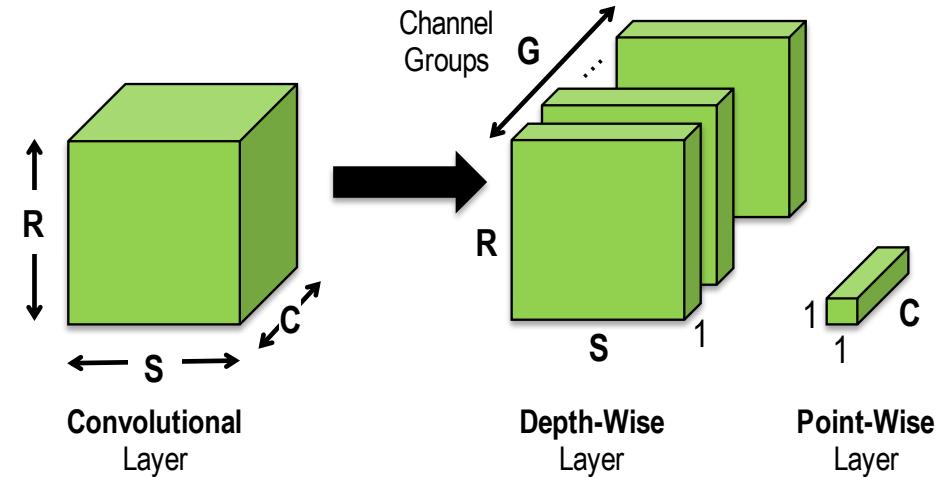
Design of Efficient DNN Algorithms

Popular efficient DNN algorithm approaches

Network Pruning

before pruning

after pruning

pruning synapses

pruning neurons

Efficient Network Architectures

Channel Groups

G

R

S

C

R

S

1

1

1

C

Convolutional Layer

Depth-Wise Layer

Point-Wise Layer
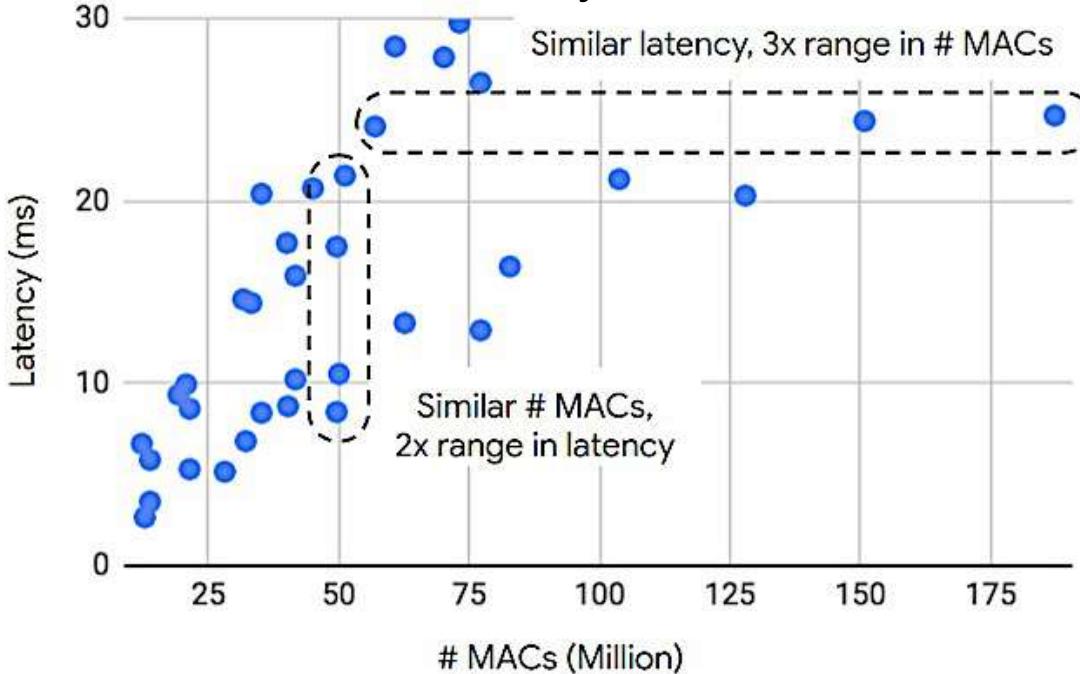
Examples: SqueezeNet, MobileNet

... also reduced precision

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

Number of MACs and Weights are Not Good Proxies
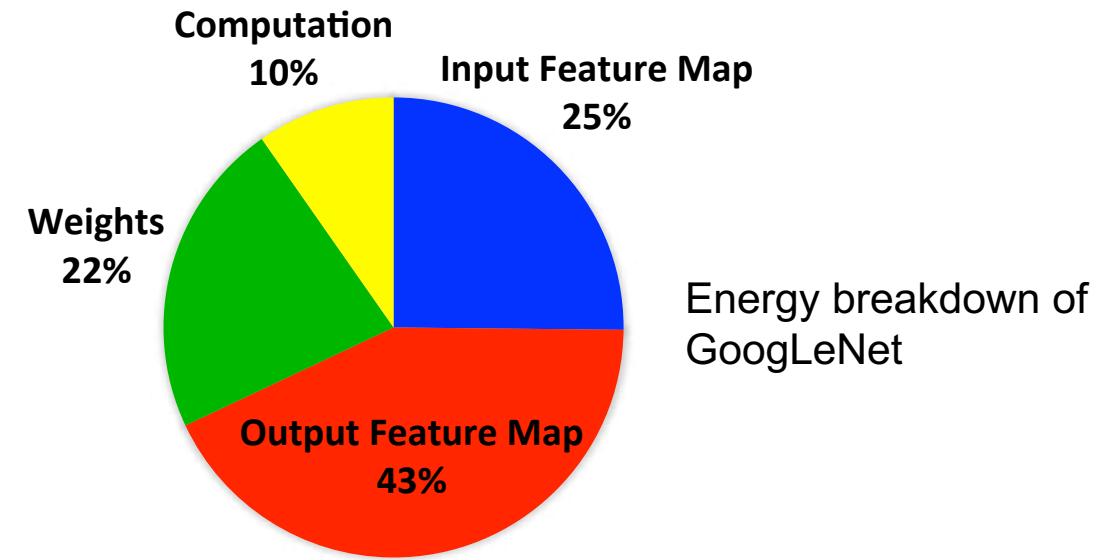
of operations (MACs) does not approximate latency well

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

# of weights *alone* is not a good metric for energy
(**All data types** should be considered)

Energy breakdown of GoogLeNet

https://energyestimation.mit.edu/

[**Yang**, *CVPR* 2017]

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

Energy-Aware Pruning

Normalized Energy (AlexNet)
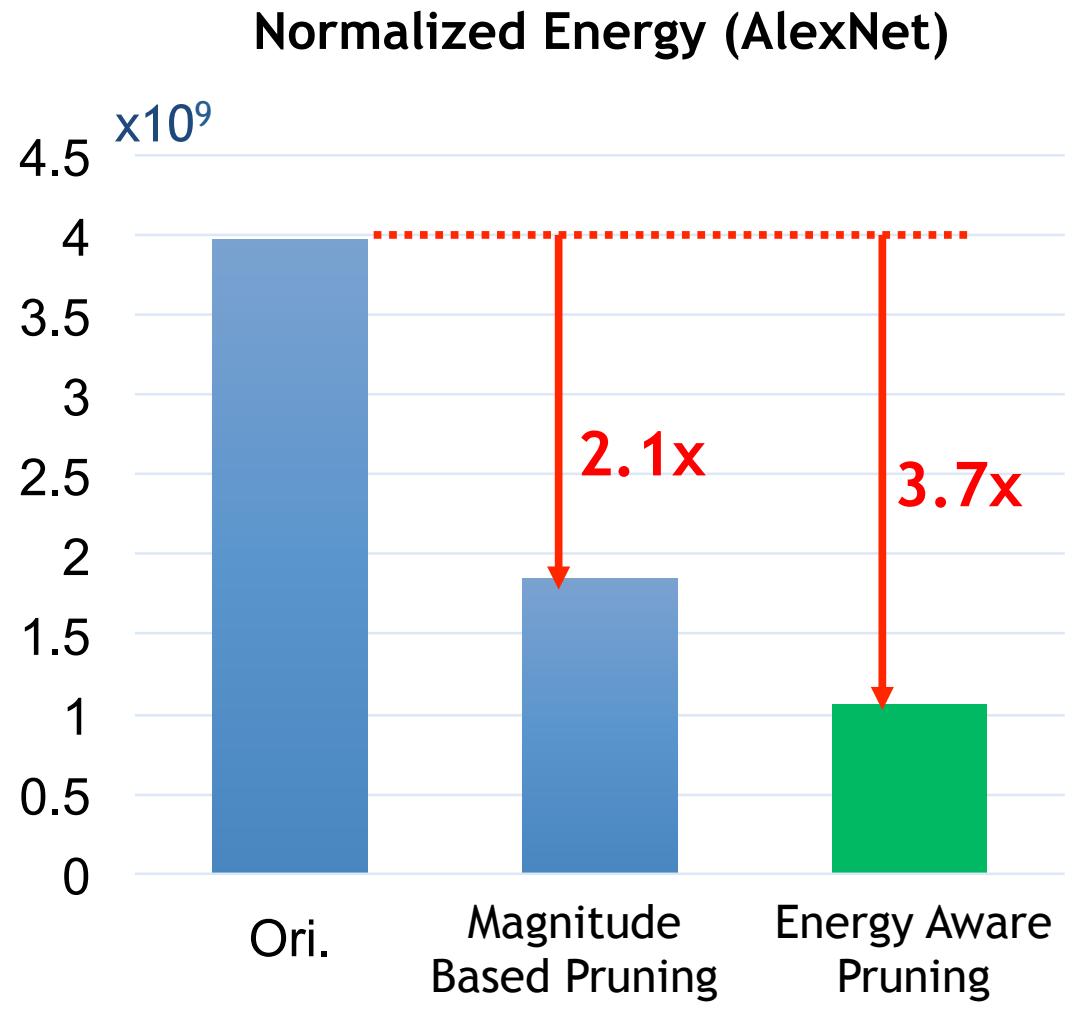
Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

- Sort layers based on energy and prune layers that consume the most energy first

- **Energy-aware pruning** reduces AlexNet energy by **3.7x** w/ similar accuracy

- Outperforms magnitude-based pruning by **1.7x**
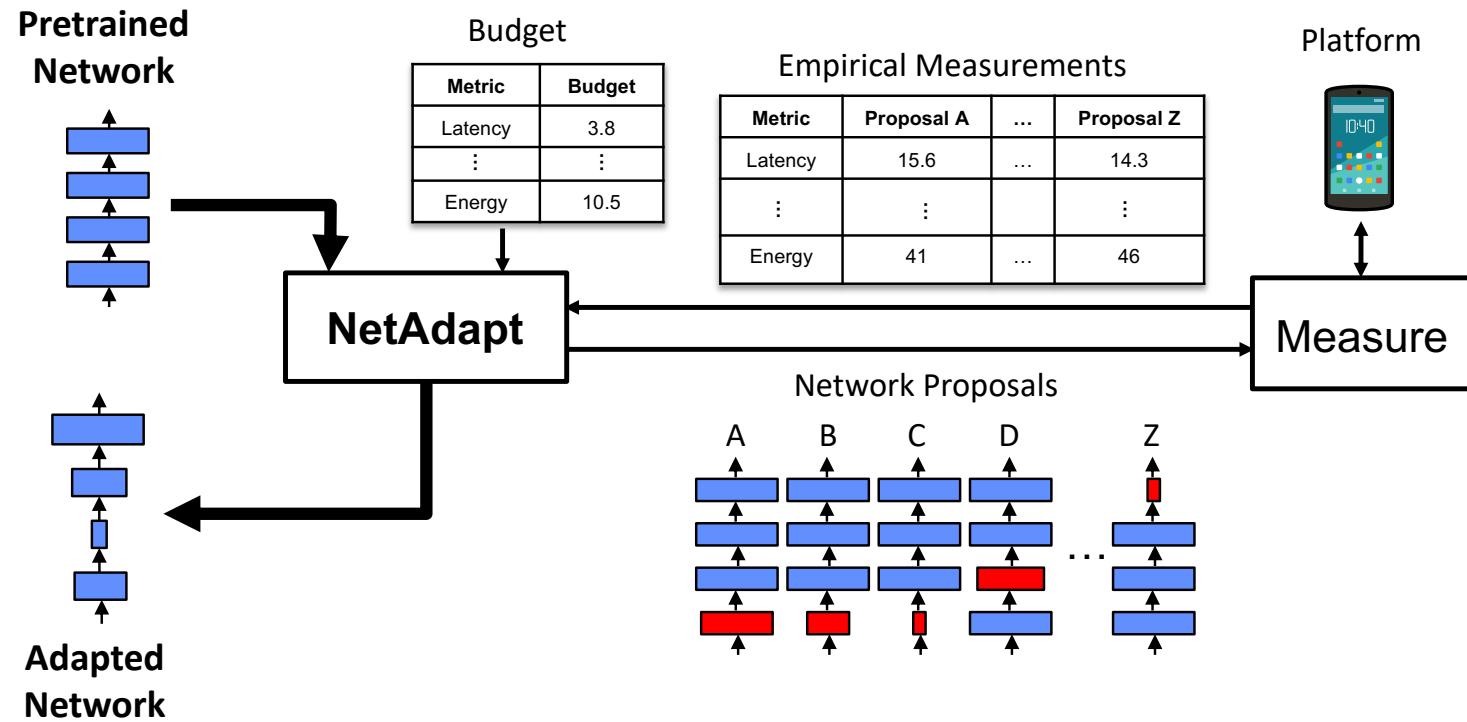
[**Yang**, *CVPR* 2017]

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

- **Few hyperparameters** to reduce tuning effort

- **>1.7x speed up** on MobileNet w/ similar accuracy

Pretrained Network

Budget

Metric	Budget
Latency	3.8
⋮	⋮
Energy	10.5

Empirical Measurements

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

Platform

NetAdapt

Measure

Network Proposals

A   B   C   D   Z

...

Adapted Network

Code available at
http://netadapt.mit.edu
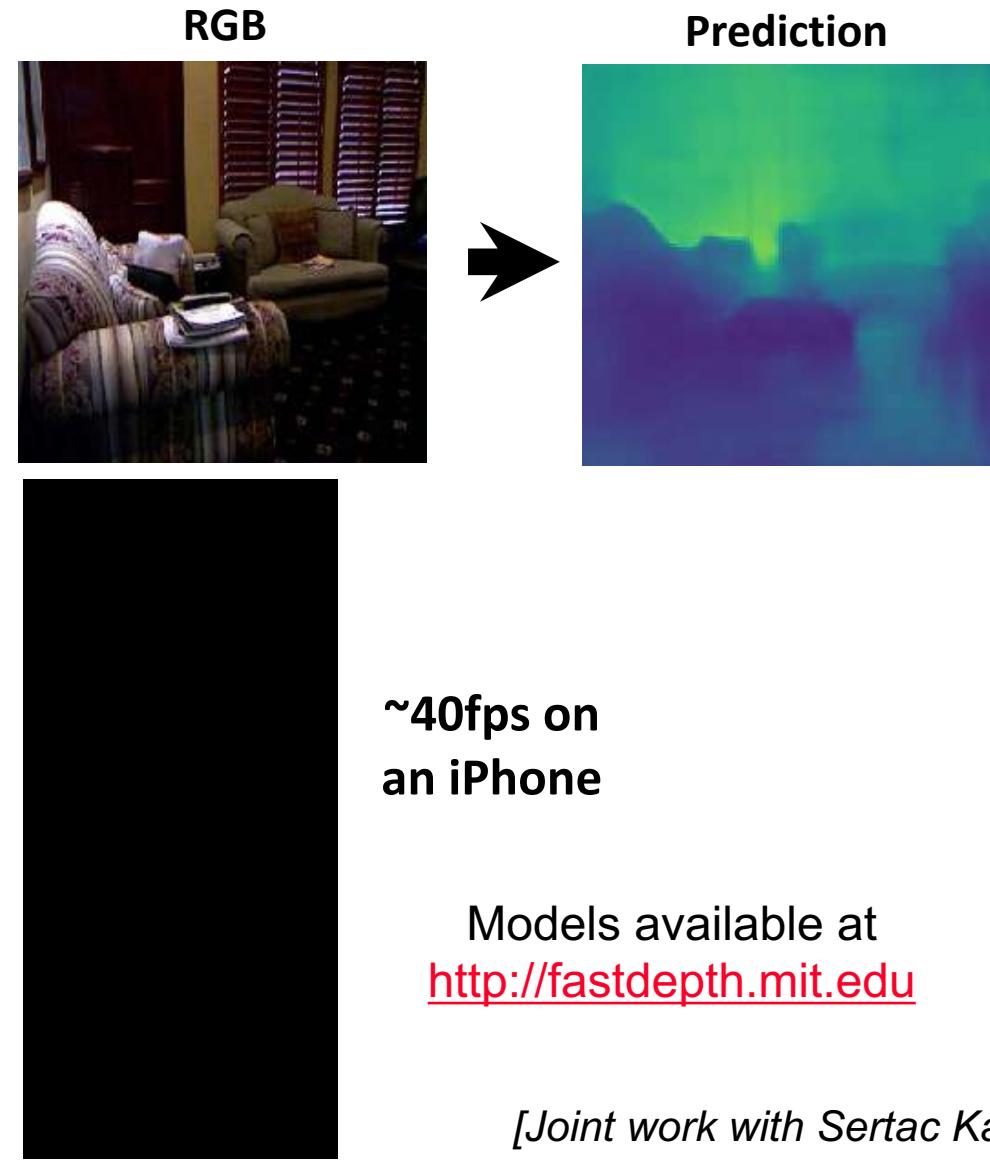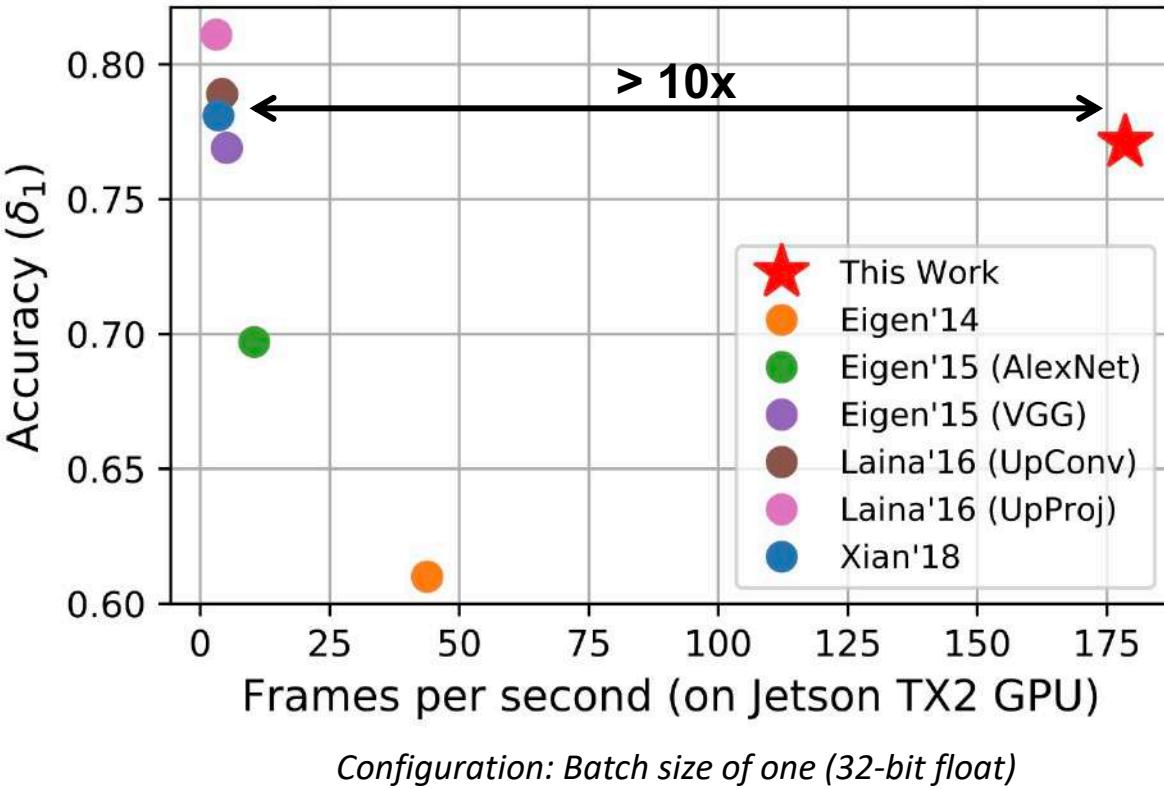
[In collaboration with Google's Mobile Vision Team]

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

> 10x

Accuracy (δ_1) vs Frames per second (on Jetson TX2 GPU)

- ★ This Work
- Eigen'14
- Eigen'15 (AlexNet)
- Eigen'15 (VGG)
- Laina'16 (UpConv)
- Laina'16 (UpProj)
- Xian'18

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

~40fps on an iPhone

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

[Joint work with Sertac Karaman]

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

[Wofk*, Ma*, *ICRA* 2019]

NetAdapt v2: Reduce Adaption Time
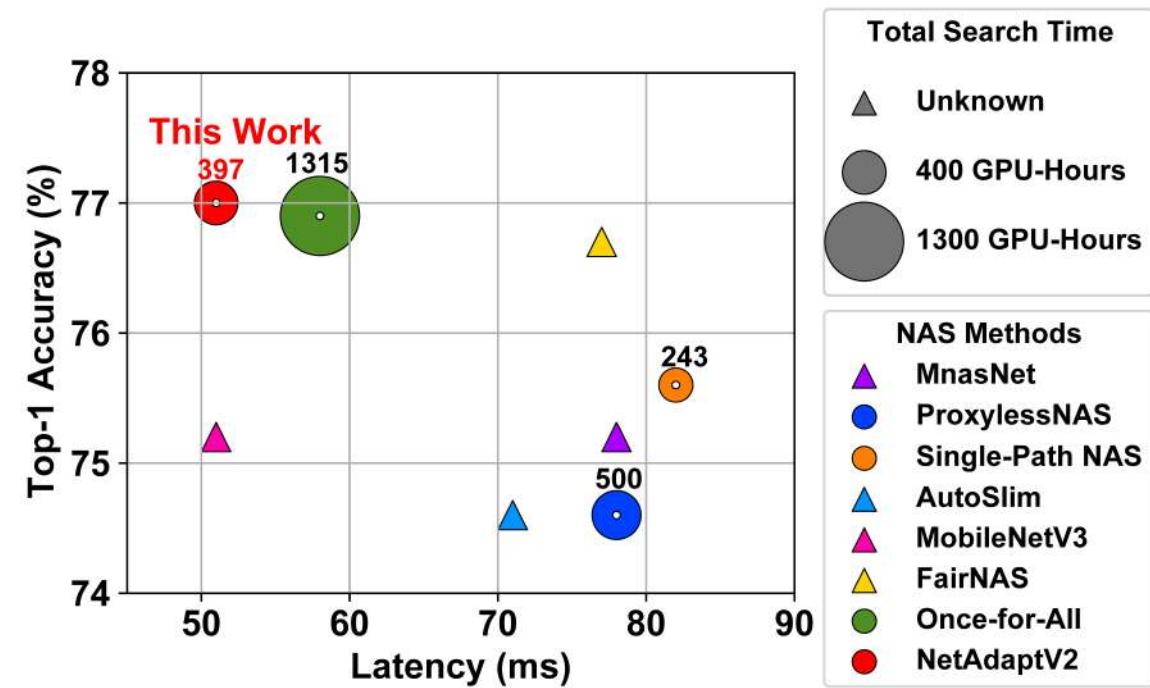
> Reduce time to find efficient DNN that adapts to hardware by up to 5.8x

Typical Steps in Neural Architecture Search (NAS):
1) Train super-network (search space of DNNs)
2) Sample and evaluate different DNNs
3) Fine tune the final DNN
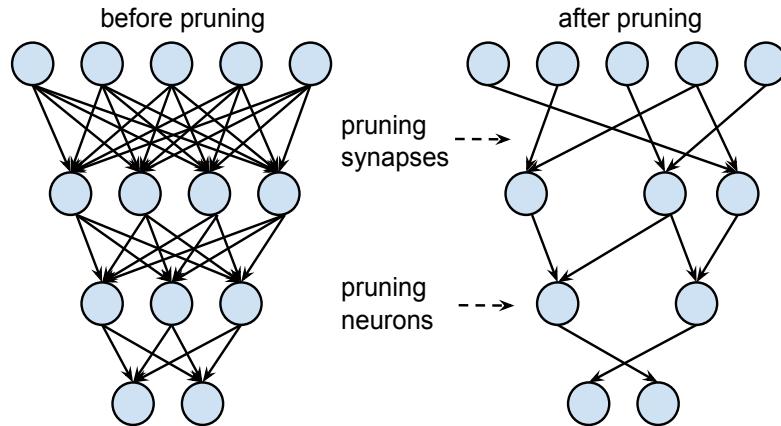
Contributions
- ***Ordered dropout:*** train multiple DNNs in *single* forward pass (reduce step 1)
- ***Channel-level bypass:*** merge layer depth and channel width into a *single* search dimension (reduce step 2)
- ***Multi-layer coordinate descent optimizer:*** consider joint effect of multiple layers (reduce step 2 & support non-differentiable metrics, e.g., latency)

More info at http://netadapt.mit.edu
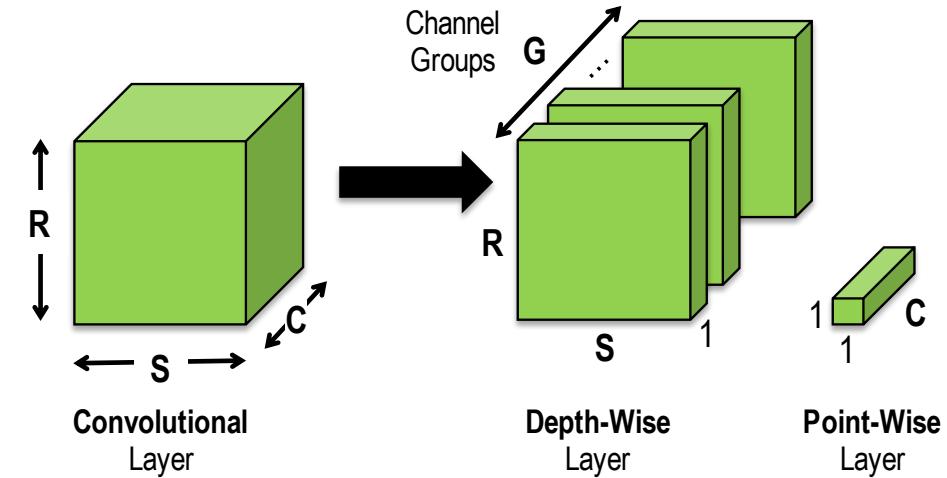
Many Efficient DNN Design Approaches

Network Pruning

before pruning

after pruning

pruning synapses

pruning neurons

Compact Network Architectures

Channel Groups

G

R

C

S

Convolutional Layer

R

S

1

Depth-Wise Layer

1

1

1

C

Point-Wise Layer

Reduce Precision

32-bit float 1010010101000000000101000000000100
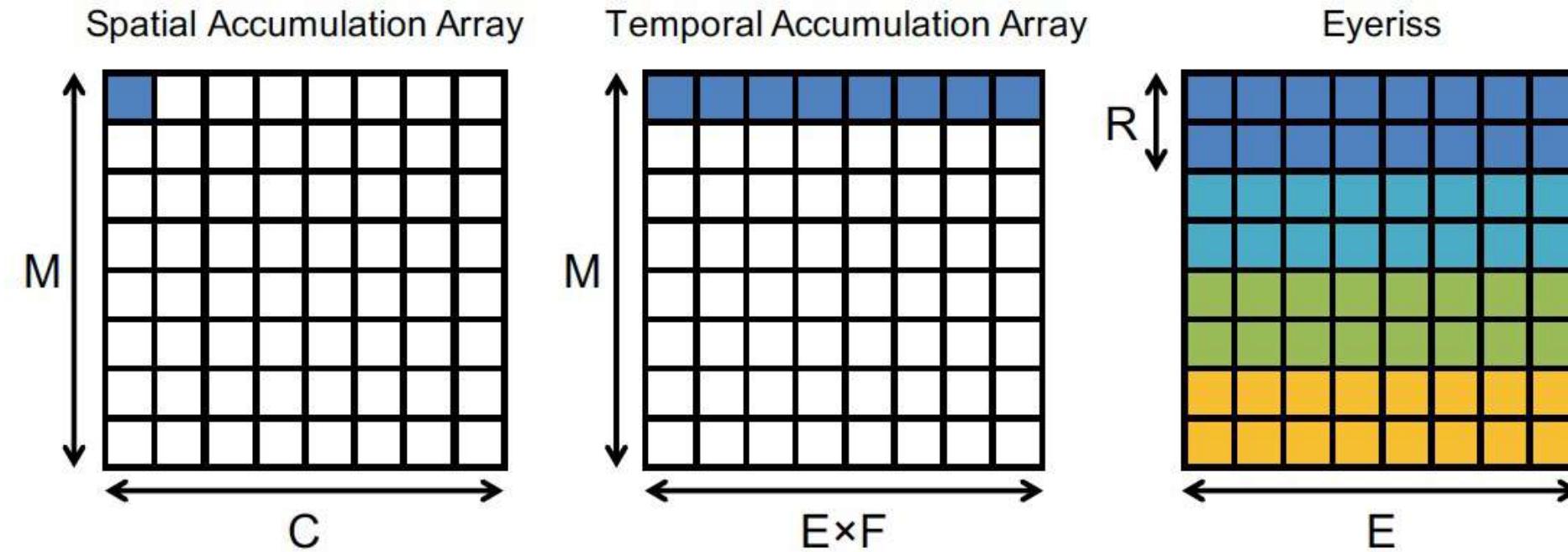
8-bit fixed 01100110

Binary 0

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

Need Flexible Dataflow & Mapping
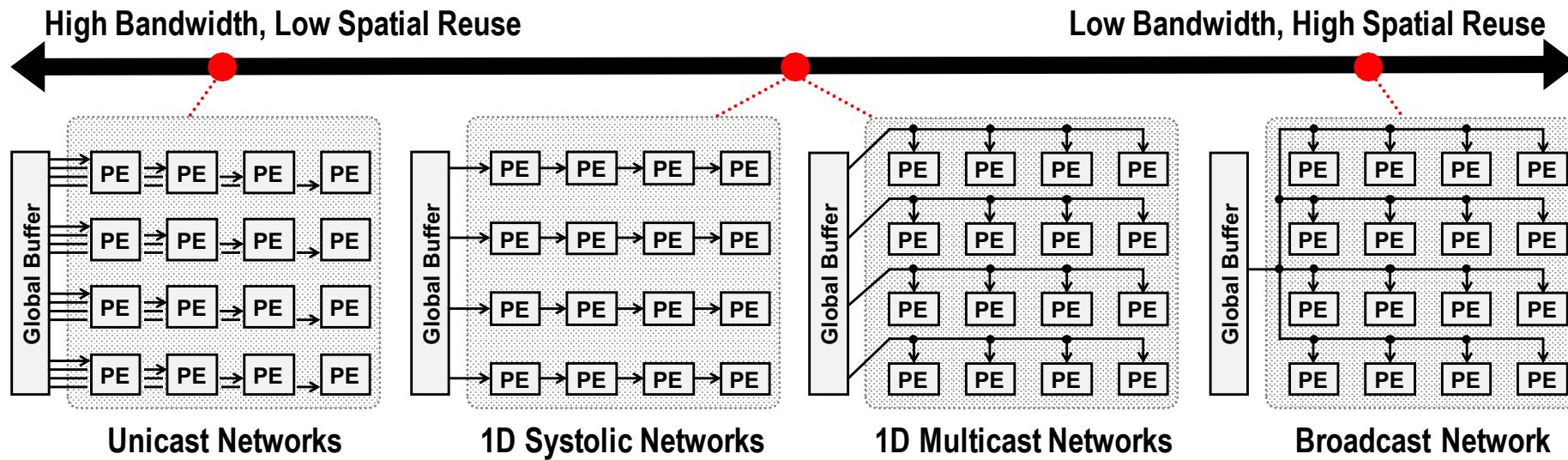
- Use flexible dataflow (**Row Stationary**) to exploit reuse in any dimension of DNN to increase energy efficiency and array utilization
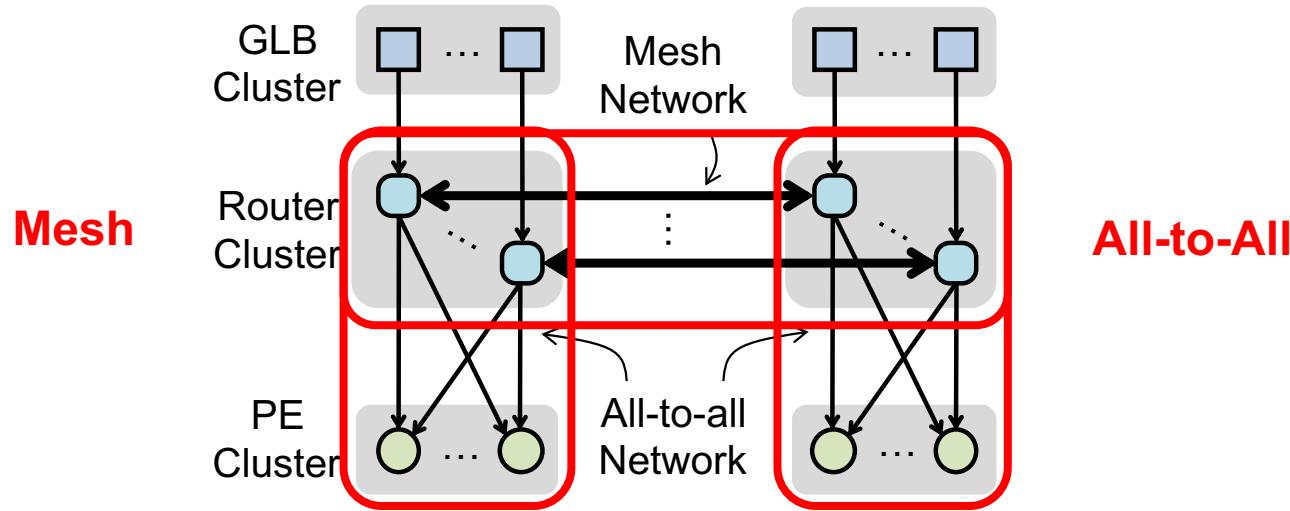
Example: Depth-wise layer

Need Flexible NoC for Varying Reuse
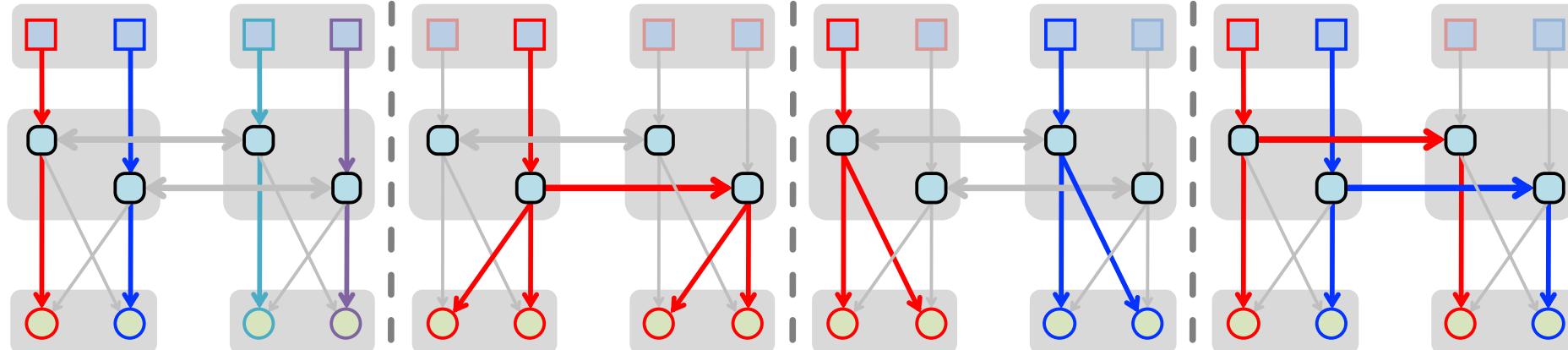
- When reuse available, need **multicast** to exploit spatial data reuse for energy efficiency and high array utilization

- When reuse not available, need **unicast** for high BW for weights for FC and weights & activations for high PE utilization

- An **all-to-all** satisfies above but too expensive and not scalable

High Bandwidth, Low Spatial Reuse ← → Low Bandwidth, High Spatial Reuse

Unicast Networks   1D Systolic Networks   1D Multicast Networks   Broadcast Network

Hierarchical Mesh

Mesh

GLB Cluster

Mesh Network

Router Cluster

PE Cluster

All-to-all Network

All-to-All

High Bandwidth

High Reuse

Grouped Multicast

Interleaved Multicast
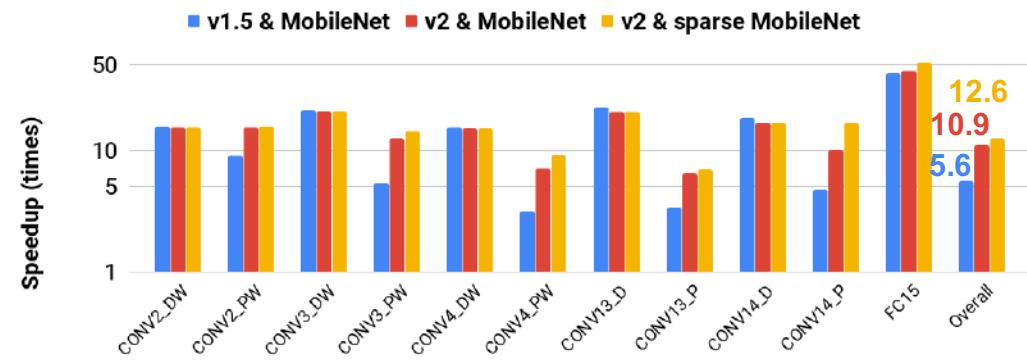
Eyeriss v2: Balancing Flexibility and Efficiency

- **Uses a flexible hierarchical mesh on-chip network to efficiently support**
  - Wide range of filter shapes
  - Different layers
  - Wide range of sparsity

- **Scalable architecture**

Over an order of magnitude faster and more energy efficient than Eyeriss v1
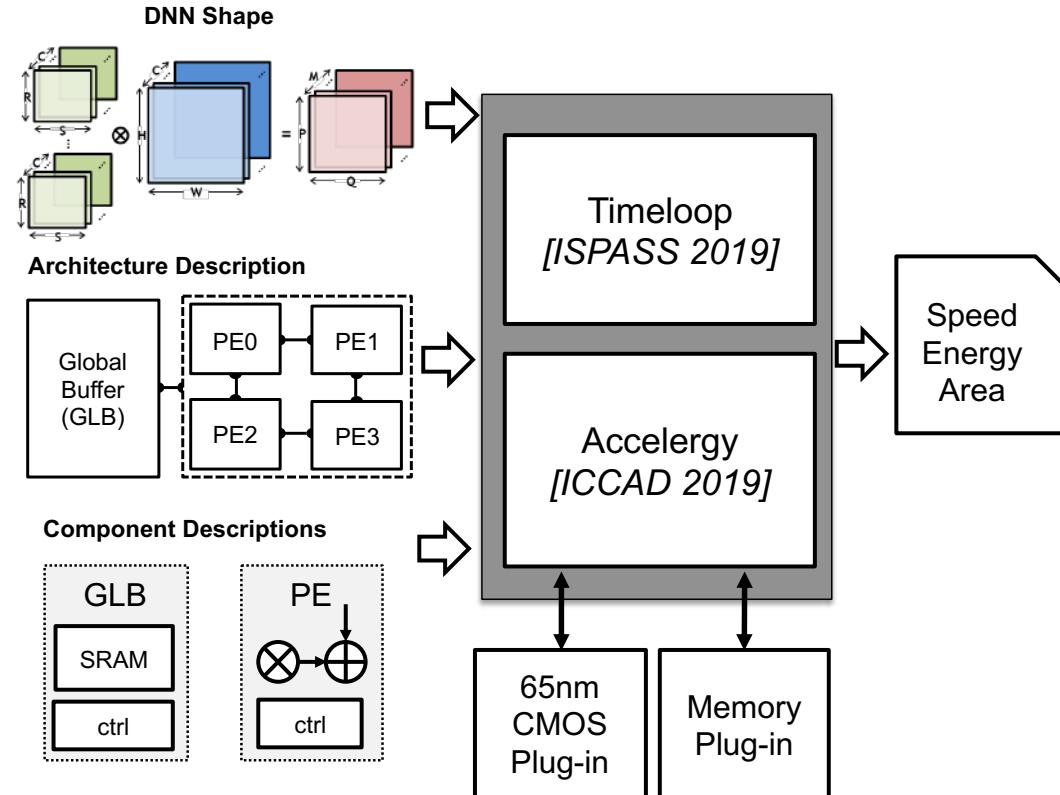
Speed up over Eyeriss v1 scales with number of PEs

# of PEs	256	1024	16384
AlexNet	17.9x	71.5x	1086.7x
GoogLeNet	10.4x	37.8x	448.8x
MobileNet	15.7x	57.9x	873.0x

[Joint work with Joel Emer]

DNN Accelerator Evaluation Tools
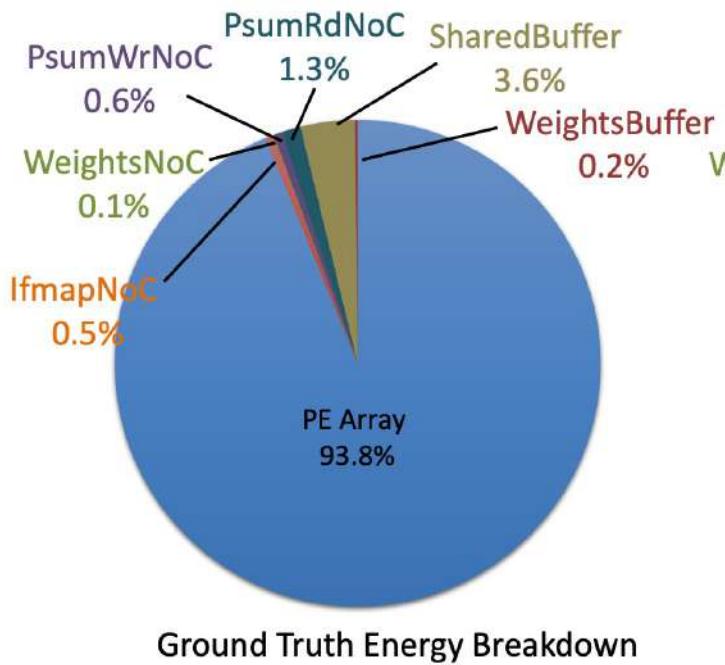
- Require systematic way to
  - Evaluate and compare DNN accelerators
  - Rapidly explore design space

- Accelergy [**Wu**, *ICCAD* 2019]
  - Early-stage estimation tool at the architecture level
    - Estimate energy based on architecture level components (e.g., # of PEs, memory size, on-chip network)
  - Evaluate architecture level impact of emerging devices
    - Plug-ins for different technologies

- Timeloop [**Parashar**, *ISPASS* 2019]
  - DNN mapping tool
  - Performance Simulator → Action counts

- Bridge architecture, circuit, and device research

Open-source code available at:
http://accelergy.mit.edu
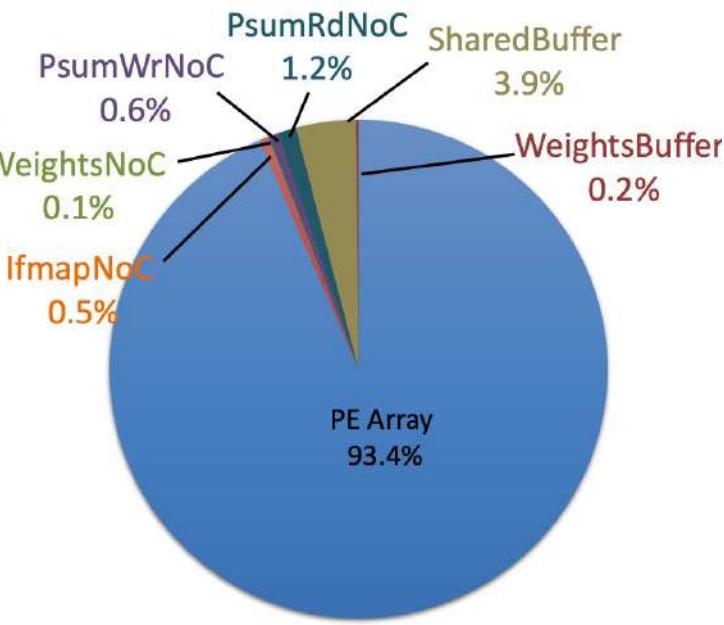
Accelergy Estimation Validation

- Validation on Eyeriss [**Chen**, *ISSCC* 2016]
  - Achieves 95% accuracy compared to post-layout simulations
  - Can accurately captures energy breakdown at different granularities
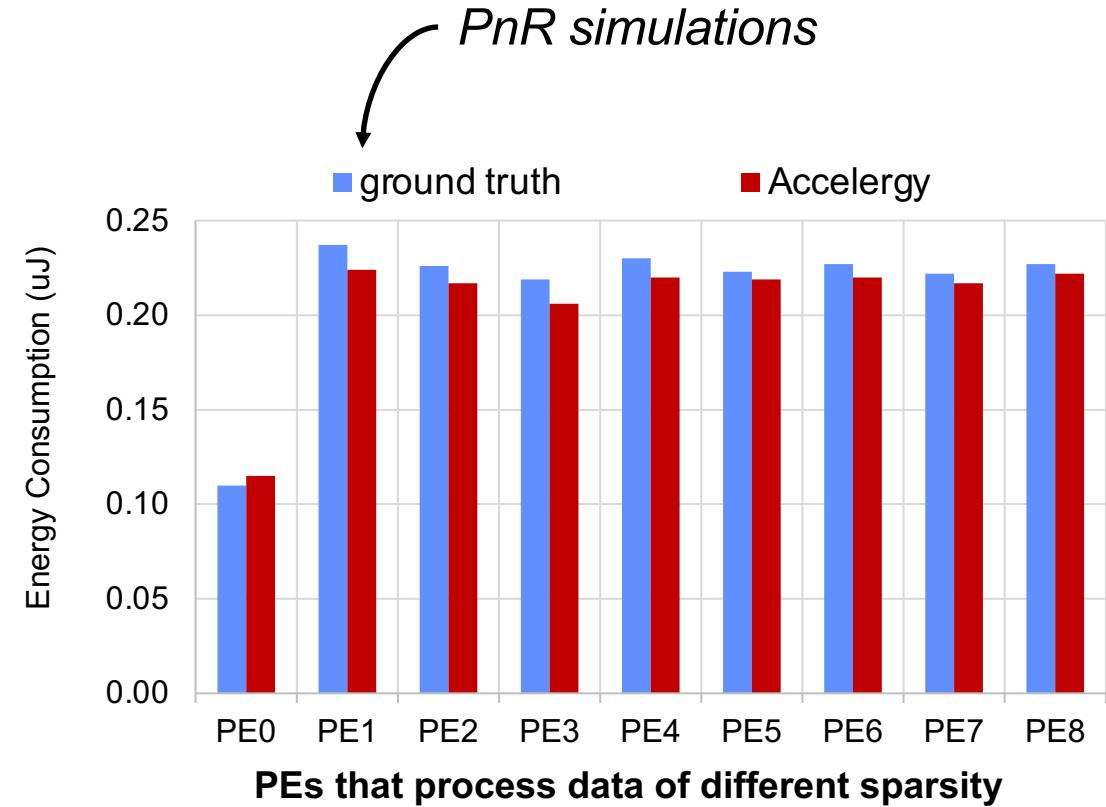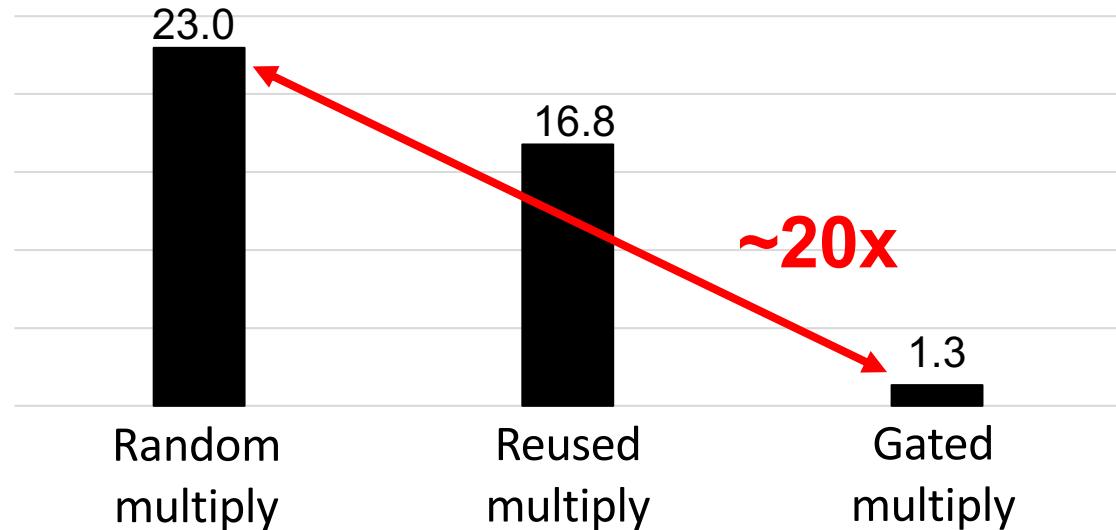
*Total energy might not add up to exact 100.0% due to rounding

Open-source code available at: http://accelergy.mit.edu

Plug-ins for Fine-Grain Action Energy Estimation

- External energy/area models that accurately reflect the properties of a macro
  - e.g., multiplier with zero-gating

PnR simulations
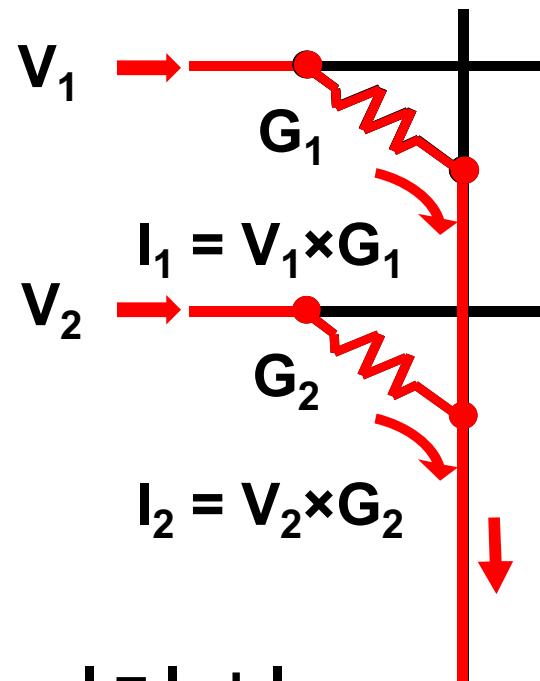
Energy characterizations of the zero-gated multiplier (normalized to idle)

23.0 — Random multiply

16.8 — Reused multiply

~20x

1.3 — Gated multiply

■ ground truth ■ Accelergy

Energy Consumption (uJ)

0.25
0.20
0.15
0.10
0.05
0.00

PE0 PE1 PE2 PE3 PE4 PE5 PE6 PE7 PE8

PEs that process data of different sparsity

With the characterization provided in the plug-in,
we can capture the energy savings for sparse workloads

In-Memory Computing (IMC*)

Activation is input voltage (V_i)
Weight is resistor conductance (G_i)

V_1

G_1

$I_1 = V_1 \times G_1$

V_2

G_2

$I_2 = V_2 \times G_2$
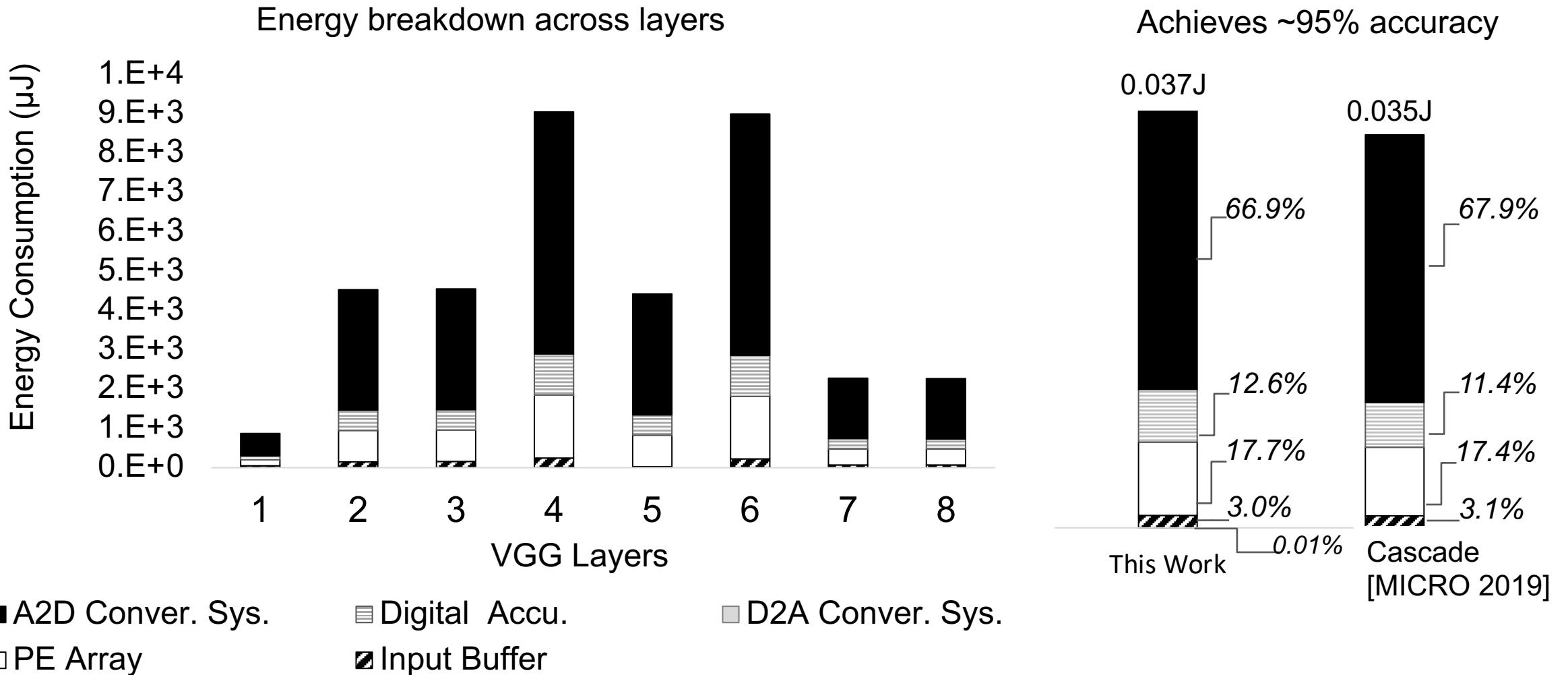
Psum is output current

$I = I_1 + I_2$
$= V_1 \times G_1 + V_2 \times G_2$

Image Source: [**Shafiee**, *ISCA* 2016]

- Reduce data movement by **moving compute into memory**

- Compute MAC with memory storage element

- **Analog Compute**
  - Activations, weights and/or partial sums are encoded with analog voltage, current, or resistance
  - Increased sensitivity to circuit non-idealities
  - A/D and D/A circuits to interface with digital domain

- Leverage **emerging memory device technology**

Accelergy for IMC

Energy breakdown across layers

Achieves ~95% accuracy

- ■ A2D Conver. Sys.
- ▤ Digital Accu.
- ▦ D2A Conver. Sys.
- ☐ PE Array
- ▨ Input Buffer
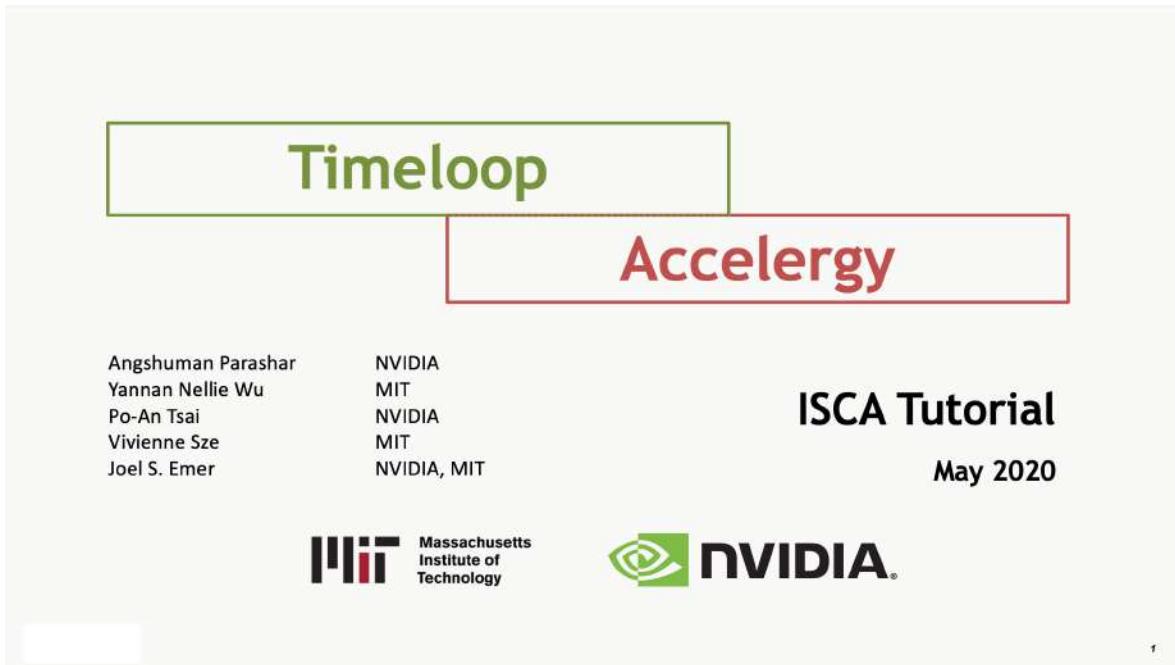
Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit   [**Wu**, *ISPASS* 2020]   MIT

Accelergy + Timeloop Tutorial

Tutorial material available at http://accelergy.mit.edu/tutorial.html
Includes videos and hands-on exercises

Designing DNNs for IMC

- Designing DNNs for IMC may differ from DNNs for digital processors

- Highest accuracy DNN on digital processor may be different on IMC
  - Accuracy drops based on robustness to non-idealities

- Reducing number of weights is less desirable
  - Since IMC is weight stationary, may be better to reduce number of activations
  - IMC tend to have larger arrays → fewer weights may lead to low utilization on IMC

Book Chapter on In-Memory Computing

253

CHAPTER 10

Advanced Technologies

As highlighted throughout the previous chapters, data movement dominates energy consumption. The energy is consumed both in the access to the memory as well as the transfer of the data. The associated physical factors also limit the bandwidth available to deliver data between memory and compute, and thus limits the throughput of the overall system. This is commonly referred to by computer architects as the "memory wall."[1]

To address the challenges associated with data movement, there have been various efforts to bring compute and memory closer together. Chapters 5 and 6 primarily focus on how to design spatial architectures that distribute the on-chip memory closer to the computation (e.g., scratch pad memory in the PE). This chapter will describe various other architectures that use *advanced memory, process,* and *fabrication technologies* to bring the compute and memory together.

First, we will describe efforts to bring the off-chip high-density memory (e.g., DRAM) closer to the computation. These approaches are often referred to as *processing near memory* or *near-data processing,* and include memory technologies such as embedded DRAM and 3-D stacked DRAM.

Next, we will describe efforts to integrate the computation *into* the memory itself. These approaches are often referred to as *processing in memory* or *in-memory computing,* and include memory technologies such as Static Random Access Memories (SRAM), Dynamic Random Access Memories (DRAM), and emerging non-volatile memory (NVM). Since these approaches rely on mixed-signal circuit design to enable processing in the analog domain, we will also discuss the design challenges related to handling the increased sensitivity to circuit and device non-idealities (e.g., nonlinearity, process and temperature variations), as well as the impact on area density, which is critical for memory.

Significant data movement also occurs between the sensor that collects the data and the DNN processor. The same principles that are used to bring compute near the memory, where the weights are stored, can be used to bring the compute *near* the sensor, where the input data is collected. Therefore, we will also discuss how to integrate some of the compute *into* the sensor.

Finally, since photons travel much faster than electrons and the cost of moving a photon can be *independent* of distance, processing in the optical domain using light may provide significant improvements in energy efficiency and throughput over the electrical domain. Accordingly, we will conclude this chapter by discussing the recent work that performs DNN processing in the optical domain, referred to as *Optical Neural Networks.*

[1] Specifically, the memory wall refers to data moving between the off-chip memory (e.g., DRAM) and the processor.
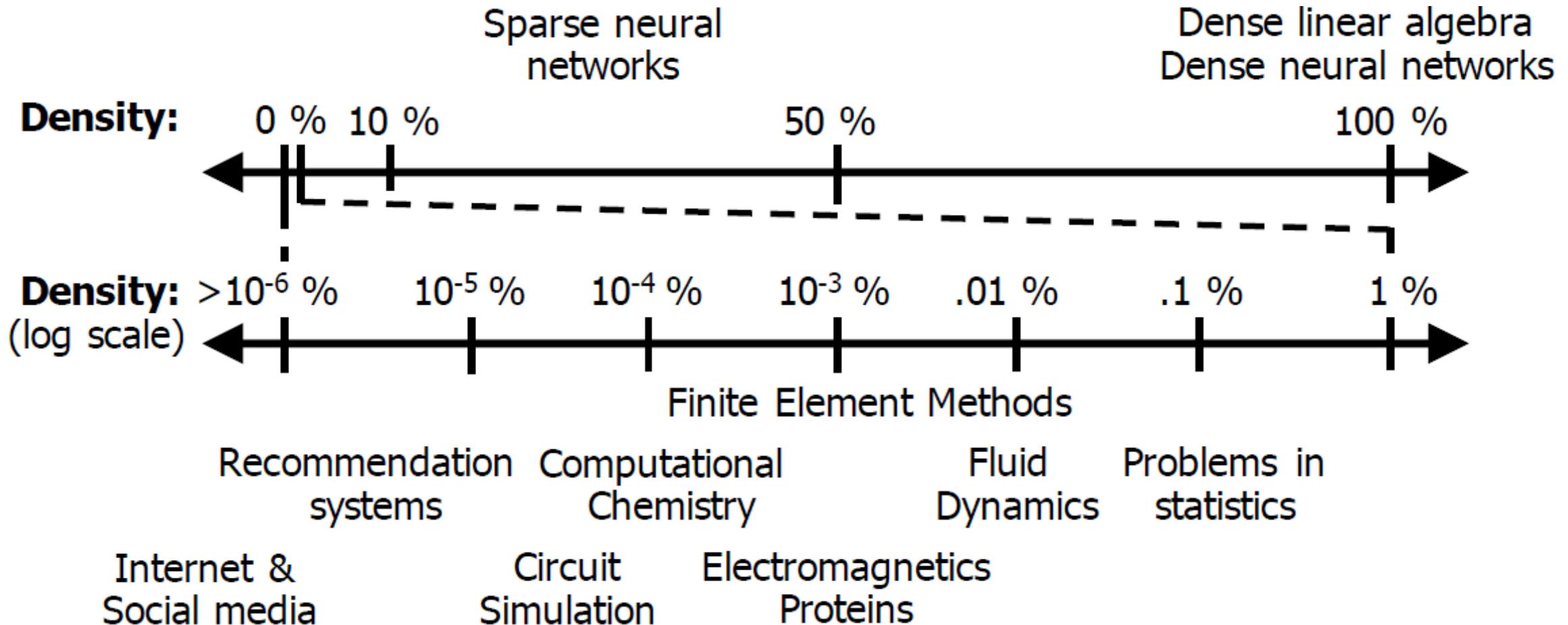
Many Design Considerations for In-Memory Computing

- Number of Storage Elements per Weight
- Array Size
- Number of Rows Activated in Parallel
- Number of Columns Activated in Parallel
- Time to Deliver Input
- Time to Compute MAC

Tradeoffs between energy efficiency, throughput, area density, and accuracy, which *reduce the achievable gains over conventional architectures*

Available on DNN tutorial website
http://eyeriss.mit.edu/tutorial.html
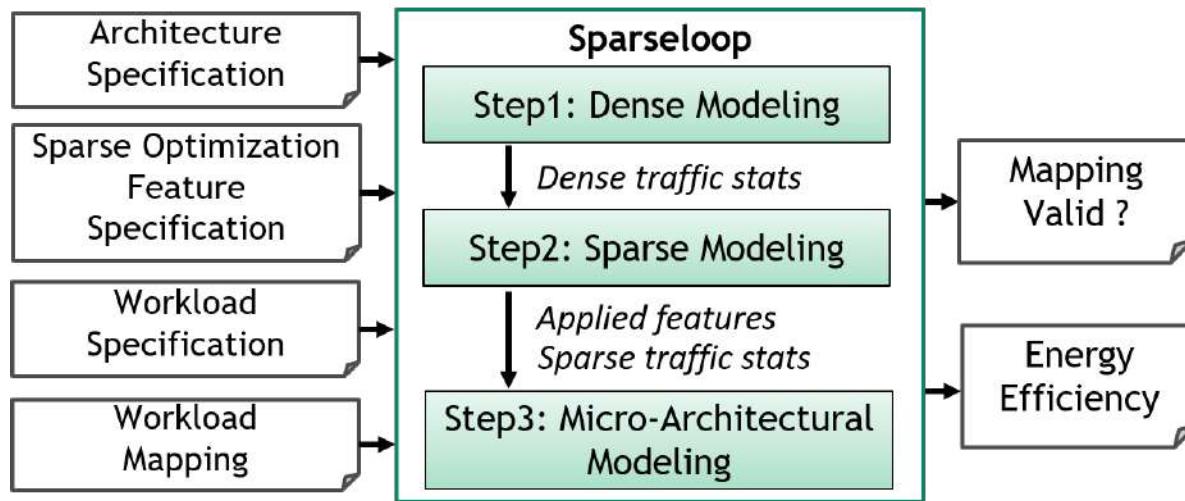
Applications that use Sparse Tensor

[**Hedge**, *MICRO* 2019]
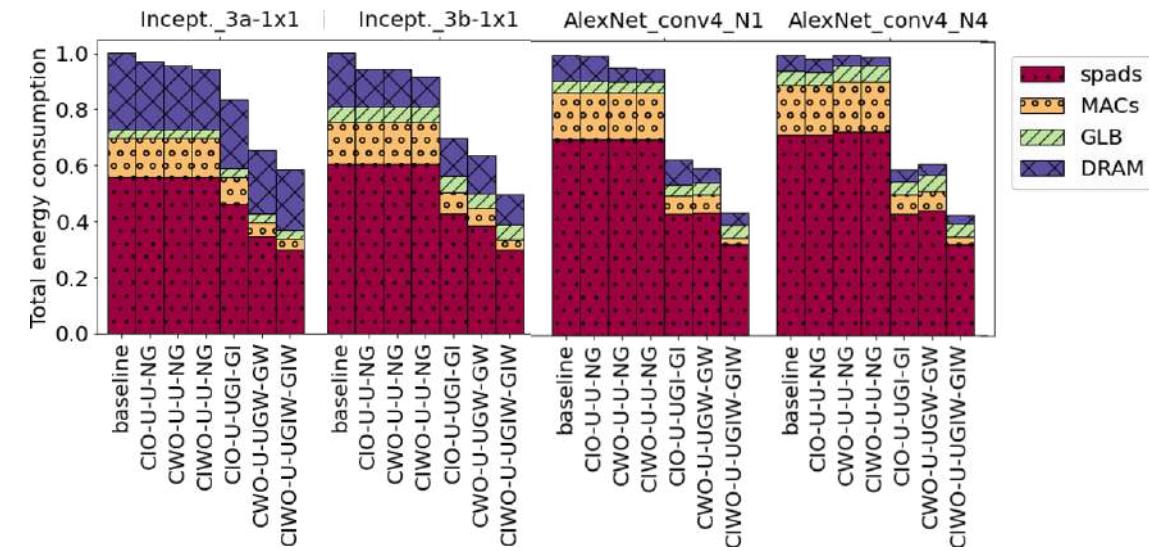
Sparseloop: Design Space Exploration for Sparse Tensor Accelerators

- An analytical design exploration framework that comprehends a wide range of
  - Sparse optimizations (e.g., zero-gating, zero-skipping, zero-compression)
  - Data representations (e.g., uncompressed, run length coding, bitmask)

Propose modularized three-step evaluation process

Energy impact of sparse optimizations at different levels of the memory hierarchy in Eyeriss-based topology

Tutorial at ISCA 2021 (June 19): http://accelergy.mit.edu/sparse_tutorial.html

Book Chapter on Sparse Computations

CHAPTER 8

Exploiting Sparsity

A salient characteristic of the data used in DNN computations is that it is (or can be made to be) sparse. By saying that the data is sparse, we are referring to the fact that there are many repeated values in the data. Much of the time the repeated value is zero, which is what we will assume unless explicitly noted. Thus, we will talk about the sparsity or density of the data as the percentage of zeros or non-zeros, respectively in the data. The existence of sparse data leads broadly to two potential architectural benefits: (1) sparsity can reduce the footprint of the data, which provides an opportunity to reduce storage requirements and data movement. This is because sparse data is amenable to being compressed, as described in Section 8.2;[1] and (2) sparsity presents an opportunity for a reduction in MAC operations. The reduction in MAC operations results from the fact that $0 \times \textit{anything}$ is 0. This can result in either savings in energy or time or both. In Section 8.3, we will discuss how the dataflows for sparse data can translate sparsity into improvements in energy-efficiency and throughput. However, first in Section 8.1 we discuss the origins and ways that one can increase sparsity in the data used in DNN computations.

8.1   SOURCES OF SPARSITY

Efficient processing of feature map activations becomes increasingly important as the size of the input to the DNN model grows (e.g., increased image resolution), while efficient processing of filter weights becomes increasingly important as the size of the DNN model grows (e.g., increased number of layers).

This section will discuss various approaches that can exploit properties such as redundancy and correlation in the feature maps and filters to increase their activation sparsity (Section 8.1.1) and weight sparsity (Section 8.1.2), respectively. The requirements for these approaches may differ as activation sparsity is often data dependent and not known a priori, while weight sparsity can be known a priori. As a result, methods to increase sparsity for weights can be performed offline (as opposed to during inference) and can be more computationally complex than methods applied to increase activation sparsity. For instance, increasing weight sparsity can be incorporated into training.

[1]Note: We use the words sparsity or density to refer to a statistical property of the data, while we use the words compressed or uncompressed to describe the characteristics of a representation of the (typically sparse) data.

- Sources of Sparsity

- Compression and Sparse Tensor Representation

- Sparse Dataflows

https://tinyurl.com/EfficientDNNBook

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit
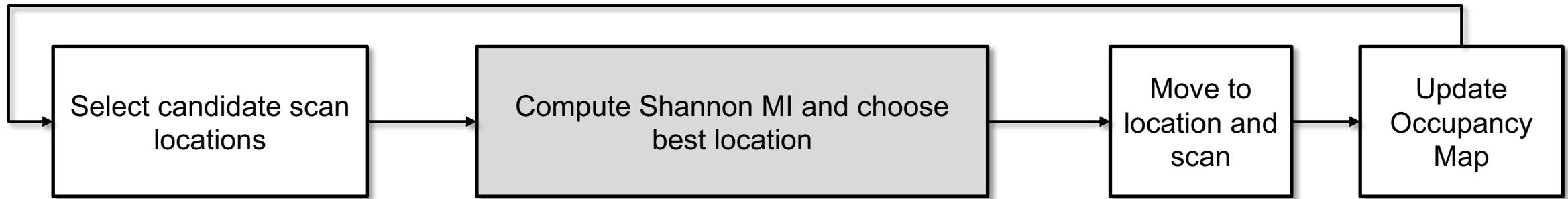
Where to Go Next: Planning and Mapping
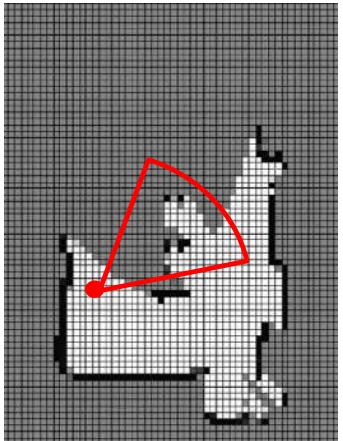
Robot Exploration

Where to Go Next: Planning and Mapping
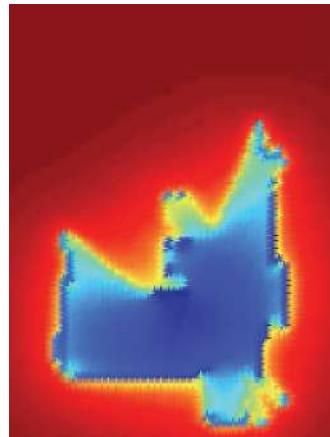
*Robot Exploration: Decide where to go by computing Shannon Mutual Information*

Select candidate scan locations → Compute Shannon MI and choose best location → Move to location and scan → Update Occupancy Map
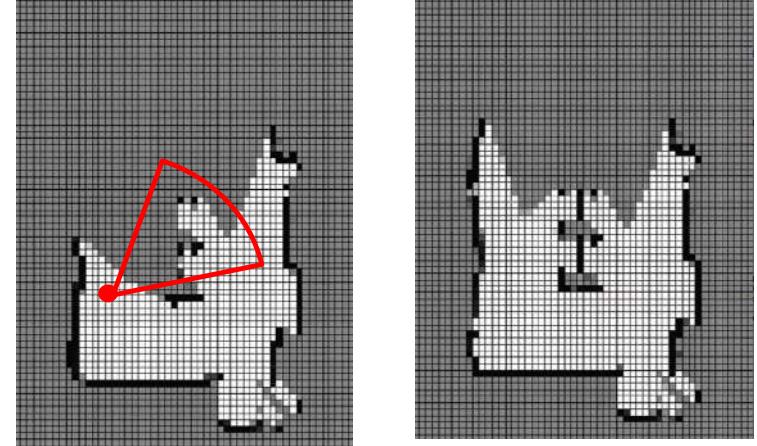
Where to scan?   Mutual Information   Updated Map

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit   *[Joint work with Sertac Karaman]*

Experimental Results (4x Real Time)

Occupancy map with planned path using RRT* (compute MI on all possible paths)

MI surface

Exploration with a mini race car using motion capture for localization

Building Hardware Accelerator to Compute MI
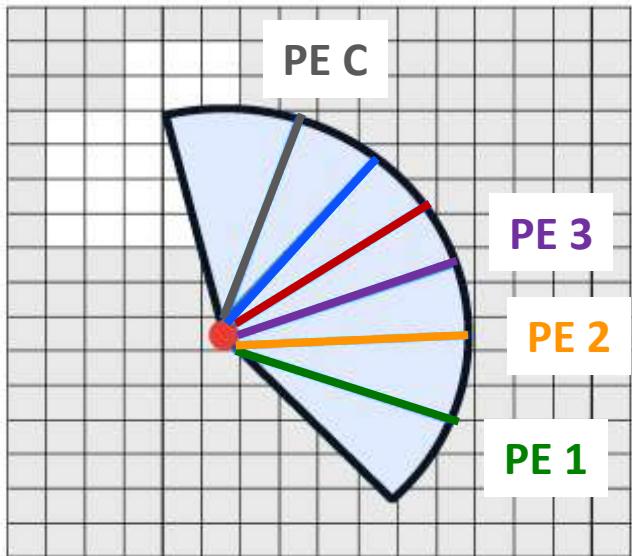
Motivation: Compute MI faster for faster exploration!

$$I(M;Z) = \sum_{j=1}^{n} \sum_{k=j-\Delta}^{j+\Delta} P(e_j) C_k G_{k,j}$$
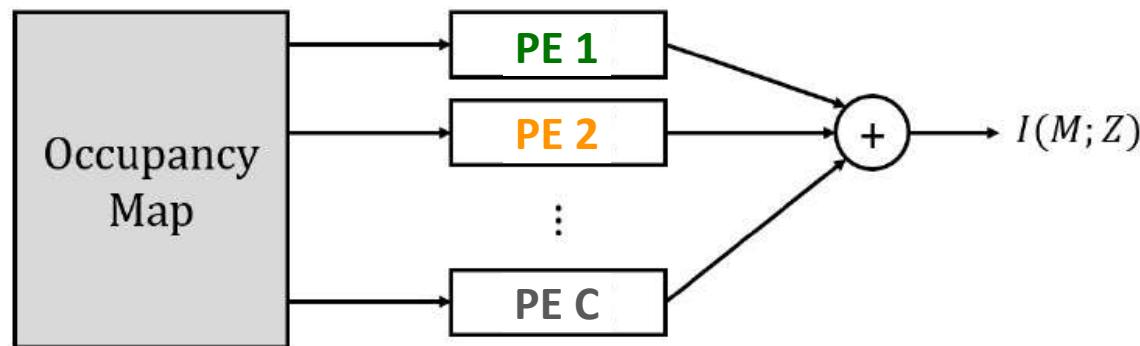
Fast Shannon
Mutual Information (FSMI)
[**Zhang**, *ICRA* 2019]

Algorithm is **embarrassingly** parallel!
High throughput **should** be possible with multiple processing elements (PE)
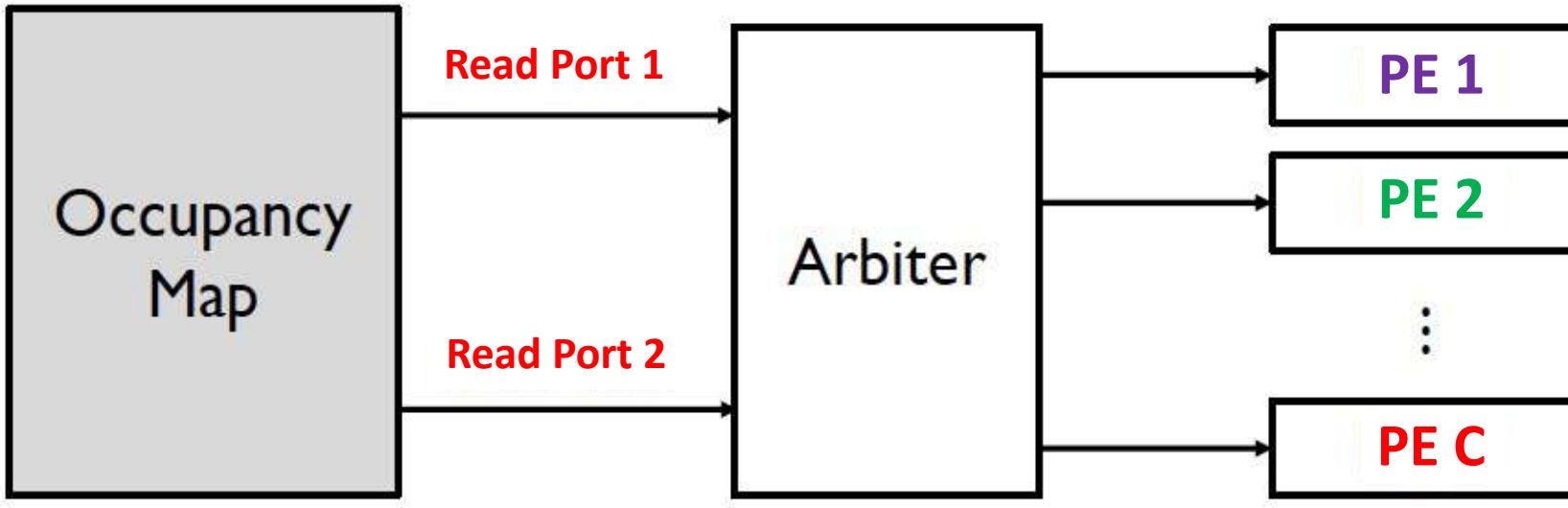
Process sensor beams in parallel with multiple PEs

Challenge is Data Delivery to All PEs
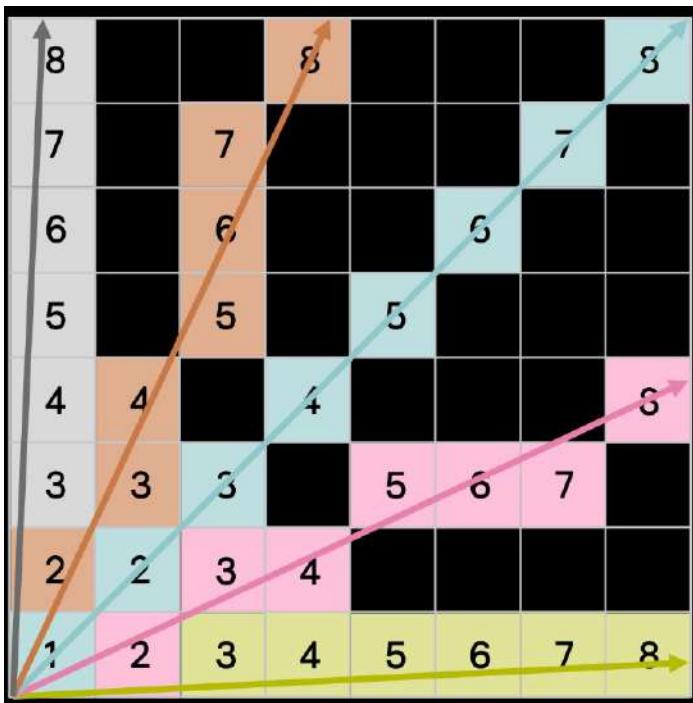
Power consumption of memory scales with number of ports.
Low power SRAM limited to two-ports!

Data delivery, specifically memory bandwidth,
limits the throughput (not compute)
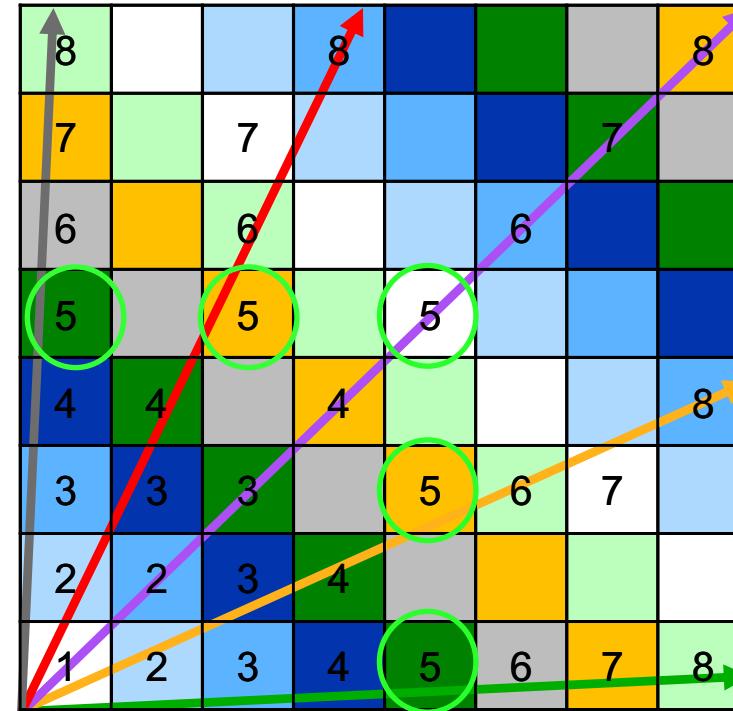
Optimized Memory Banking Pattern

Memory Access Pattern

PEs read the map at the **same row or column every cycle**
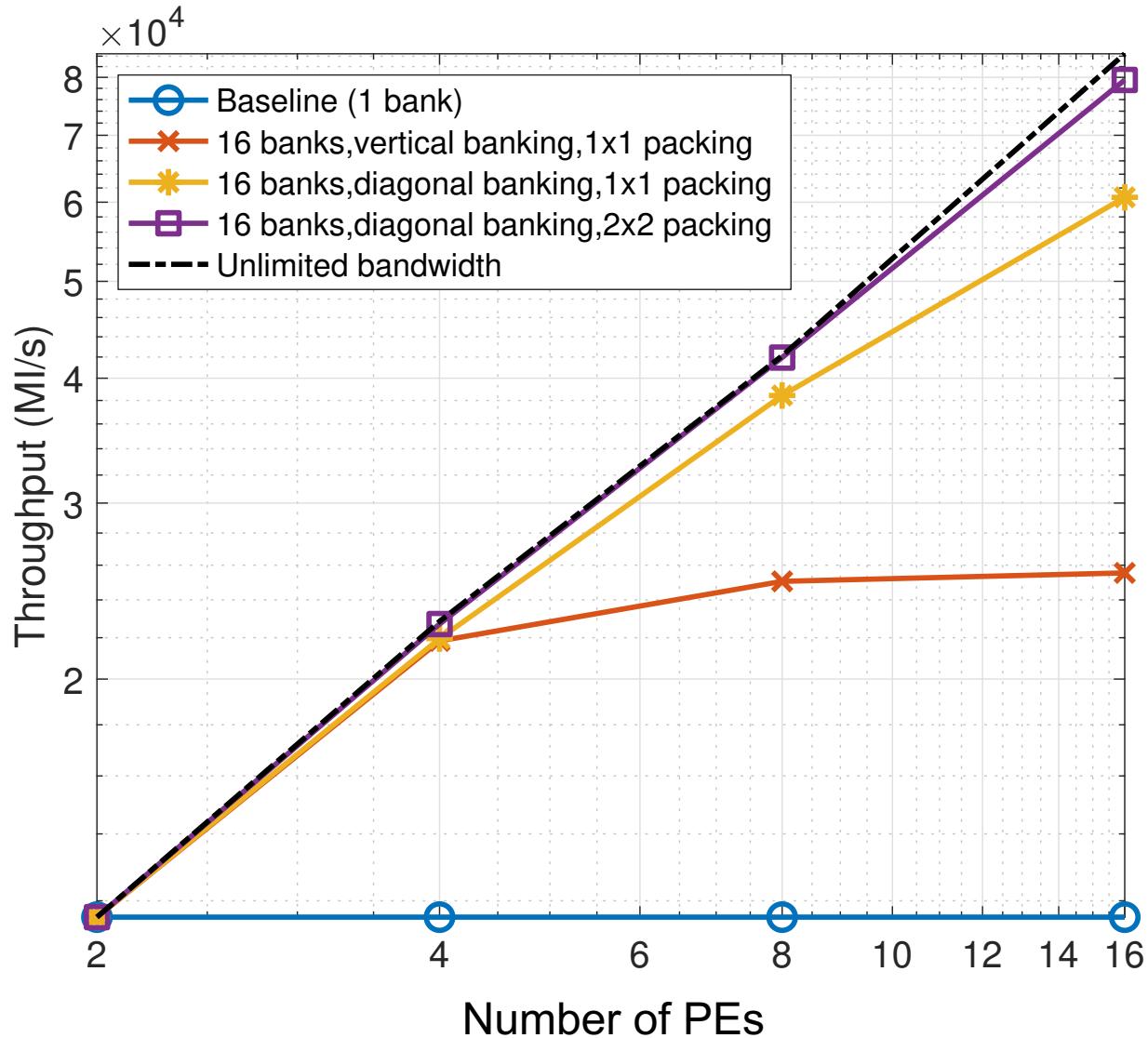
Diagonal Banking Pattern

☐ Bank 0
☐ Bank 1
☐ Bank 2
☐ Bank 3
☐ Bank 4
☐ Bank 5
☐ Bank 6
☐ Bank 7

Reduced conflicts across banks

Experimental Results
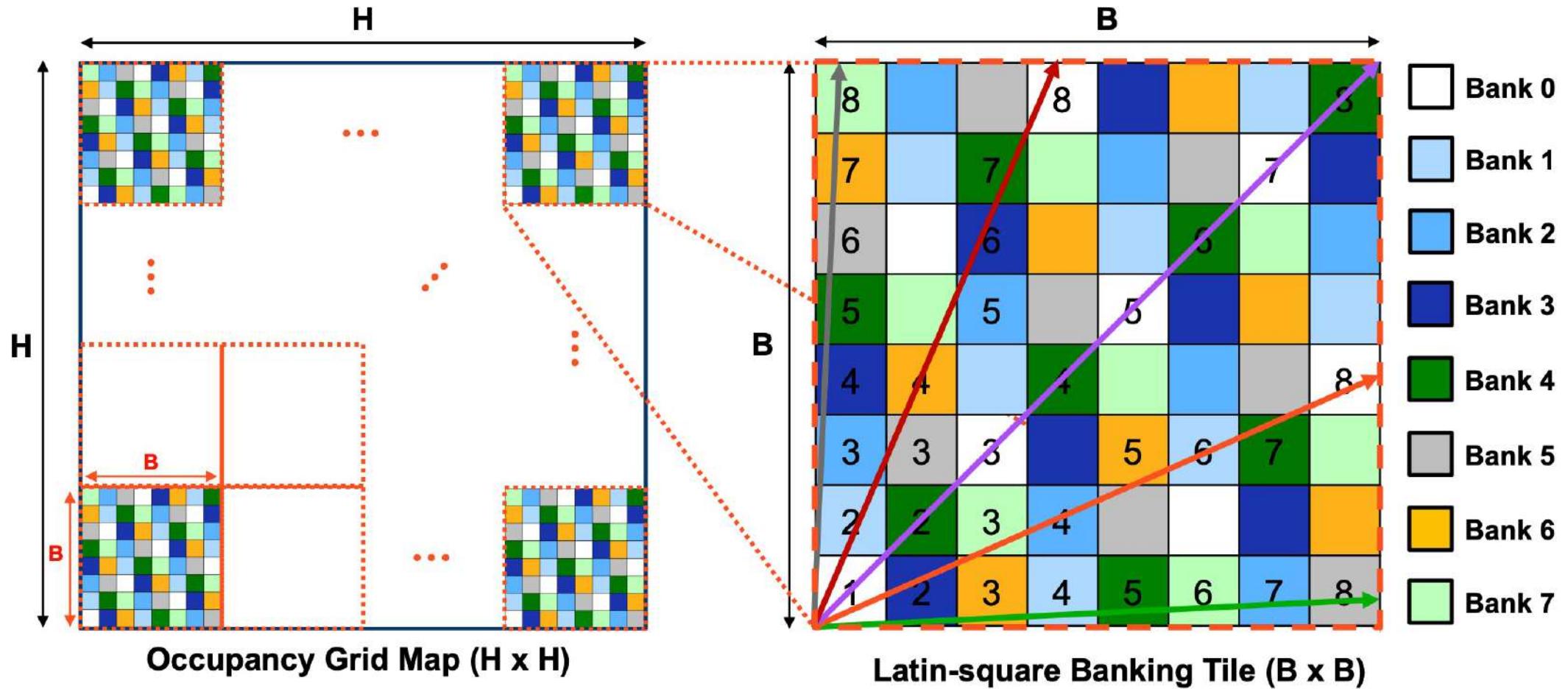
Specialized banking, efficient memory arbiter and packing multiple values at each address results in throughput **within 94% of theoretical limit** (unlimited bandwidth)

Compute MI for an **entire map** of 20m x 20m at 0.1m resolution **in under a second** on a ZC706 FPGA
(100x faster than CPU at 10x lower power)

Generalize to a Class of Banking Patterns
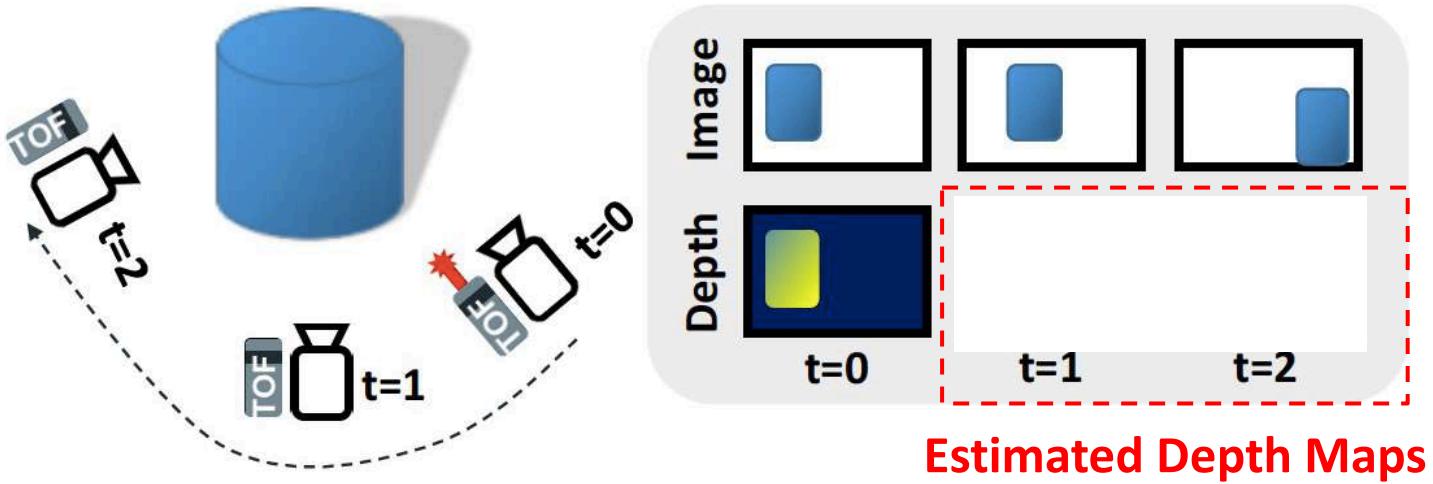
- **Latin-square banking tile:** cells in each column and row is assigned to different banks

Occupancy Grid Map (H x H)

Latin-square Banking Tile (B x B)

We **rigorously proved** that Latin-square tiles usage minimizes read conflicts between PEs
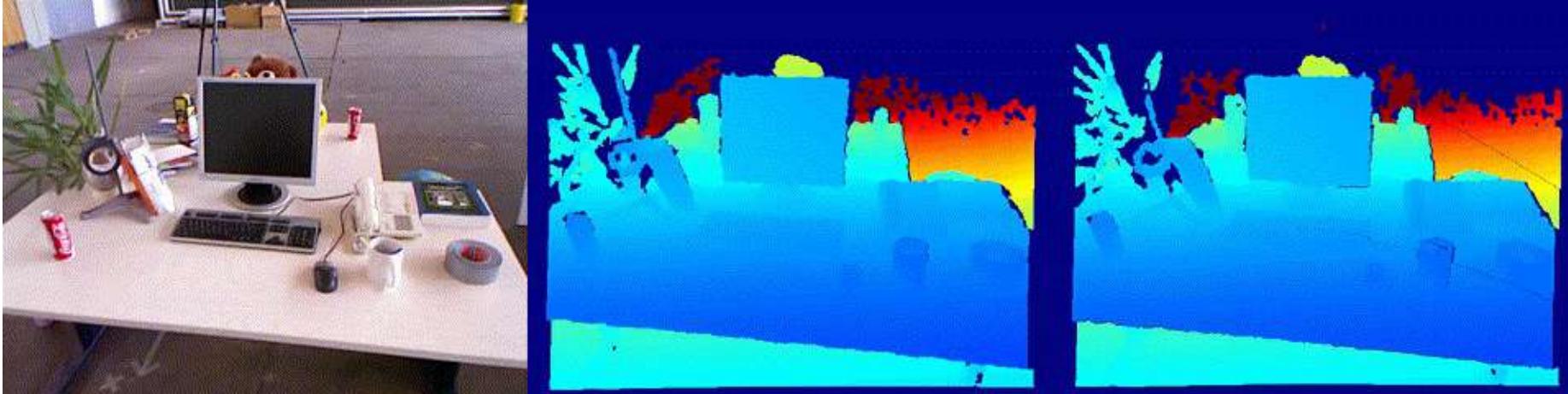
Low Power 3D Time of Flight Imaging

- Pulsed Time of Flight: Measure distance using round trip time of laser light for each image pixel

  – **Illumination + Imager Power: 2.5 – 20 W for range from 1 - 8 m**

- Use computer vision techniques and passive images to estimate changes in depth without turning on laser

  – **CMOS Imaging Sensor Power: < 350 mW**

Estimated Depth Maps

Real-time Performance on Embedded Processor
VGA @ 30 fps on Cortex-A7 (< 0.5W active power)

Results of Low Power Depth ToF Imaging

RGB Image

Depth Map
Ground Truth
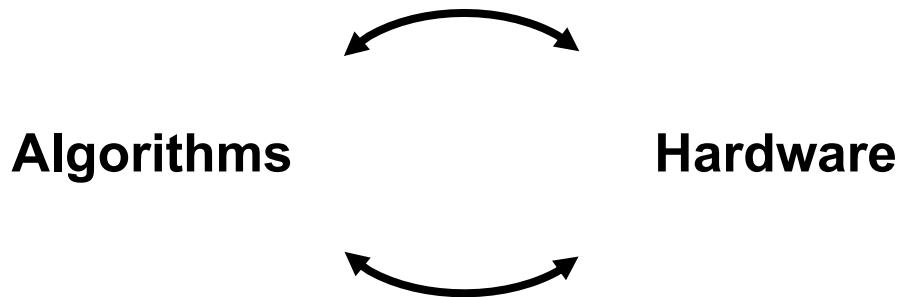
Depth Map
Estimated

Mean Relative Error: 0.7%
Duty Cycle (on-time of laser): 11%

Summary

- Efficient computing is critical for advancing the progress of AI & autonomous robots → **Critical step to making AI & autonomy ubiquitous!**

- In order to meet computing demands in terms of power and speed, need to redesign computing hardware from the ground up → **Focus on data movement!**

- Specialized hardware creates new opportunities for the co-design of algorithms and hardware → **Innovation opportunities for the future of AI & robotics!**

Algorithms   **Hardware**

Acknowledgements

Joel Emer

Sertac Karaman

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

MIT

Low-Energy Autonomy and Navigation (LEAN) Group

LEAN | HOME | TEAM | **RESEARCH** | PUBLICATIONS | PRESS | RECOGNITION
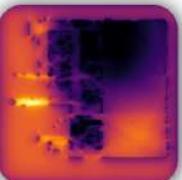
A broad range of next-generation applications will be enabled by low-energy, miniature mobile robotics including insect-size flapping wing robots that can help with search and rescue, chip-size satellites that can explore nearby stars, and blimps that can stay in the air for years to provide communication services in remote locations. While the low-energy, miniature actuation, and sensing systems have already been developed in many of these cases, the processors currently used to run the algorithms for autonomous navigation are still energy-hungry. Our research addresses this challenge as well as brings together the robotics and hardware design communities.

We enable efficient computing on various key modules of other autonomous navigation systems including perception, localization, exploration and planning. We also consider the overall system by considering the energy cost of computing in conjunction with actuation and sensing.

Motion Planning

Many motion planning and control algorithms aim to design trajectories and controllers that minimize actuation energy. However, in low-energy robotics, computing such trajectories and controls themselves may consume a large amount of energy. We develop algorithms that optimize this trade-off.

Mutual Information for Exploration

Computing mutual information between the map and future measurements is critical to efficient exploration. Unfortunately, mutual information computation is computationally very challenging. We develop new algorithms and hardware for efficient computation of mutual information, and demonstrate real-time computation for the whole map in a reasonably-sized map.

Depth Sensing and Perception

Depth sensing is a critical function for robotic tasks such as localization, mapping and obstacle detection. State-of-the-art single-view depth estimation algorithms are based on fairly complex deep neural networks that are too slow for real-time inference on an embedded platform, for instance, mounted on a micro aerial vehicle. We address the problem of fast depth estimation on embedded systems.
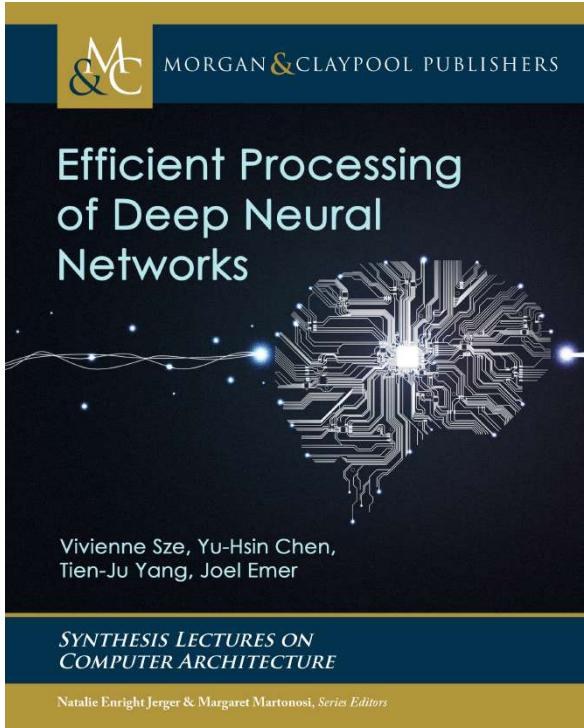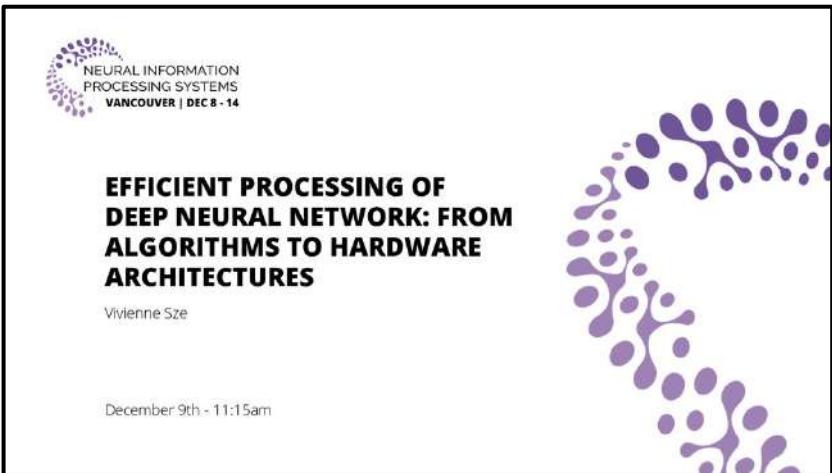
Localization and Mapping

Autonomous navigation of miniaturized robots (e.g., nano/pico aerial vehicles) is currently a grand challenge for robotics research, due to the need for processing a large amount of sensor data (e.g., camera frames) with limited on-board computational resources. We focus on the design of a visual-inertial odometry (VIO) system in which the robot estimates its ego-motion (and a landmark-based map) from on-board camera and IMU data.

Group Website: http://lean.mit.edu

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit

Resources on Efficient Processing of DNNs

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

Excerpts of Book

Available on DNN tutorial website
http://eyeriss.mit.edu/tutorial.html

Additional Resources

Talks and Tutorial Available Online
https://tinyurl.com/ISCAS2021Sze

YouTube Channel
EEMS Group – PI: Vivienne Sze

References

- **Efficient Processing for Deep Neural Networks**

  - **Project website:** http://eyeriss.mit.edu

  - Y. N. Wu, P.-A. Tsai, A. Parashar, V. Sze, J. S. Emer, "Sparseloop: An Analytical, Energy-Focused Design Space Exploration Methodology for Sparse Tensor Accelerators," IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), March 2021

  - 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), Vol. 9, No. 2, pp. 292-308, June 2019.

  - 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, "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.

  - Y. N. Wu, J. S. Emer, V. Sze, "Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs," International Conference on Computer Aided Design (ICCAD), November 2019. http://accelergy.mit.edu/

  - Y. N. Wu, V. Sze, J. S. Emer, "An Architecture-Level Energy and Area Estimator for Processing-In-Memory Accelerator Designs," to appear in IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), April 2020.

  - A. Suleiman*, Y.-H. Chen*, J. Emer, V. Sze, "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," IEEE International Symposium of Circuits and Systems (ISCAS), Invited Paper, May 2017.

  - Hardware Architecture for Deep Neural Networks: http://eyeriss.mit.edu/tutorial.html

References

- **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, V. Sze, "Design Considerations for Efficient Deep Neural Networks on Processing-in-Memory Accelerators," IEEE International Electron Devices Meeting (IEDM), Invited Paper, December 2019.
  - T.-J. Yang, Y.-L. Liao, V. Sze, "NetAdaptV2: Efficient Neural Architecture Search with Fast Super-Network Training and Architecture Optimization ," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

- **Low Power Time of Flight Imaging**
  - J. Noraky, V. Sze, "Low Power Depth Estimation of Rigid Objects for Time-of-Flight Imaging," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2020.
  - J. Noraky, V. Sze, "Depth Map Estimation of Dynamic Scenes Using Prior Depth Information," arXiv, February 2020. https://arxiv.org/abs/2002.00297
  - J. Noraky, V. Sze, "Depth Estimation of Non-Rigid Objects For Time-Of-Flight Imaging," IEEE International Conference on Image Processing (ICIP), October 2018.

References

- **Energy-Efficient Visual Inertial Localization**

  - **Project website:** http://navion.mit.edu

  - A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Symposium on VLSI Circuits (VLSI-Circuits), June 2018.

  - Z. Zhang*, A. Suleiman*, L. Carlone, V. Sze, S. Karaman, "Visual-Inertial Odometry on Chip: An Algorithm-and-Hardware Co-design Approach," Robotics: Science and Systems (RSS), July 2017.

  - A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A 2mW Fully Integrated Real-Time Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Journal of Solid State Circuits (JSSC), VLSI Symposia Special Issue, Vol. 54, No. 4, pp. 1106-1119, April 2019.

References

- **Fast Shannon Mutual Information for Robot Exploration**
  - **Project website:** http://lean.mit.edu
  - Z. Zhang, T. Henderson, V. Sze, S. Karaman, "FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping," IEEE International Conference on Robotics and Automation (ICRA), May 2019.
  - P. Li*, Z. Zhang*, S. Karaman, V. Sze, "High-throughput Computation of Shannon Mutual Information on Chip," Robotics: Science and Systems (RSS), June 2019
  - Z. Zhang, T. Henderson, S. Karaman, V. Sze, "FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping," to appear in International Journal of Robotics Research (IJRR). http://arxiv.org/abs/1905.02238
  - T. Henderson, V. Sze, S. Karaman, "An Efficient and Continuous Approach to Information-Theoretic Exploration," IEEE International Conference on Robotics and Automation (ICRA), May 2020

- **Balancing Actuation and Computation**
  - **Project website:** http://lean.mit.edu
  - S. Sudhakar, S. Karaman, V. Sze, "Balancing Actuation and Computing Energy in Motion Planning," IEEE International Conference on Robotics and Automation (ICRA), May 2020

Vivienne Sze 🌐 http://sze.mit.edu/ 🐦 @eems_mit