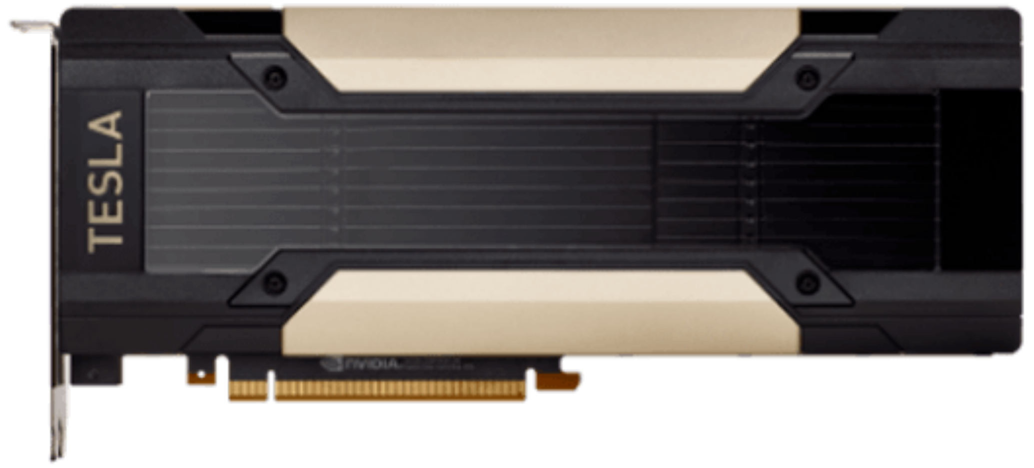
Putting AI on a Diet:

TinyML and Efficient Deep Learning

Song Han
Assistant Professor
Massachusetts Institute of Technology

https://songhan.mit.edu

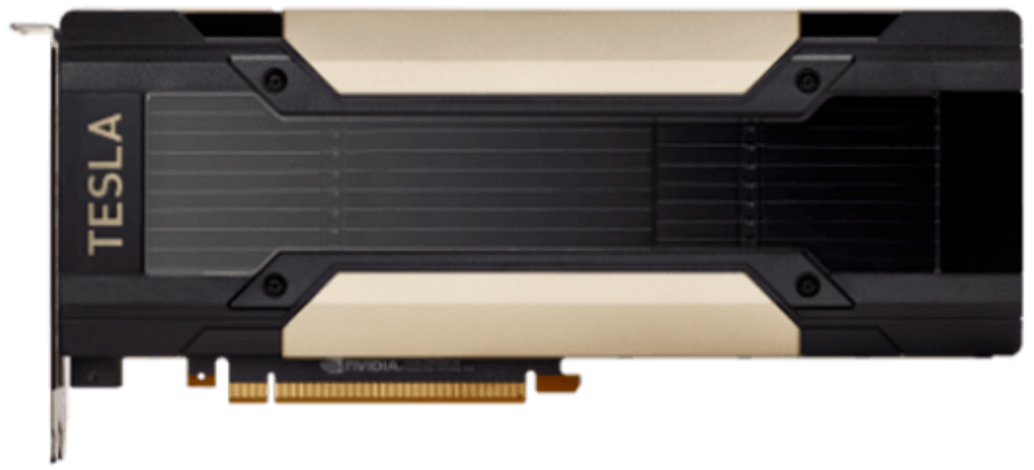
From Cloud to Mobile to Tiny AI



Cloud AI

Data centers
Expensive
Connection required
Privacy issue

From Cloud to Mobile to Tiny AI



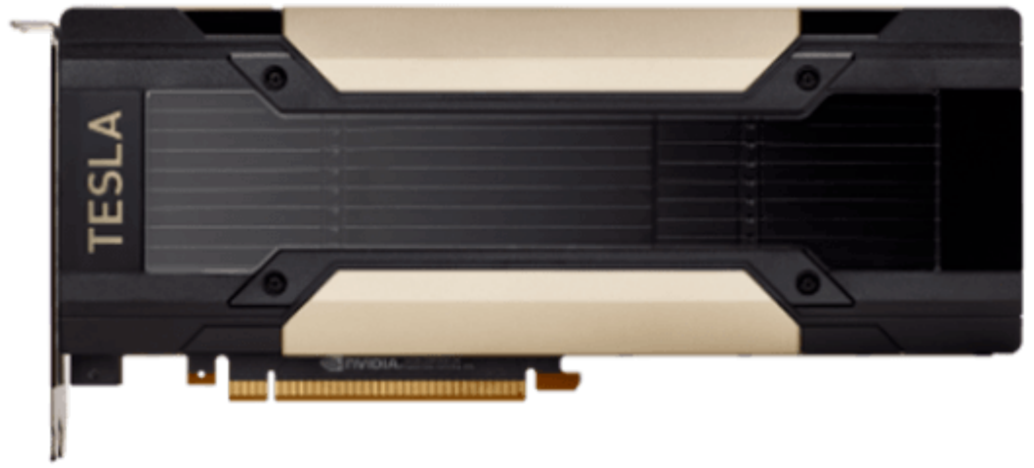
Cloud AI

Data centers
Expensive
Connection required
Privacy issue

Mobile AI

Smartphones
Low cost
Accessible
Process locally

From Cloud to Mobile to Tiny AI



?

Cloud AI

Data centers
Expensive
Connection required
Privacy issue

Mobile AI

Smartphones
Low-cost
Accessible
Process locally

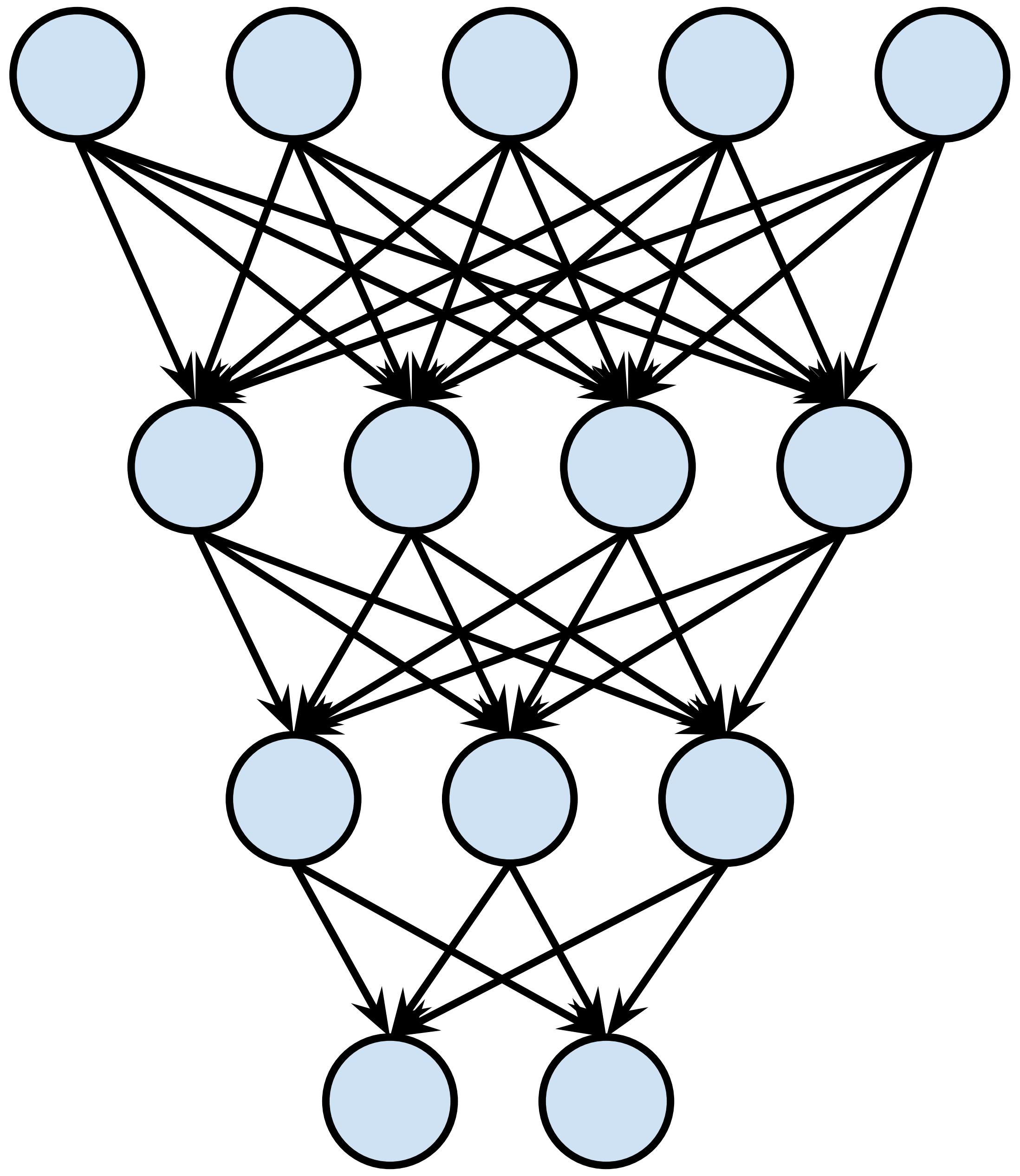
Deep Compression

Make AI run Fast and Efficiently
with Limited Hardware Resource

Han, Mao, Dally, Deep Compression, ICLR'16, best paper award

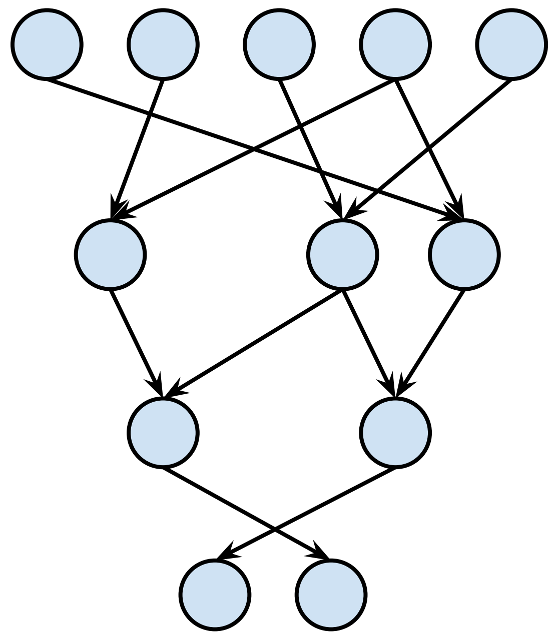
Deep Compression

Make AI run Fast and Efficiently
with Limited Hardware Resource



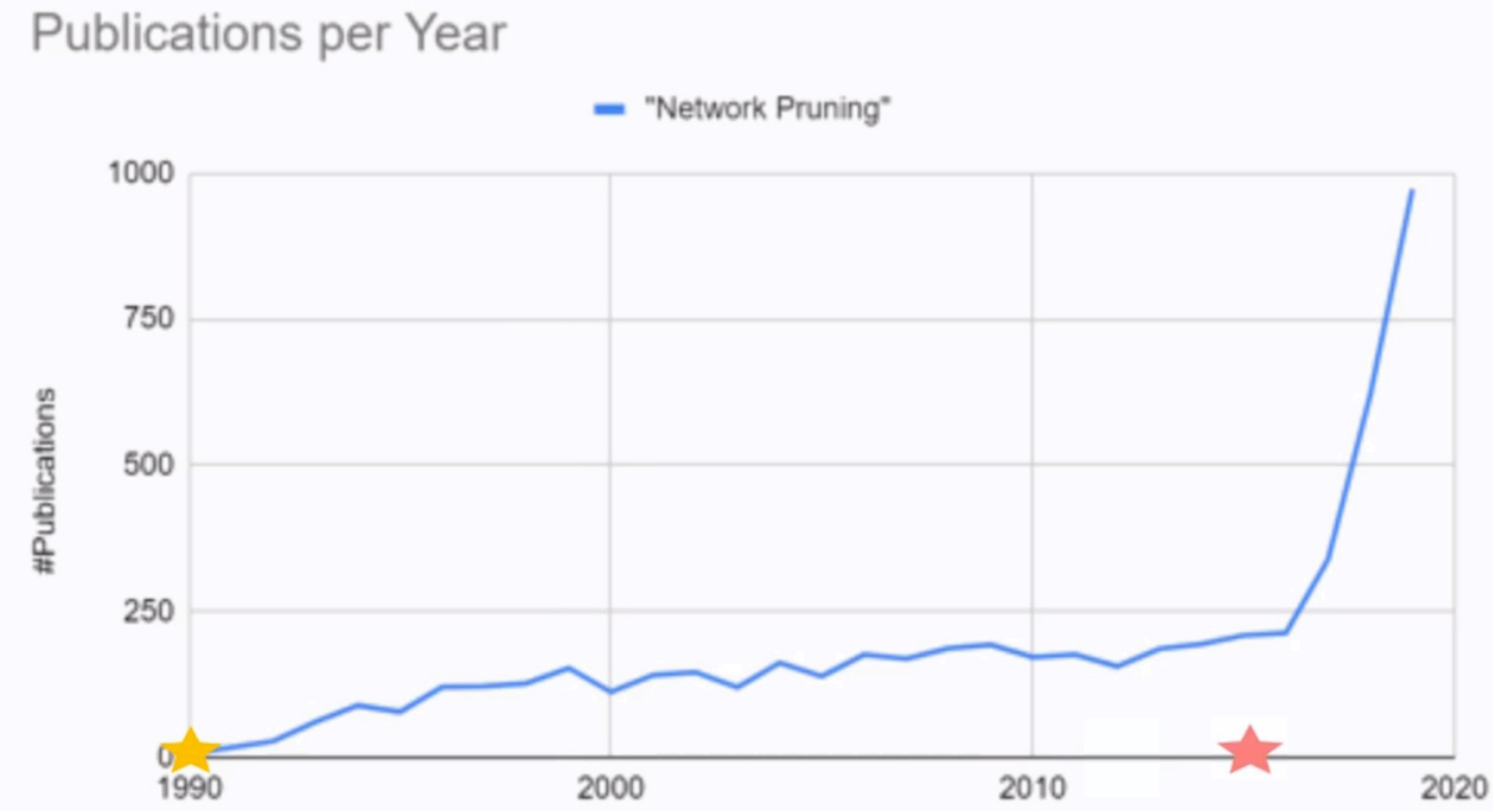
Pruning

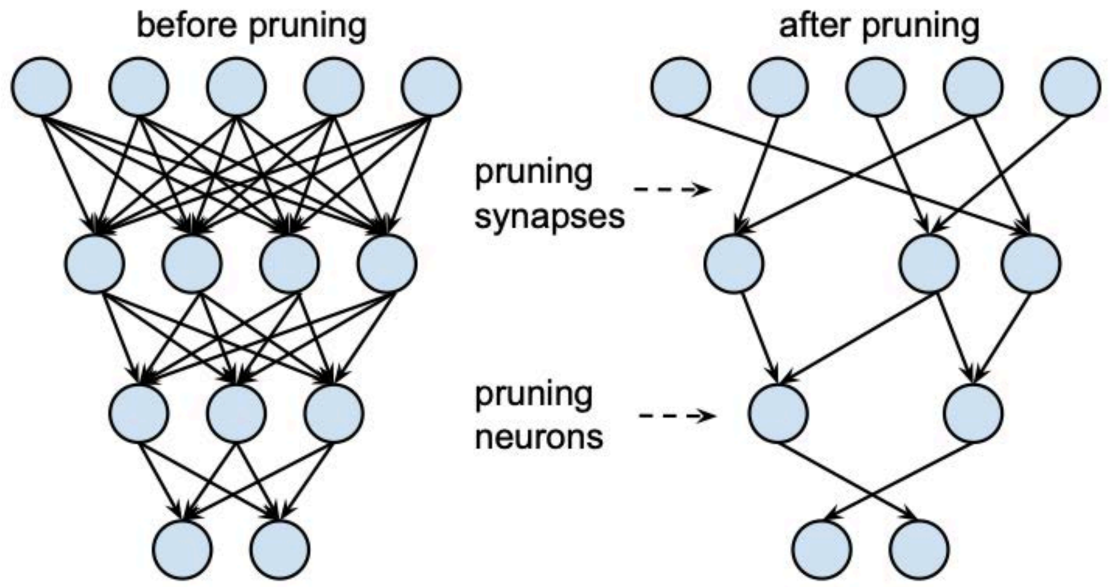
Deep Compression



Pruning & Sparsity

Increased attention since 2015





Han et al., NIPS'15

Optimal Brain Damage

Yann Le Cun, John S. Denker and Sara A. Solla
AT&T Bell Laboratories, Holmdel, N. J. 07733

Learning both Weights and Connections for Efficient Neural Networks

Song Han
Stanford University
songhan@stanford.edu

Jeff Pool
NVIDIA
jpool@nvidia.com

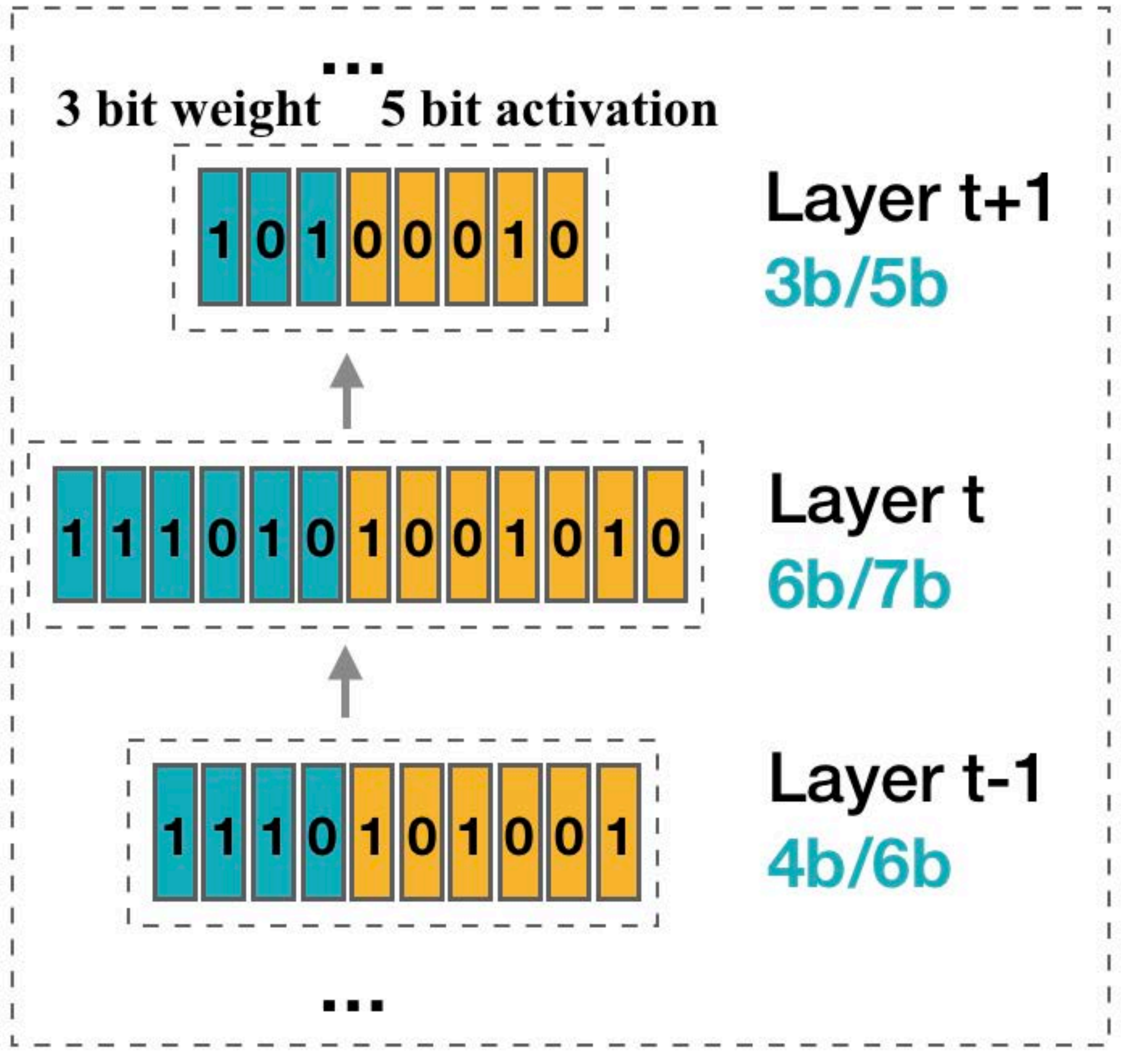
John Tran
NVIDIA
johntran@nvidia.com

William J. Dally
Stanford University
NVIDIA
dally@stanford.edu

curve credits to NVIDIA

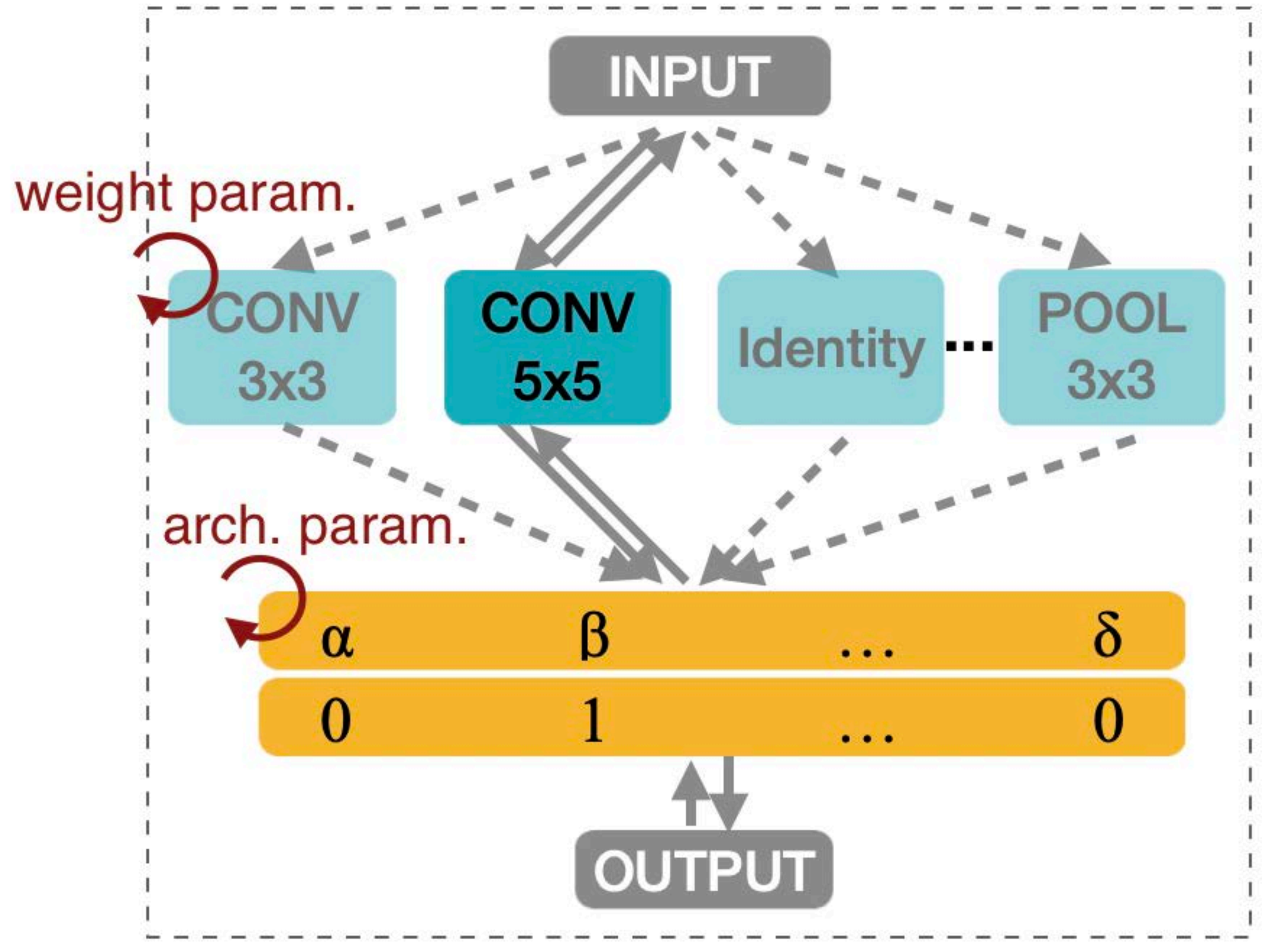
AutoML and Neural Architecture Search

auto design small models



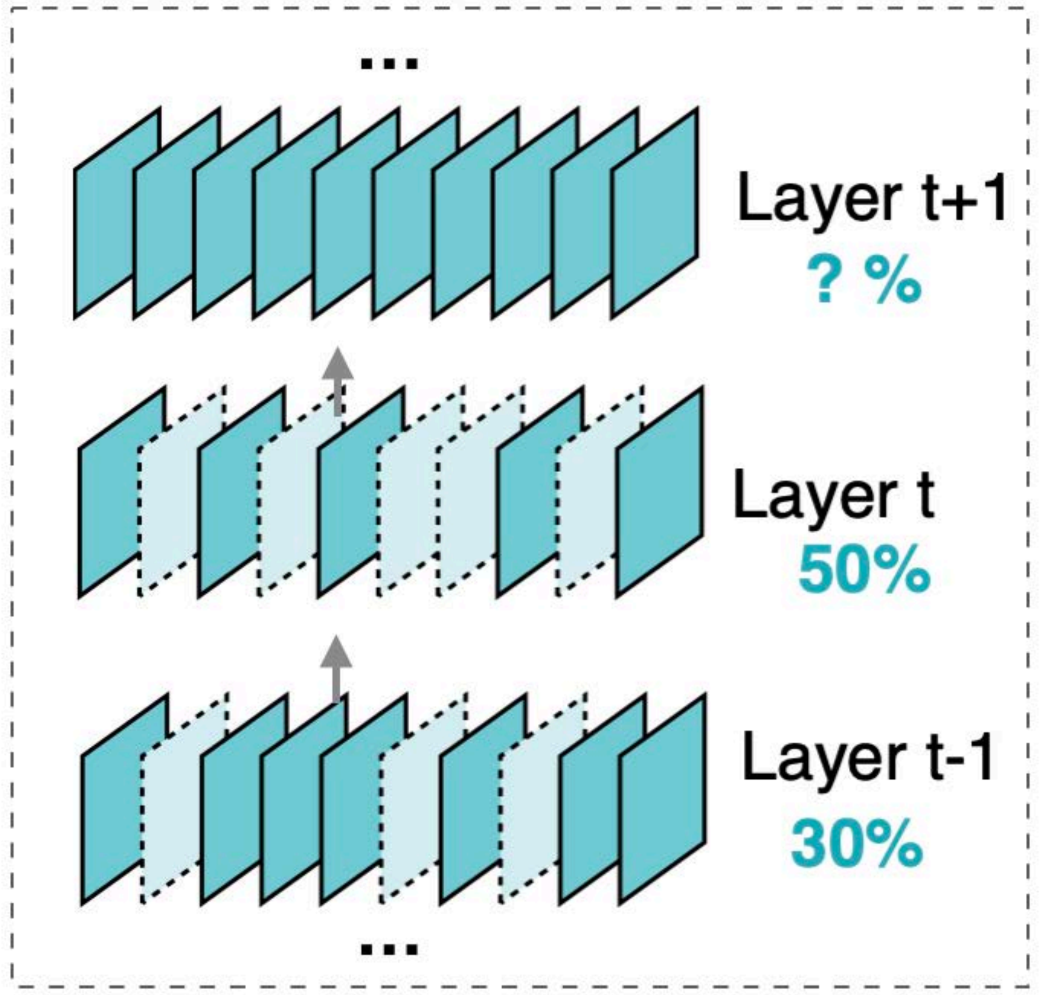
HAQ: Hardware-aware Automated Quantization

[CVPR 2019], oral



Proxyless Neural Architecture Search

[ICLR 2019]



AMC: AutoML for Model Compression

[ECCV 2018]

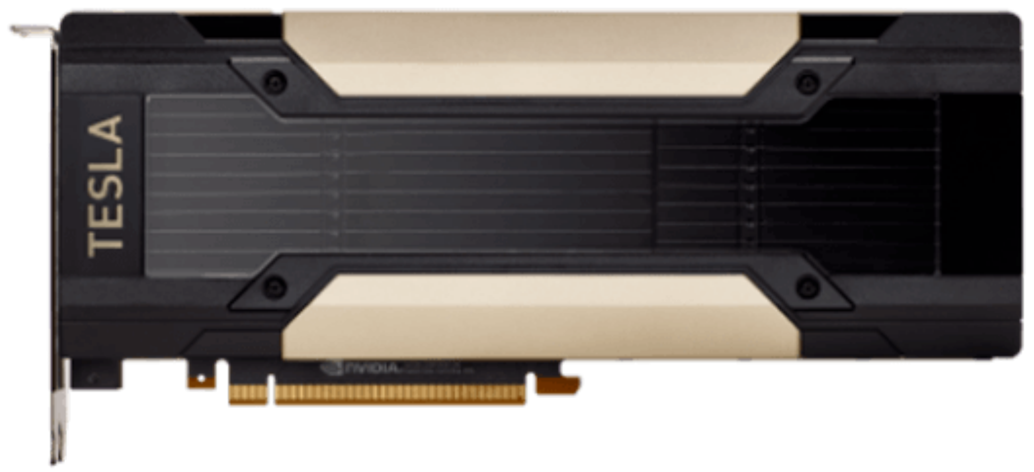
1st Place of Visual Wakeup Words (VWW) Challenge 2019

Peak Memory Usage < 250KB, ours: 245KB

Model Size < 250KB, ours: 242KB

MAC < 60M, ours: 50M, Accuracy: 94.6%

Deep Learning Going "Tiny"



Cloud AI

Data centers
Expensive
Connection required
Privacy issue

Mobile AI

Smartphones
Accessible
Process locally

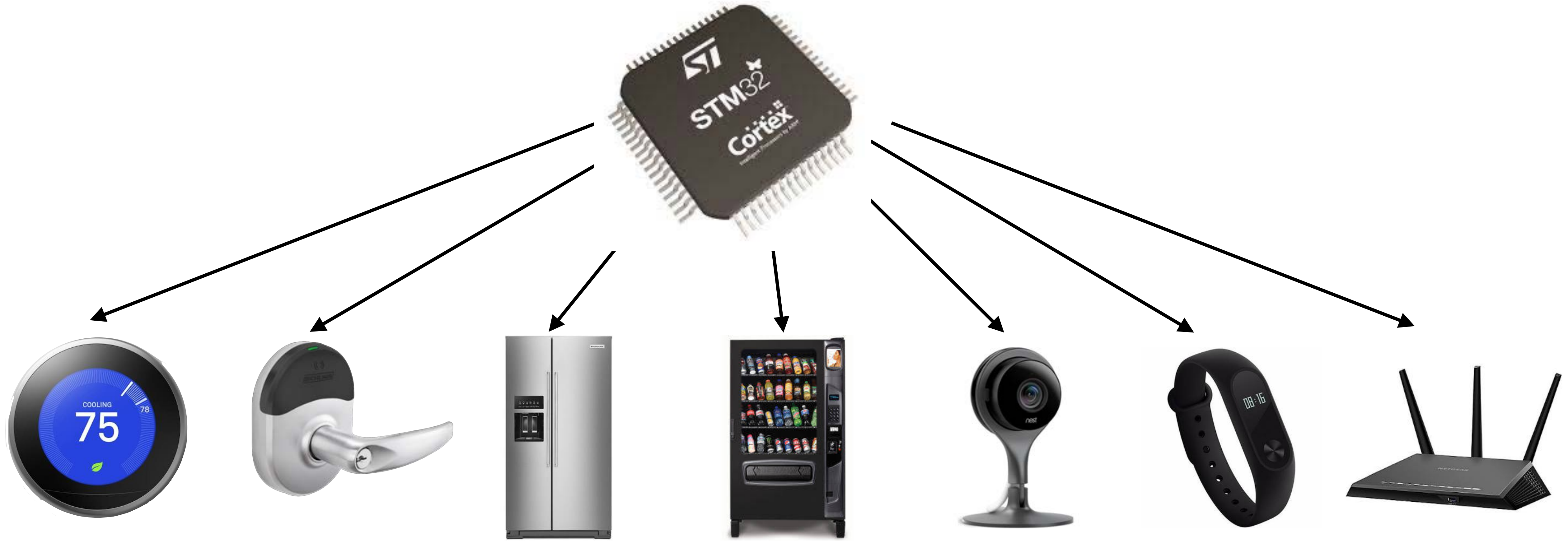
<u>Tiny AI</u>

IoT Devices/
Microcontrollers
Cheap, small, low-power
Rapid growth

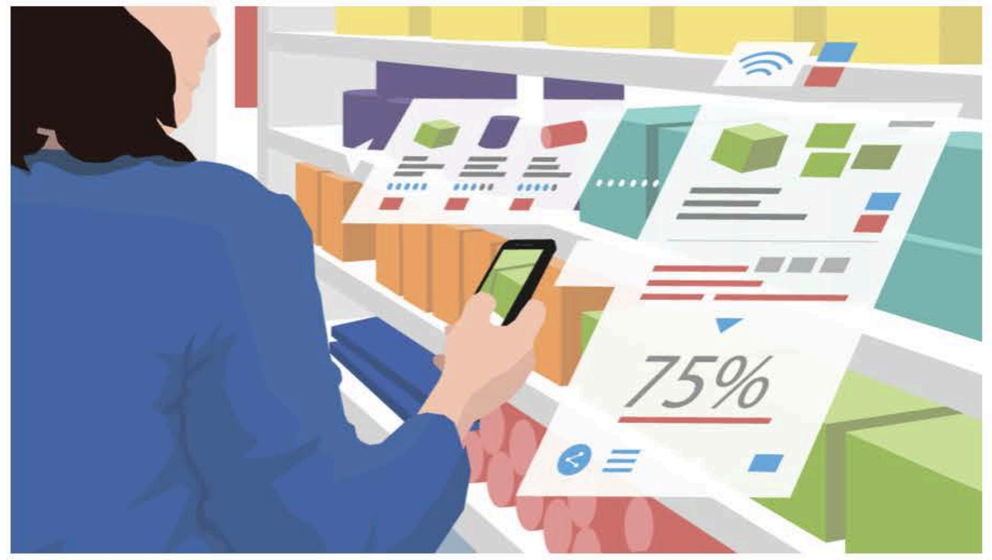
- The future belongs to Tiny AI.
- There are billions of IoT devices around the world based on microcontrollers
- Much cheaper, much smaller, almost everywhere in our lives.
- If we can enable powerful AI algorithms on those IoT devices, we can greatly democratize AI and extend the applications of deep learning.

The Era of AIoT on Microcontrollers (MCUs)

Microcontrollers



Smart Retail

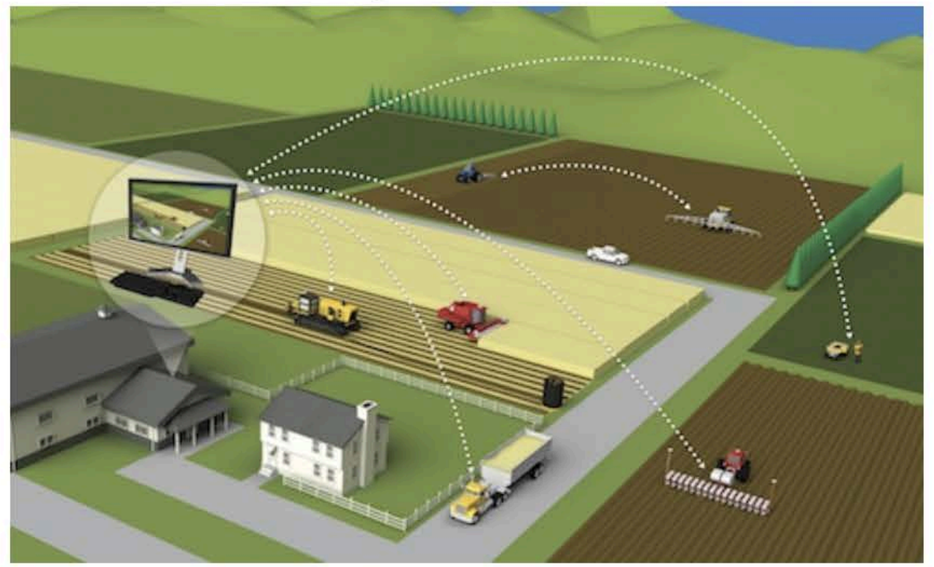


Personalized Healthcare

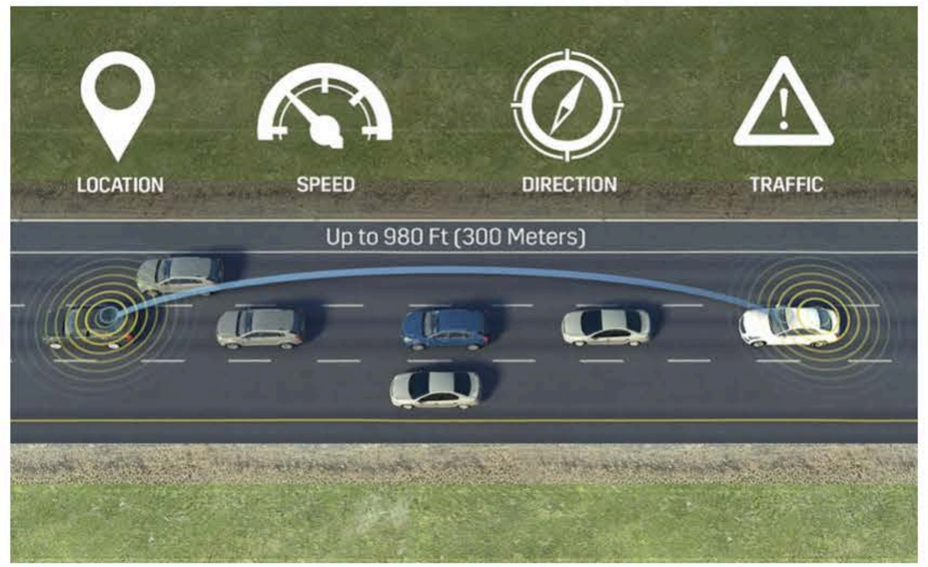
Smart Home

Smart Manufacturing

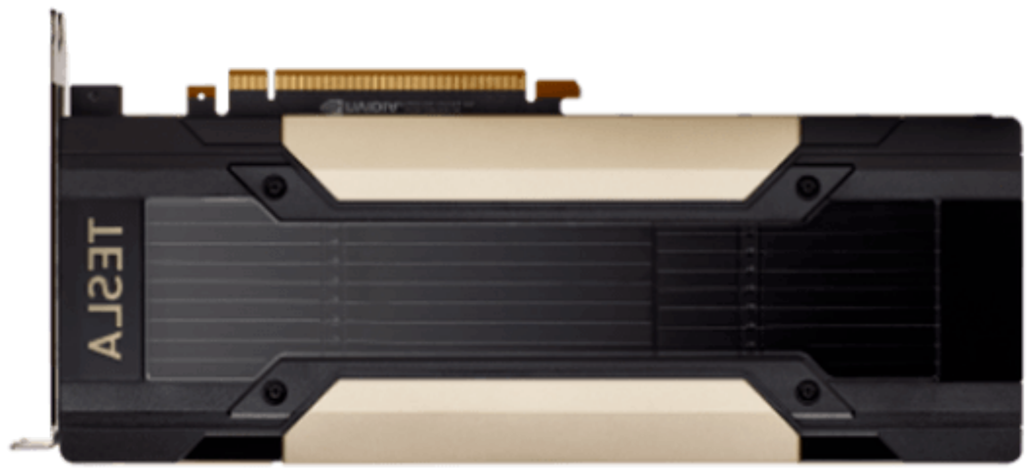
Precision Agriculture



Autonomous Driving



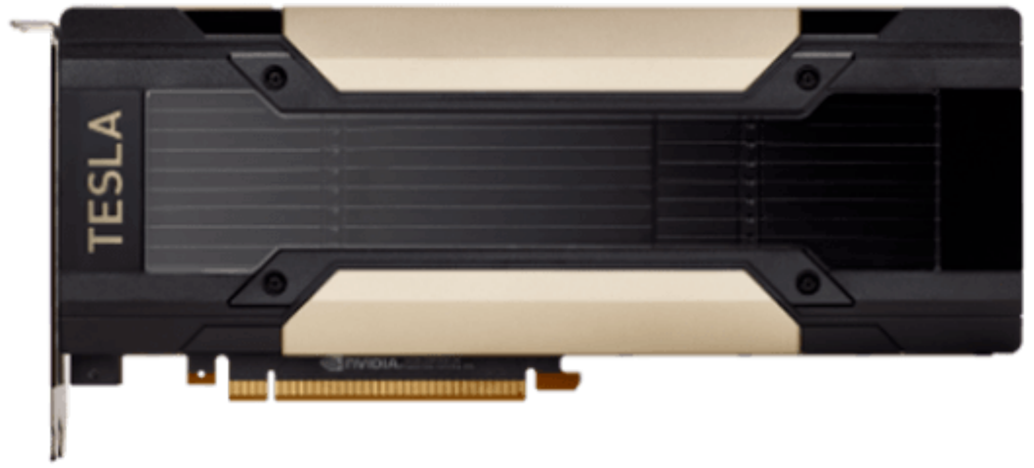
Challenge: Memory Too Small to Hold DNN



	Cloud AI	**Mobile AI**	**Tiny AI**
Memory (Activation)	16GB	4GB	320kB
Storage (Weights)	~TB/PB	256GB	1MB

- Tiny model design is fundamentally different.
- No DRAM. No operating system (no virtual memory).
- Can't directly scale. (non-proportional activation vs. params)

Challenge: Memory Too Small to Hold DNN



	Cloud AI	**Mobile AI**	**Tiny AI**
Memory (Activation)	16GB	4GB	320kB
Storage (Weights)	~TB/PB	256GB	1MB

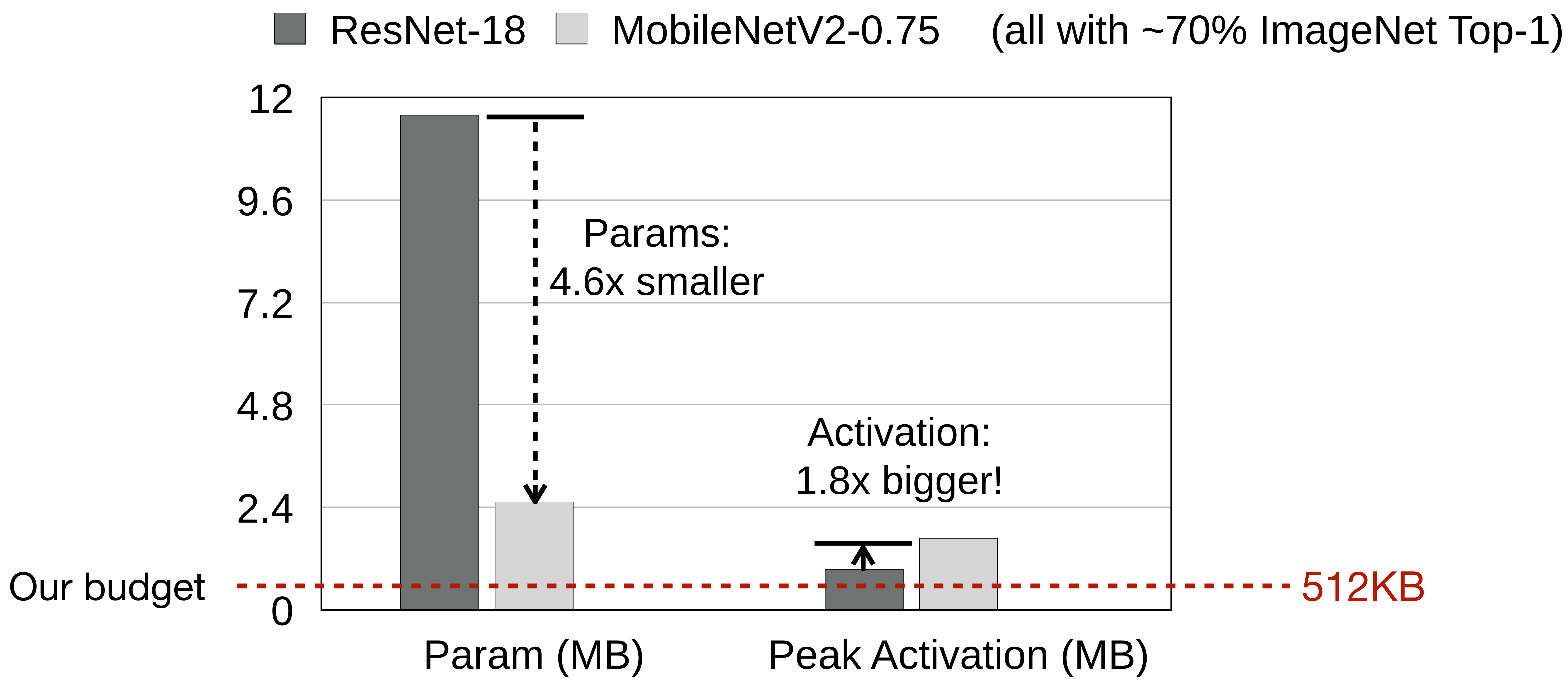
13,000x smaller

50,000x smaller

- Tiny model design is fundamentally different.
- No DRAM. No operating system (no virtual memory).
- Can't directly scale. (non-proportional activation vs. params)

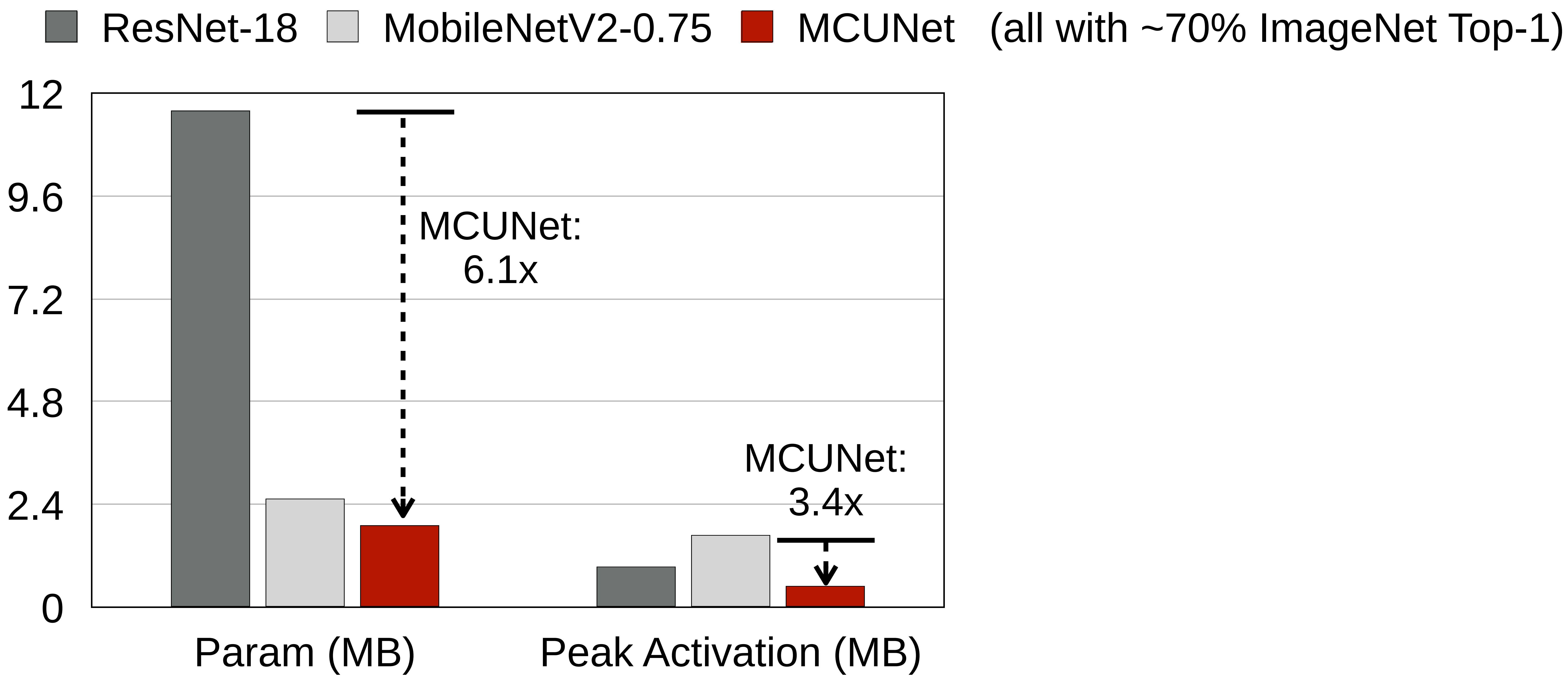
Today's AI is too big!

Existing work only reduces model size, but NOT activation



(calculated in INT8)

Reduce Both Model Size and Activation Size

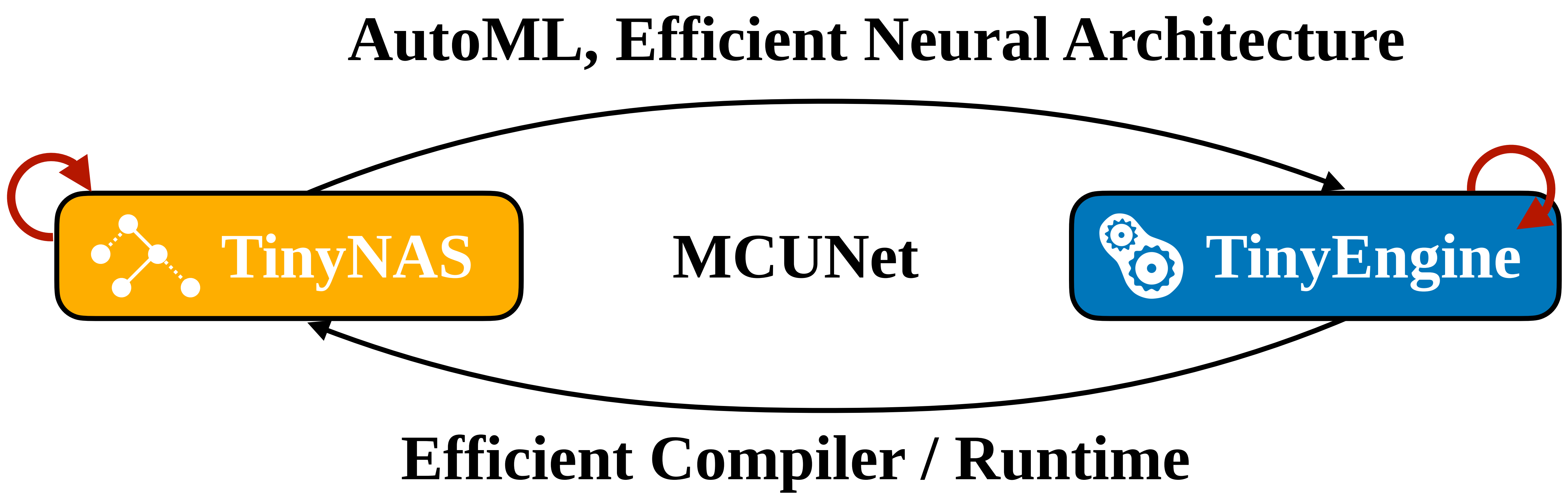


simple applications

MCUNet

ImageNet-1K classification

MCUNet: TinyNAS+TinyEngine Co-design

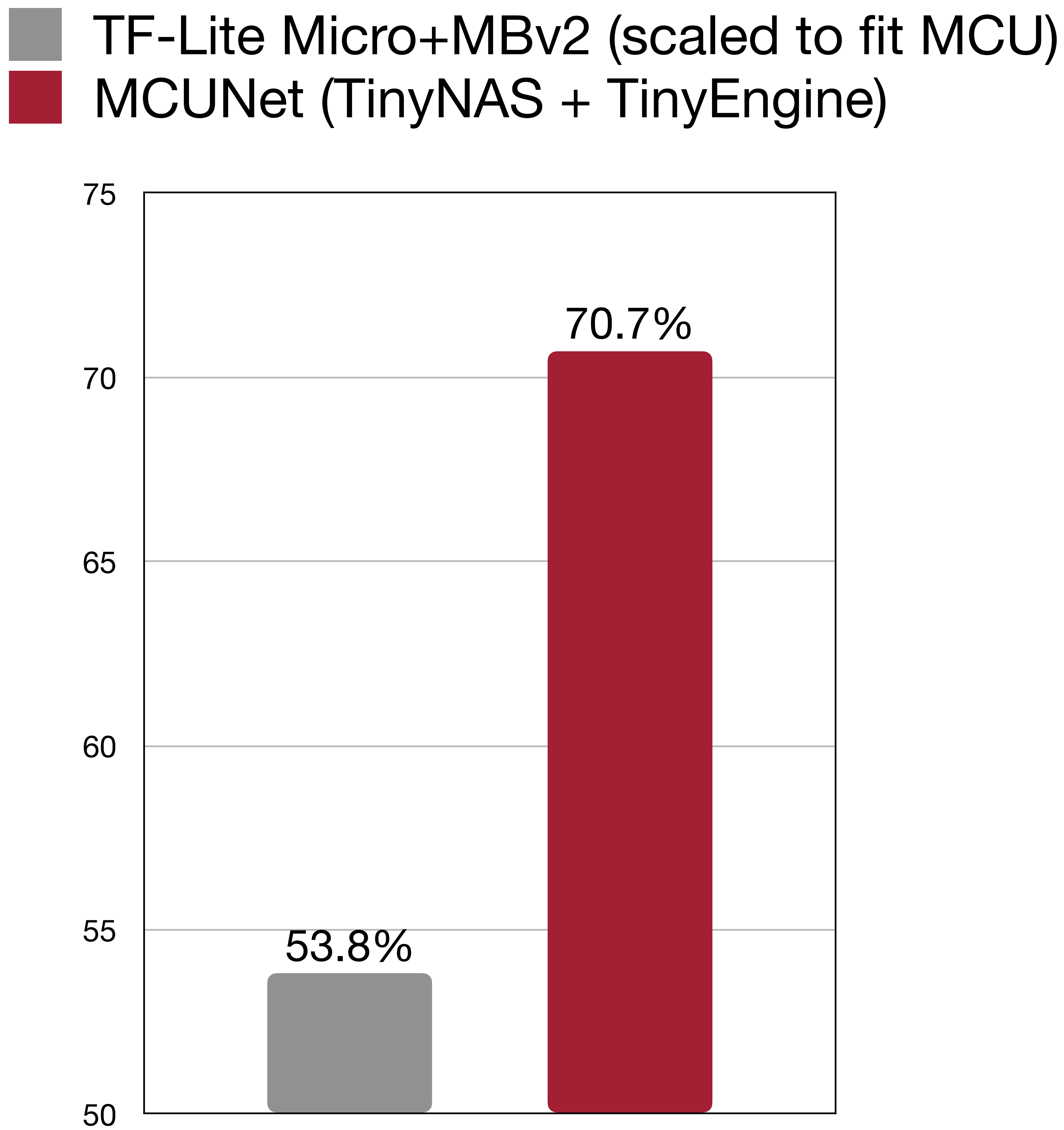


- TinyNAS:
  - Re-design the design space
  - Latency-aware
  - Energy-aware
  - Once-for-all Network: train once, get many

- TinyEngine:
  - Co-design, specialization
  - Graph optimizations
  - Memory-aware scheduling
  - Low-precision
  - Assembly-level optimizations

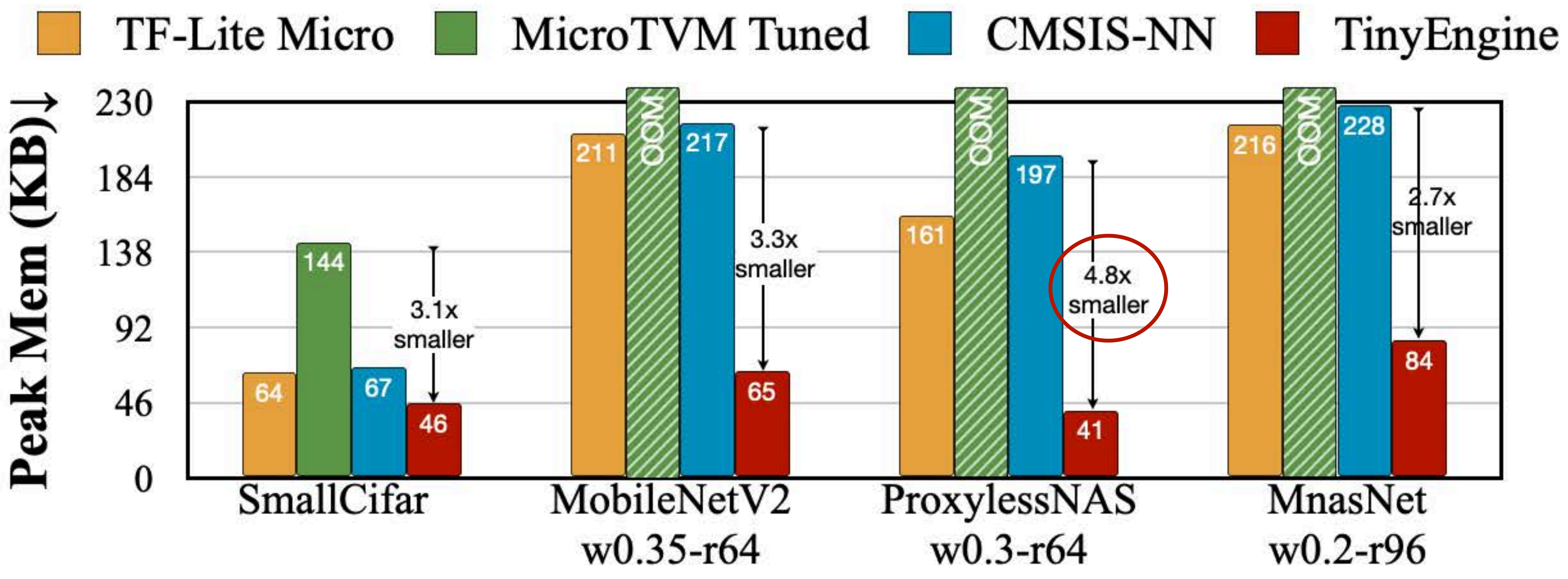
MCUNet: Bring AI to IoT Devices

MIT researchers have developed a system, called MCUNet, that brings machine learning to microcontrollers. The advance could enhance the function and security of devices connected to the Internet of Things (IoT). ——MIT News



ImageNet 1K, Top-1 Accuracy

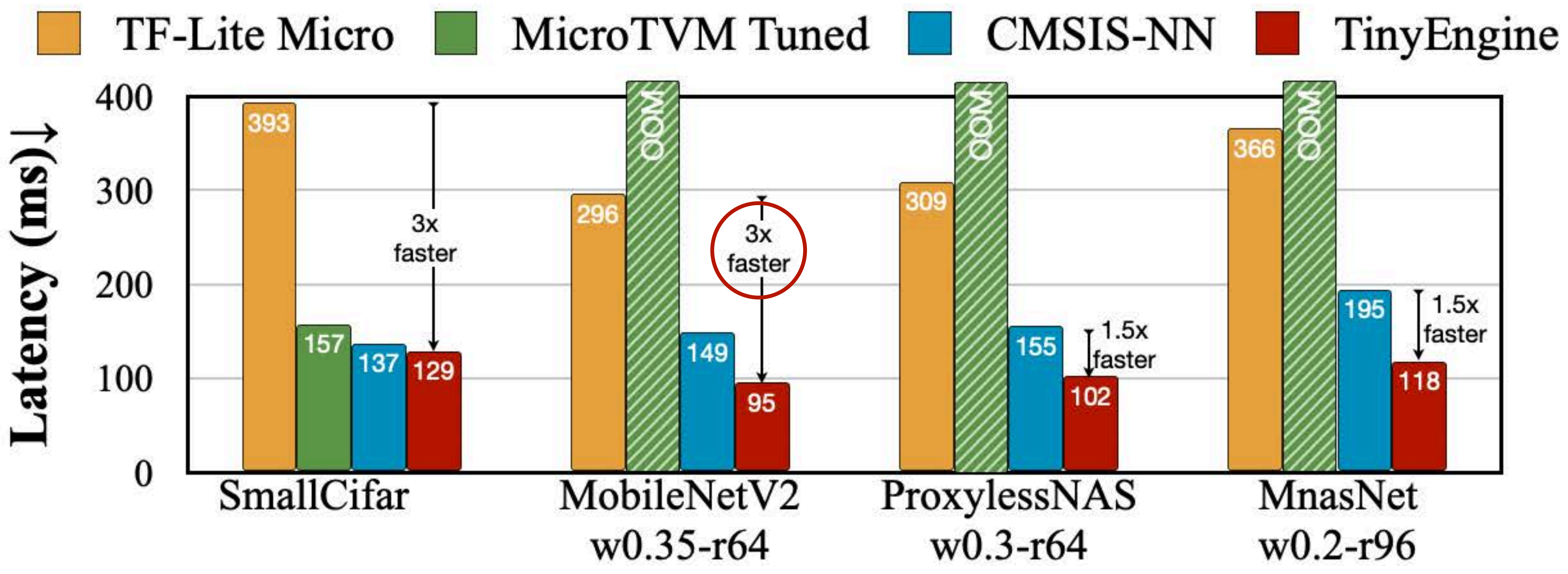
TinyEngine: Memory Saving



Measured on STM32 MCU

TinyEngine: Speedup

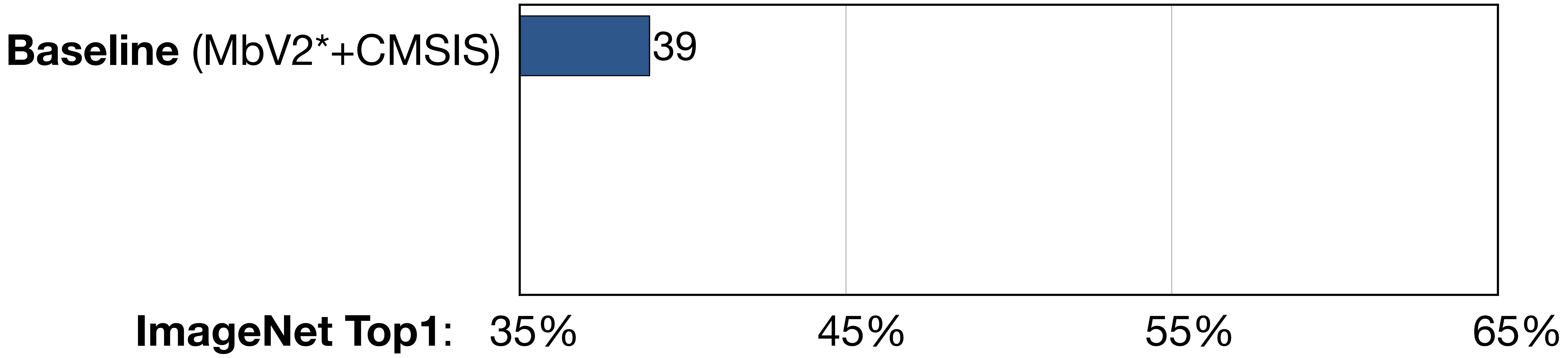
OctoML ARM



Measured on STM32 MCU

MCUNet: TinyNAS+TinyEngine

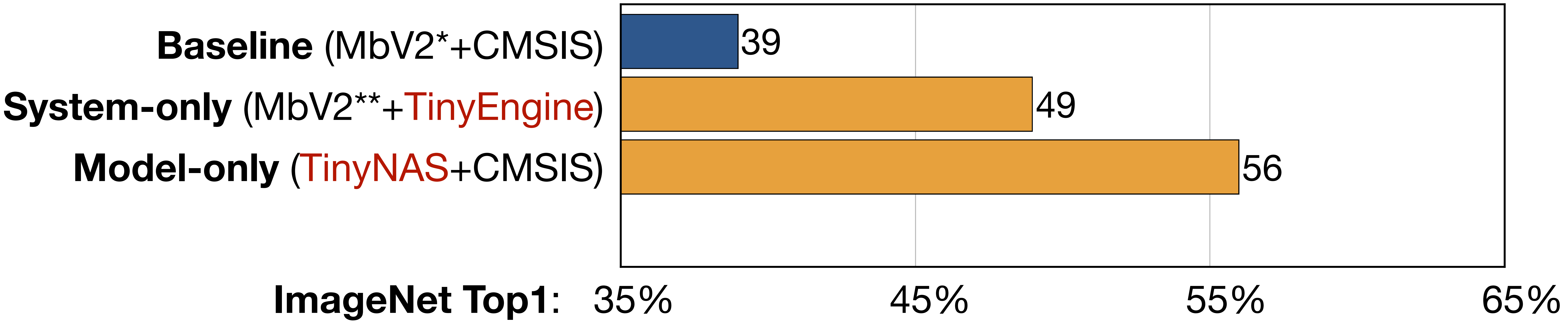
- ImageNet classification on STM32F746 MCU (**320kB SRAM**, **1MB Flash**)



* scaled down version: width multiplier 0.3, input resolution 80

MCUNet: TinyNAS+TinyEngine

- ImageNet classification on STM32F746 MCU (**320kB SRAM**, **1MB Flash**)

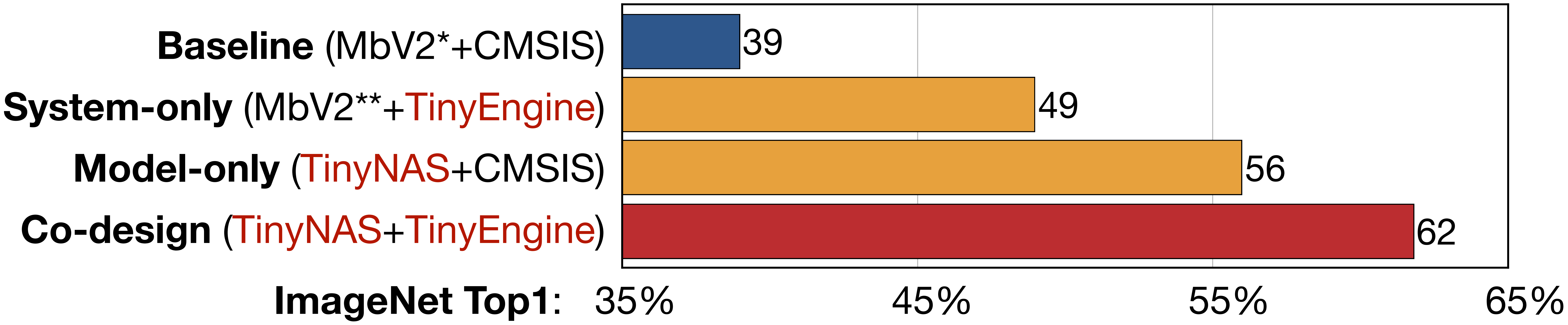


* scaled down version: width multiplier 0.3, input resolution 80

** scaled down version: width multiplier 0.35, input resolution 144

MCUNet: TinyNAS+TinyEngine

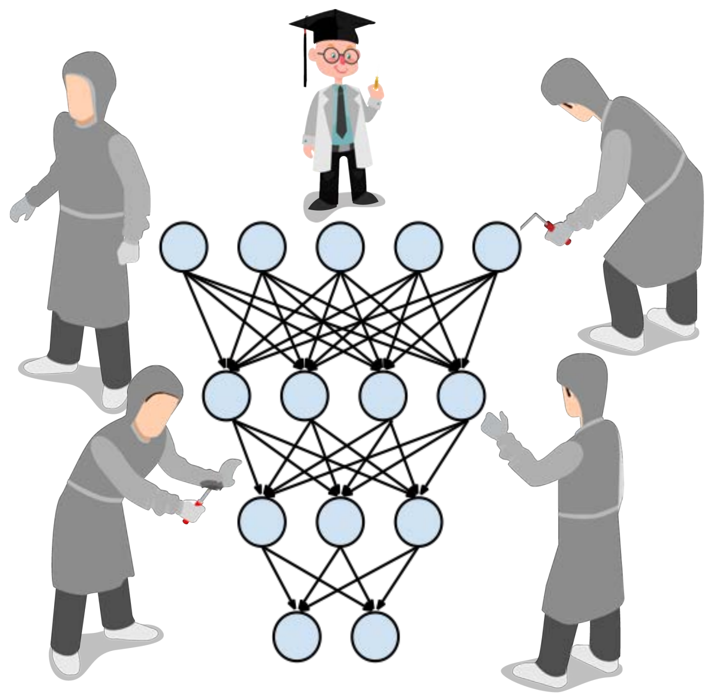
- ImageNet classification on STM32F746 MCU (**320kB SRAM**, **1MB Flash**)



* scaled down version: width multiplier 0.3, input resolution 80

** scaled down version: width multiplier 0.35, input resolution 144

TinyNAS: Neural Architecture Search



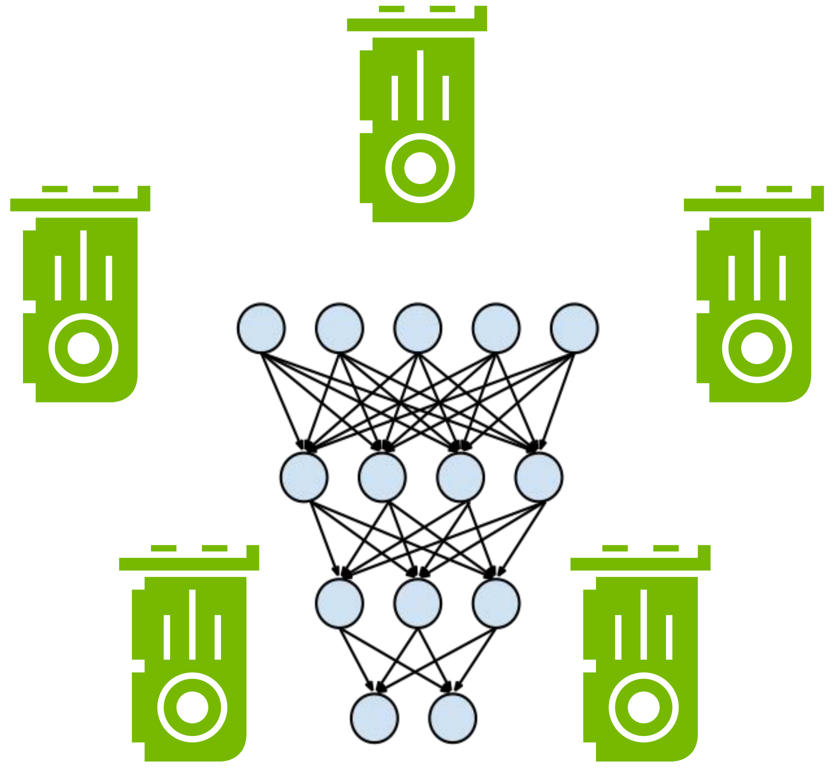
Use Human Expertise

Manual Architecture Design

AutoML

Use Machine Learning

Automatic Architecture Search



Neural Architecture Search (NAS)

Neural Architecture Search

Very expensive: can emit as much carbon as five cars in their lifetimes

Not affordable.

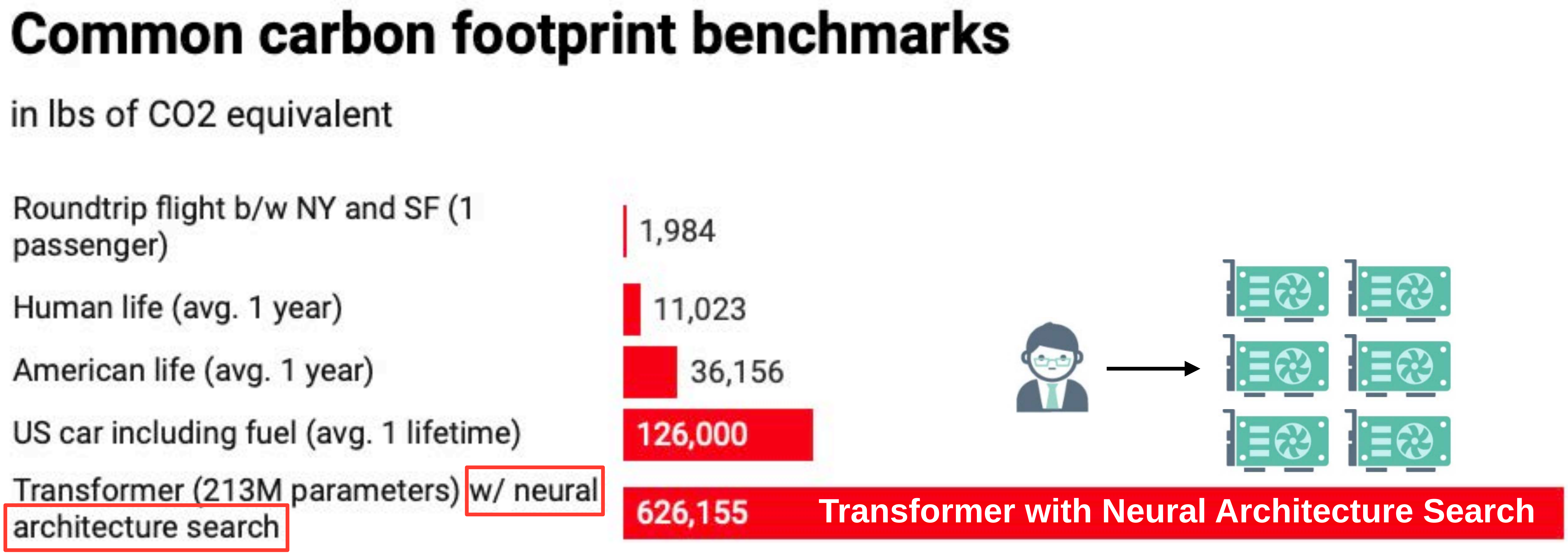
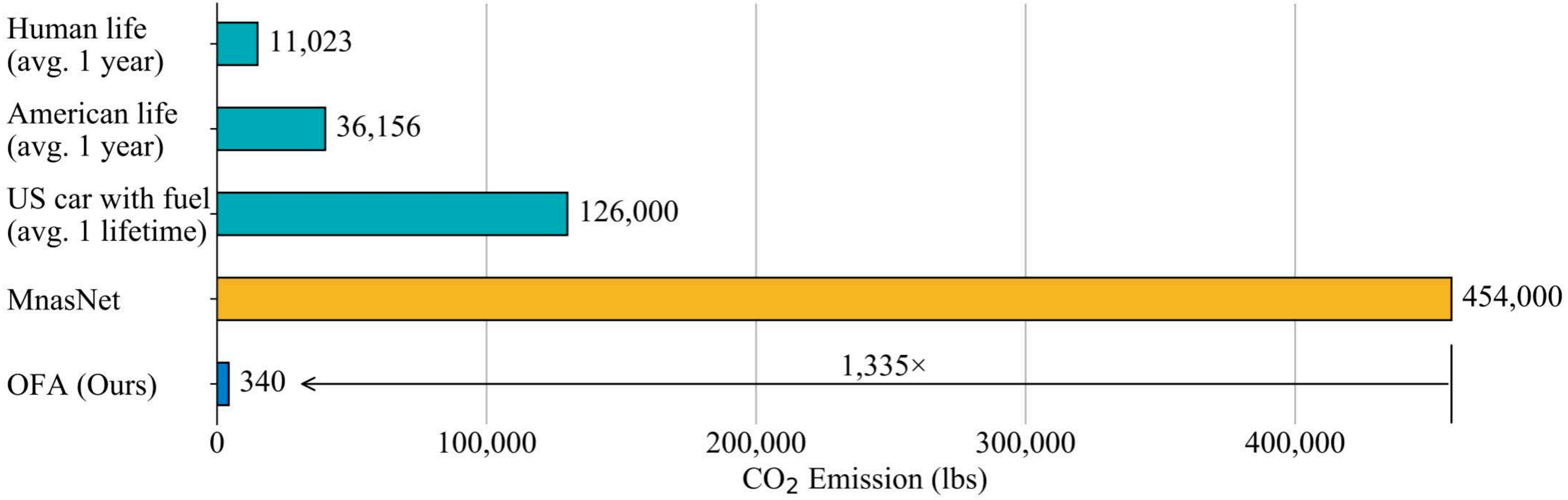


Fig: MIT Technology Review; Data: Strubell et al, ACL'19

Once-for-All Network

Low search cost



- **Six first-place finishes in top competitions in efficient AI**

How to handle diverse MCU platforms?

Cortex M7
STM32H743
(512kB/2MB)

Cortex M7
STM32F746
(320kB/1MB)

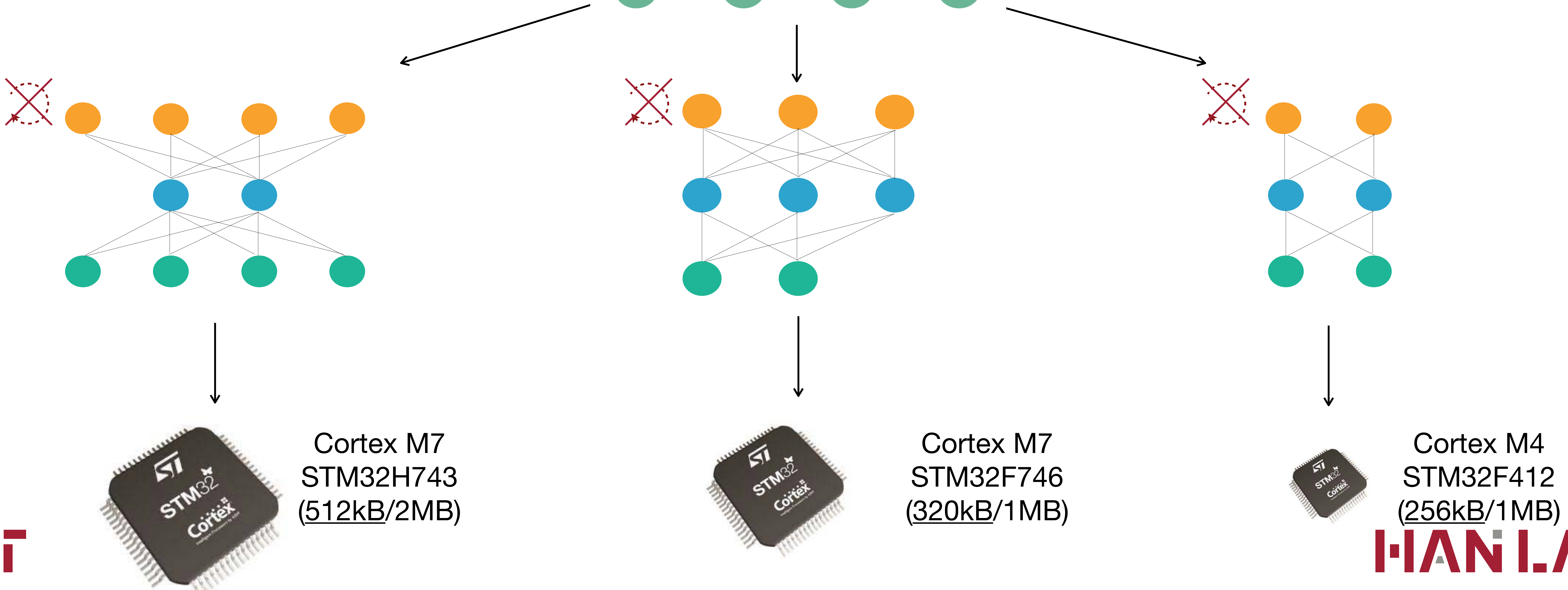
Cortex M4
STM32F412
(256kB/1MB)

Once-for-All Network

Train once, get many

Fit diverse hardware constraints

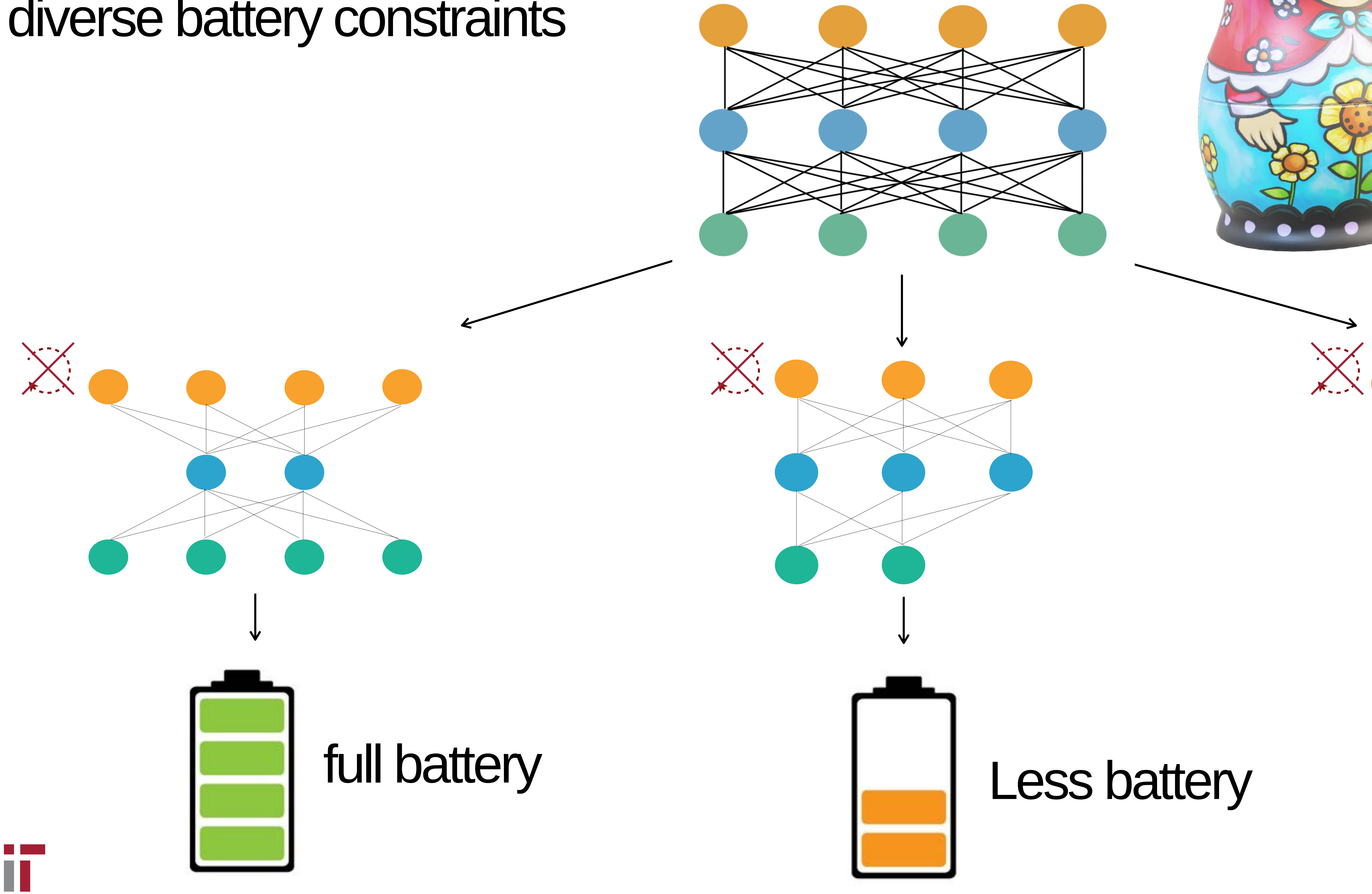
Smaller child networks are nested in larger ones



Once-for-All Network

Train once, get many
Fit diverse battery constraints

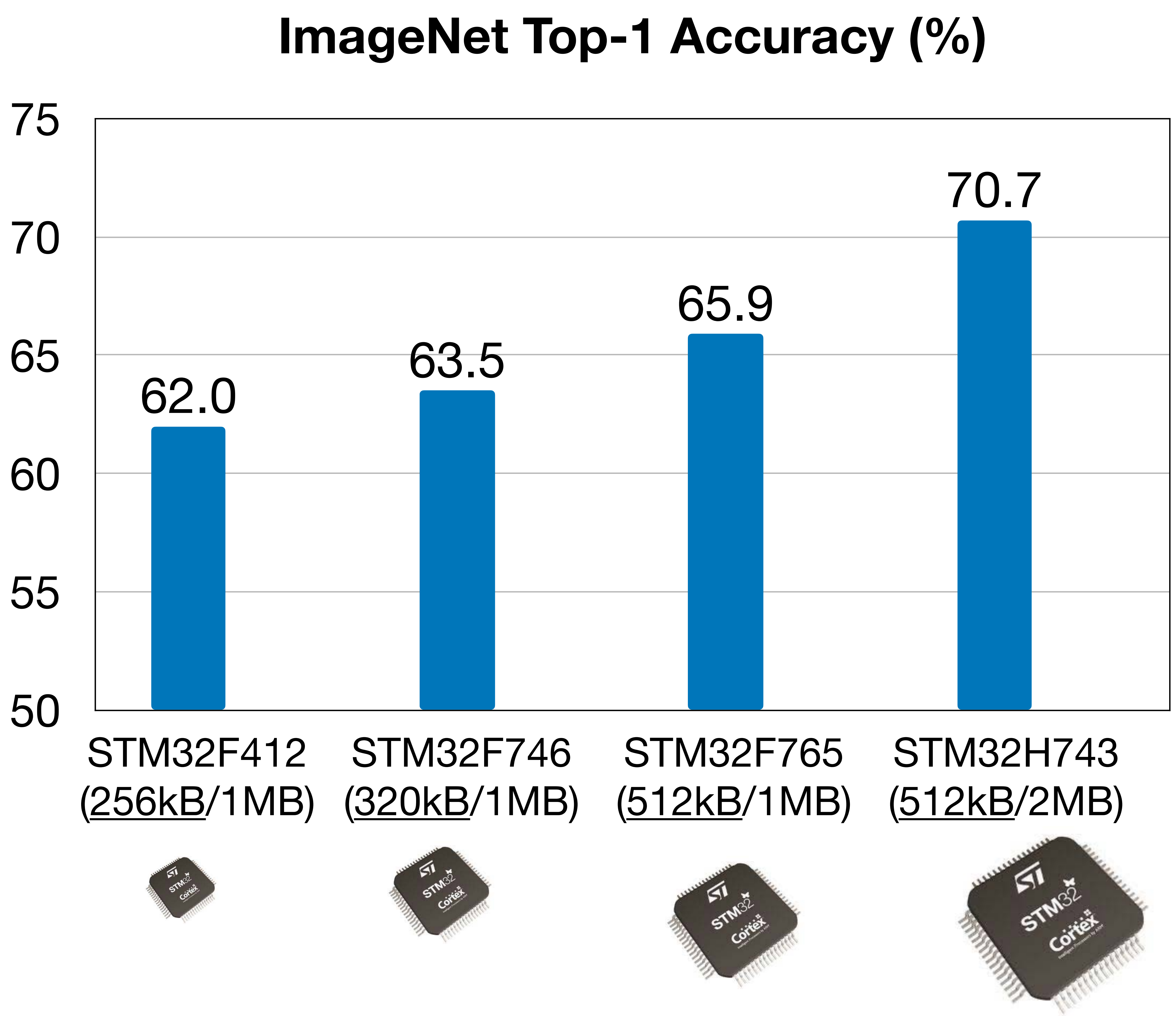
Smaller child networks are nested in larger ones



battery-saving mode

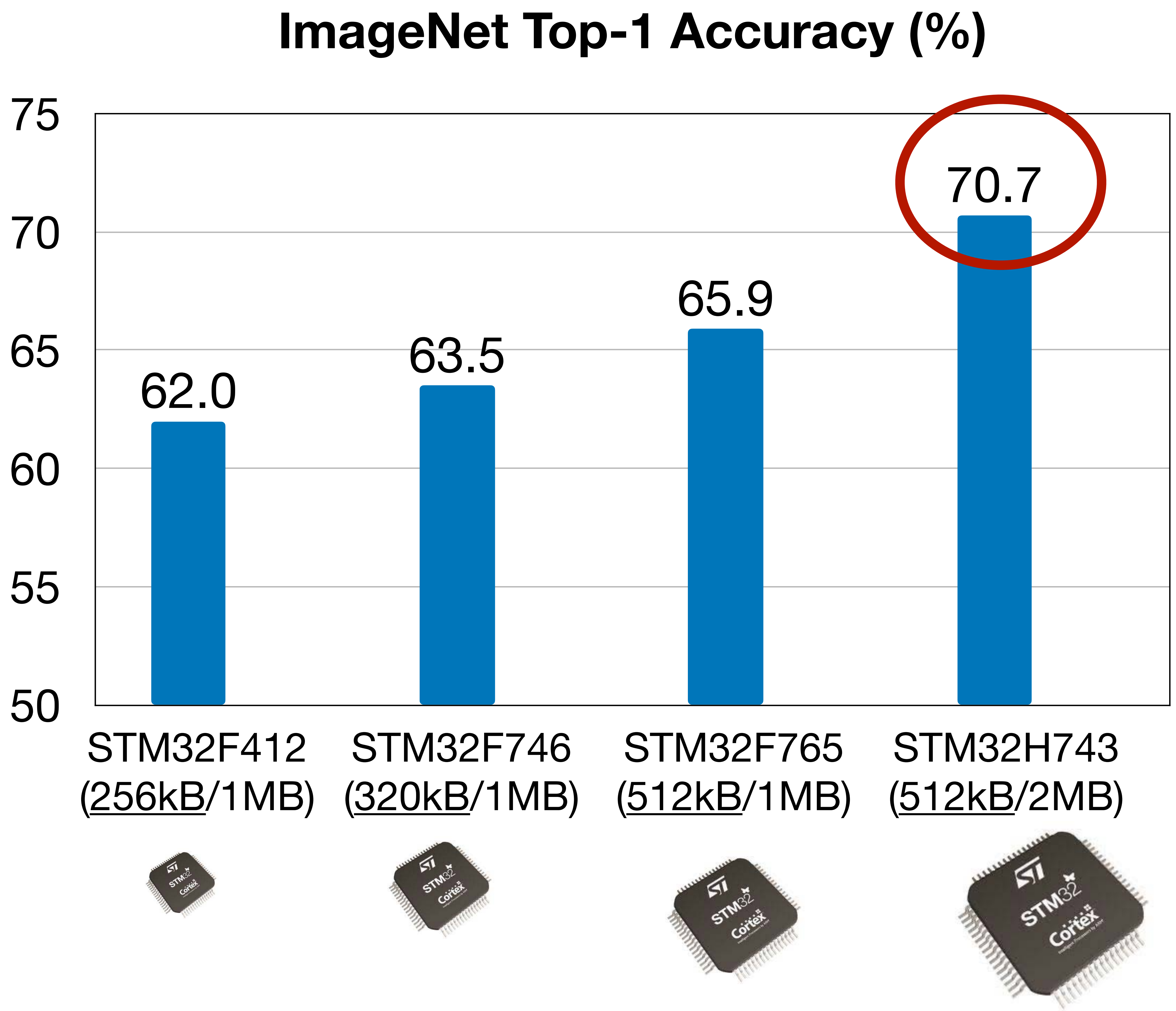
Once-for-All Network

- Specializing models (int4) for different MCUs (SRAM/Flash)



Once-for-All Network

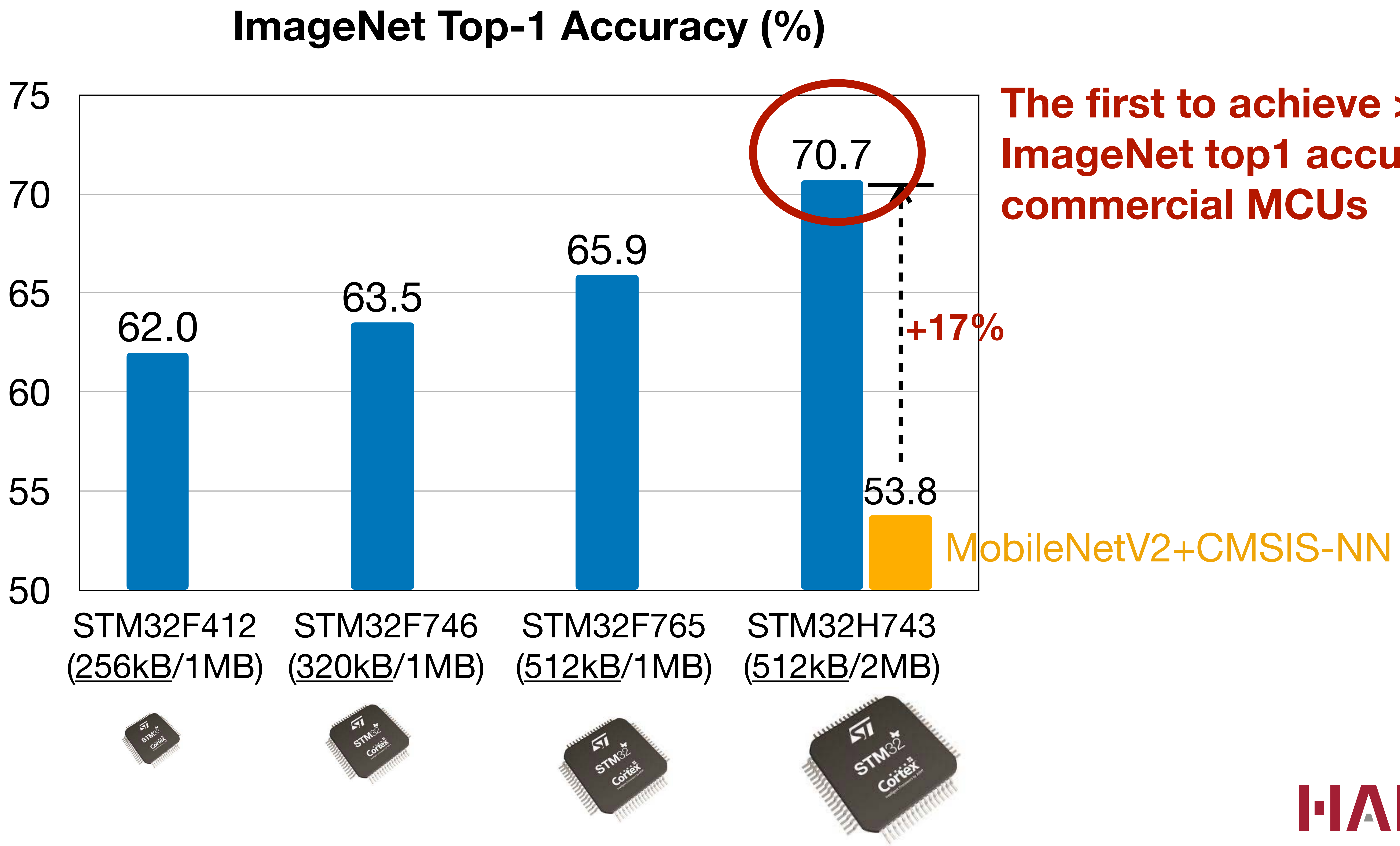
- Specializing models (int4) for different MCUs (SRAM/Flash)



The first to achieve >70% ImageNet top1 accuracy on commercial MCUs

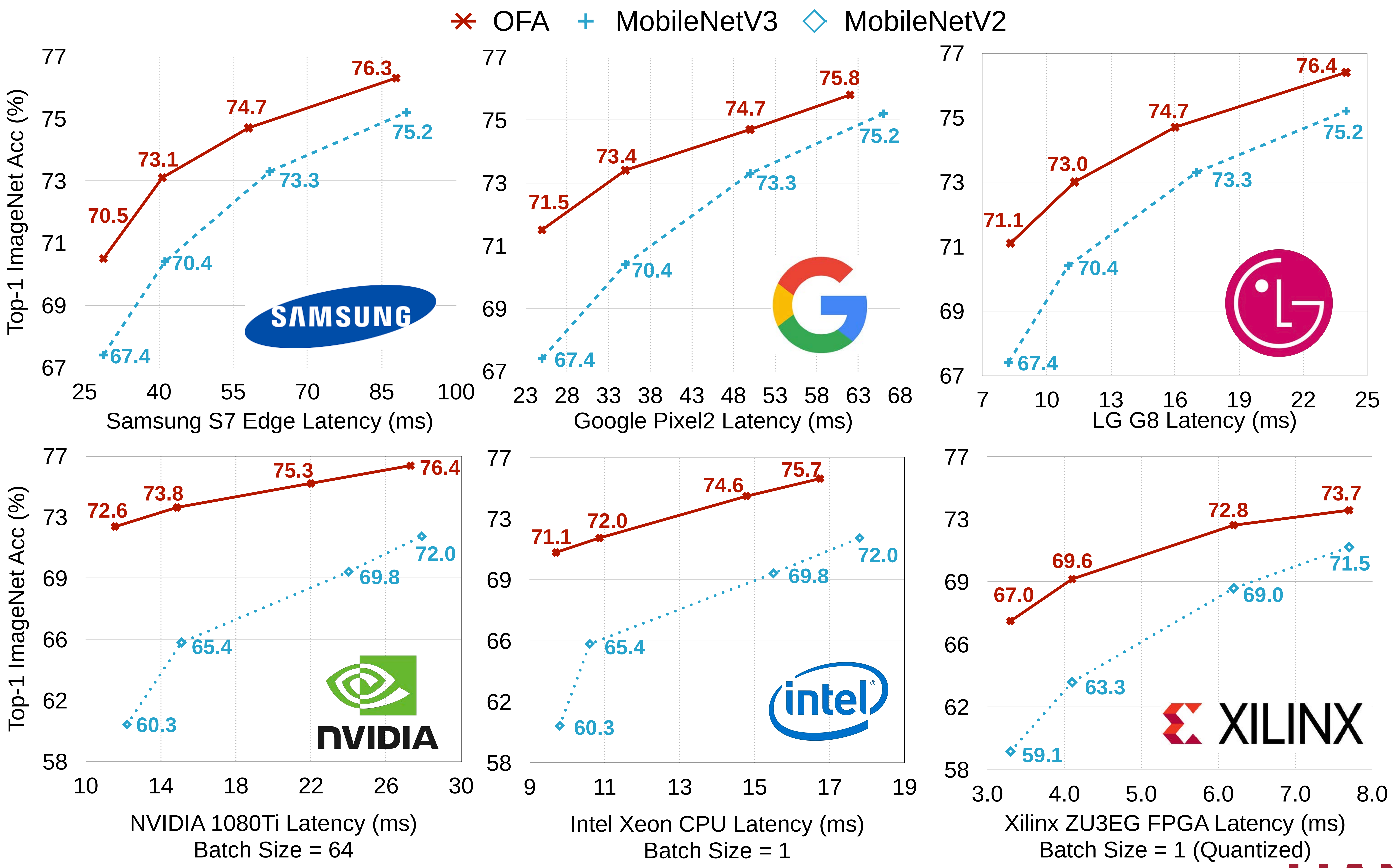
Once-for-All Network

- Specializing models (int4) for different MCUs (SRAM/Flash)



The first to achieve >70% ImageNet top1 accuracy on commercial MCUs

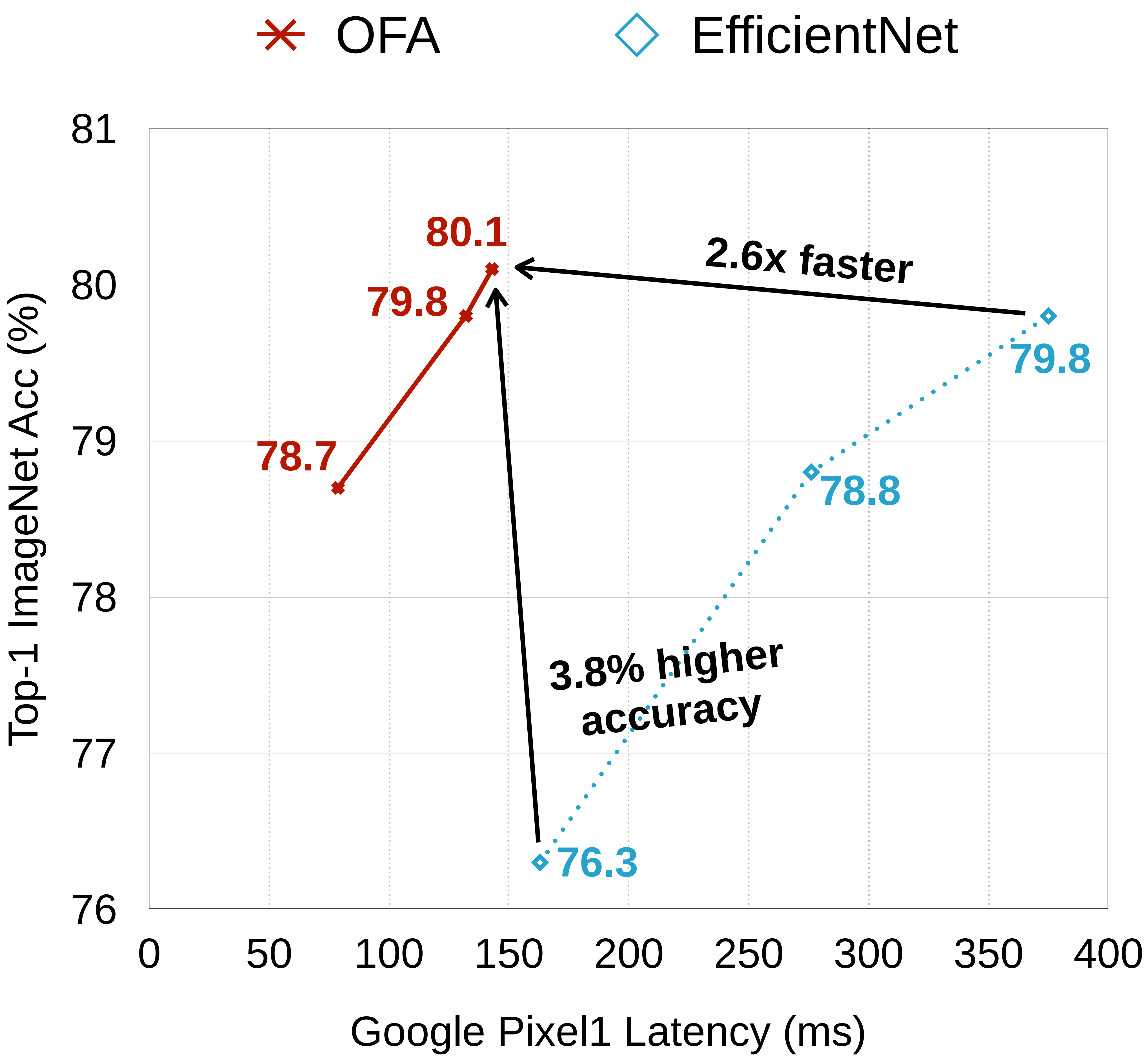
Once-for-All Network



Once-for-All, ICLR'20

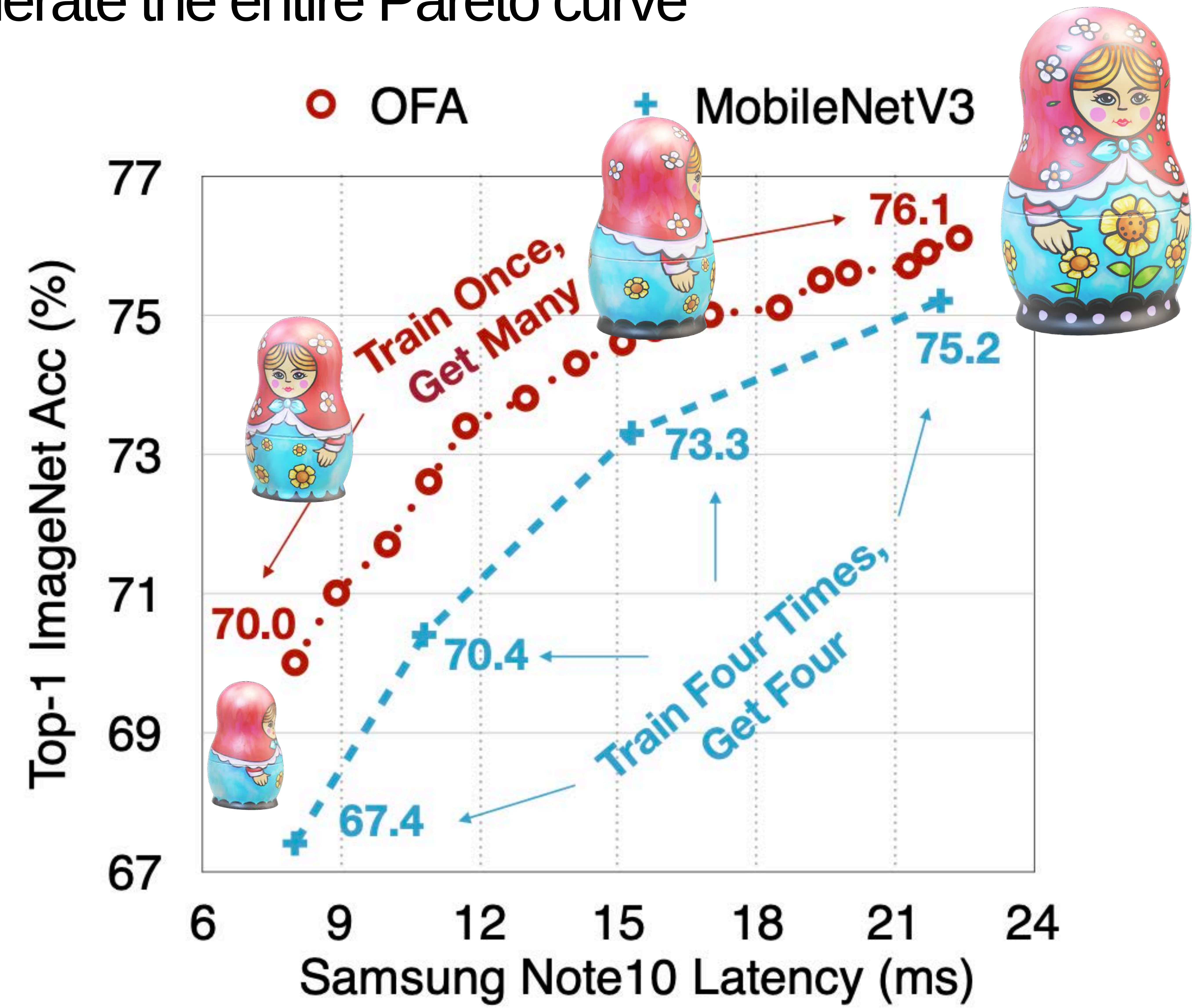
Once-for-All Network

Train only once, handle diverse hardware constraints

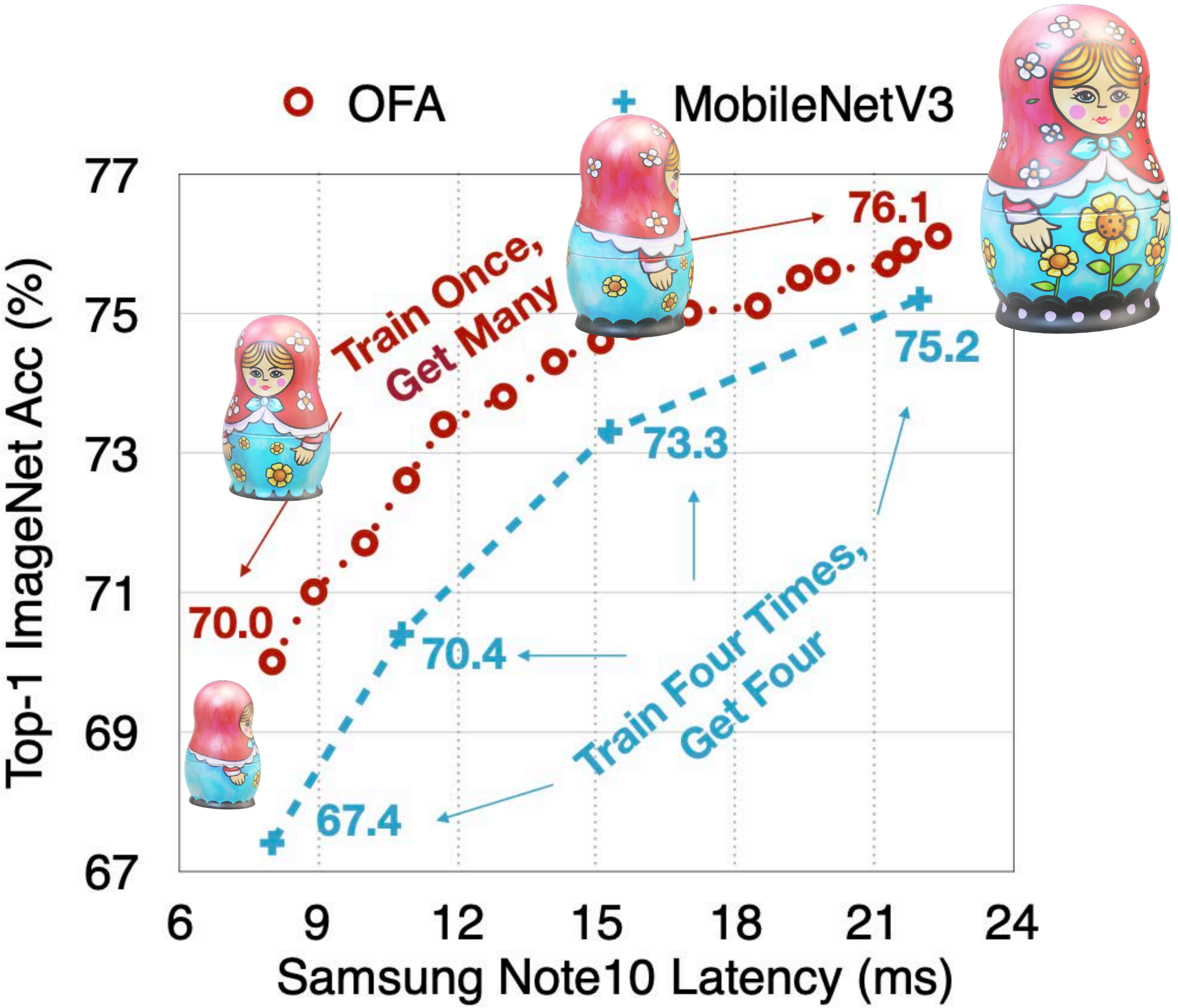


Once-for-All Network

Train only once, generate the entire Pareto curve



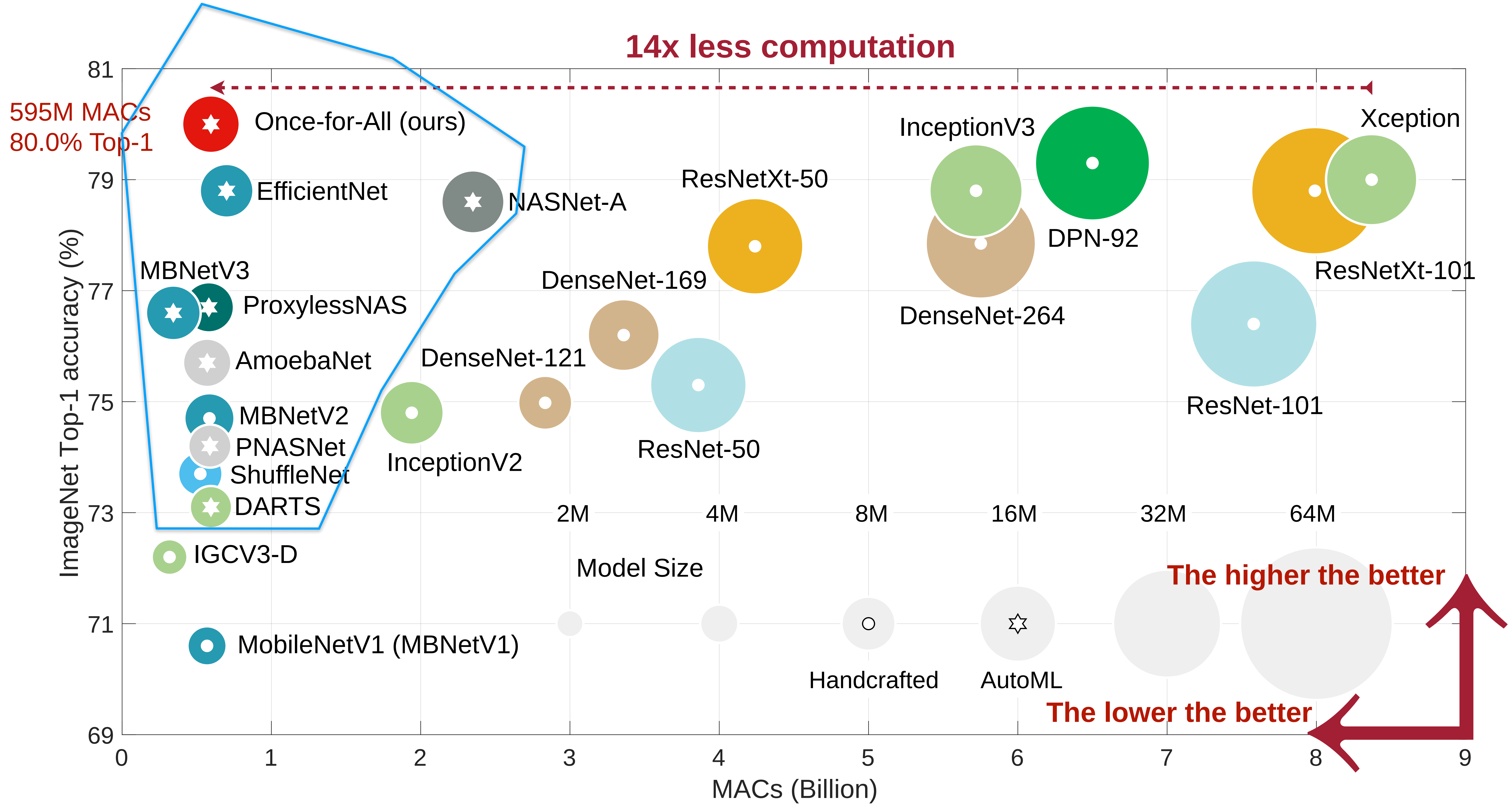
Once-for-All, ICLR'20



OFA
MobileNetV3
77
75
73
71
69
67
Top-1 ImageNet Acc (%)
6
9
12
15
18
21
24
Samsung Note10 Latency (ms)
Train Once, Get Many
76.1
70.0
75.2
73.3
70.4
67.4
Train Four Times, Get Four

Once-for-All Network

Trade-off of accuracy and MACs



- Once-for-all model (ofa.mit.edu) sets a new state-of-the-art **80% ImageNet top-1 accuracy** under the mobile vision setting (< 600M MACs).

Award Winning Technology

CPU detection
FPGA detection

5th Low-Power Computer Vision Challenge

CPU classification CPU detection

4th Low-Power Computer Vision Challenge

DSP Recognition

3rd Low-Power Computer Vision Challenge

Visual Wake Words
on TF-lite

Visual Wake Words Challenge @CVPR 2019

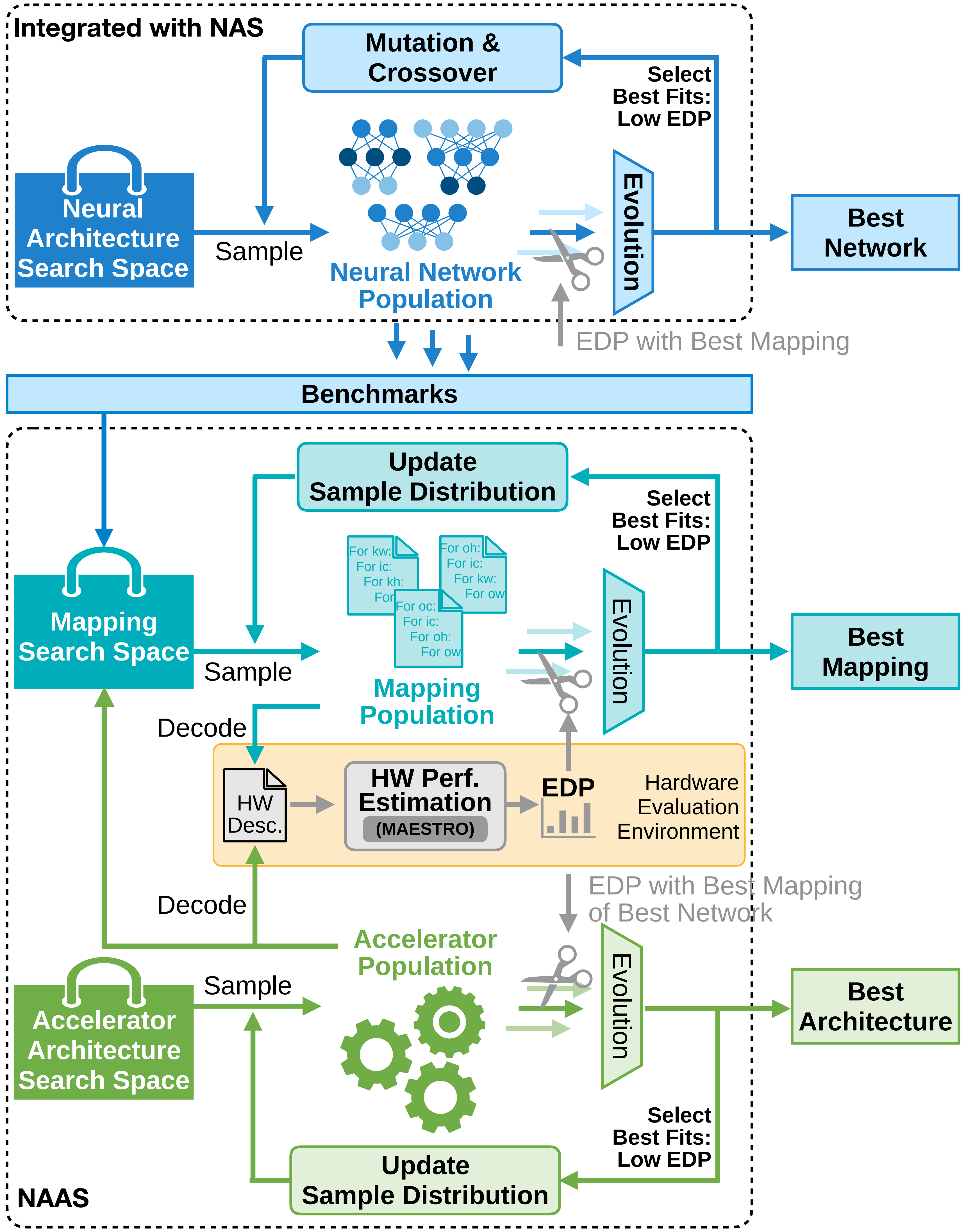
3D Semantic
Segmentation

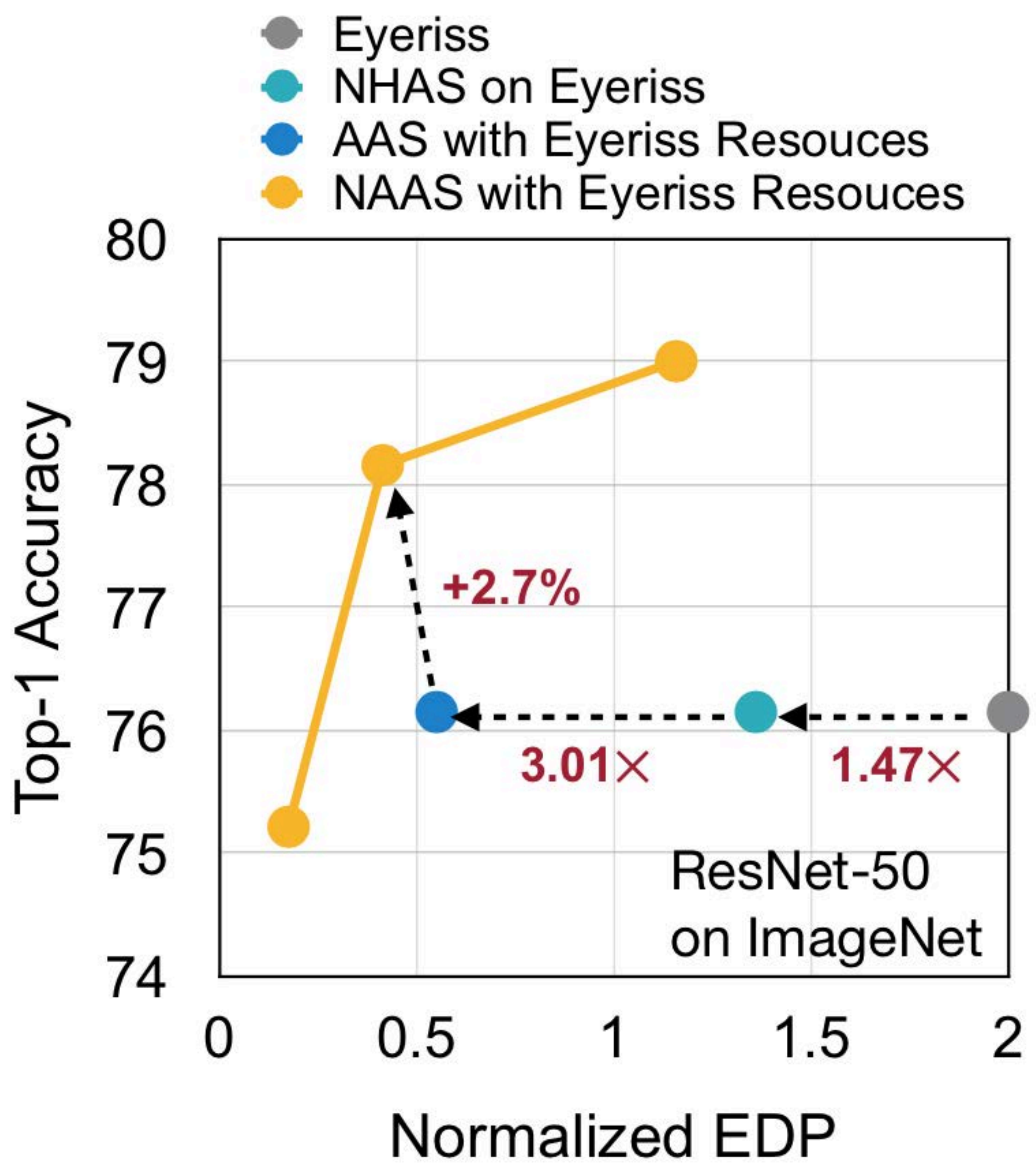
SemanticKITTI

NLP track
Language Model

MicroNet Challenge @NeurIPS 2019

NAAS: Neural Accelerator Architecture Search





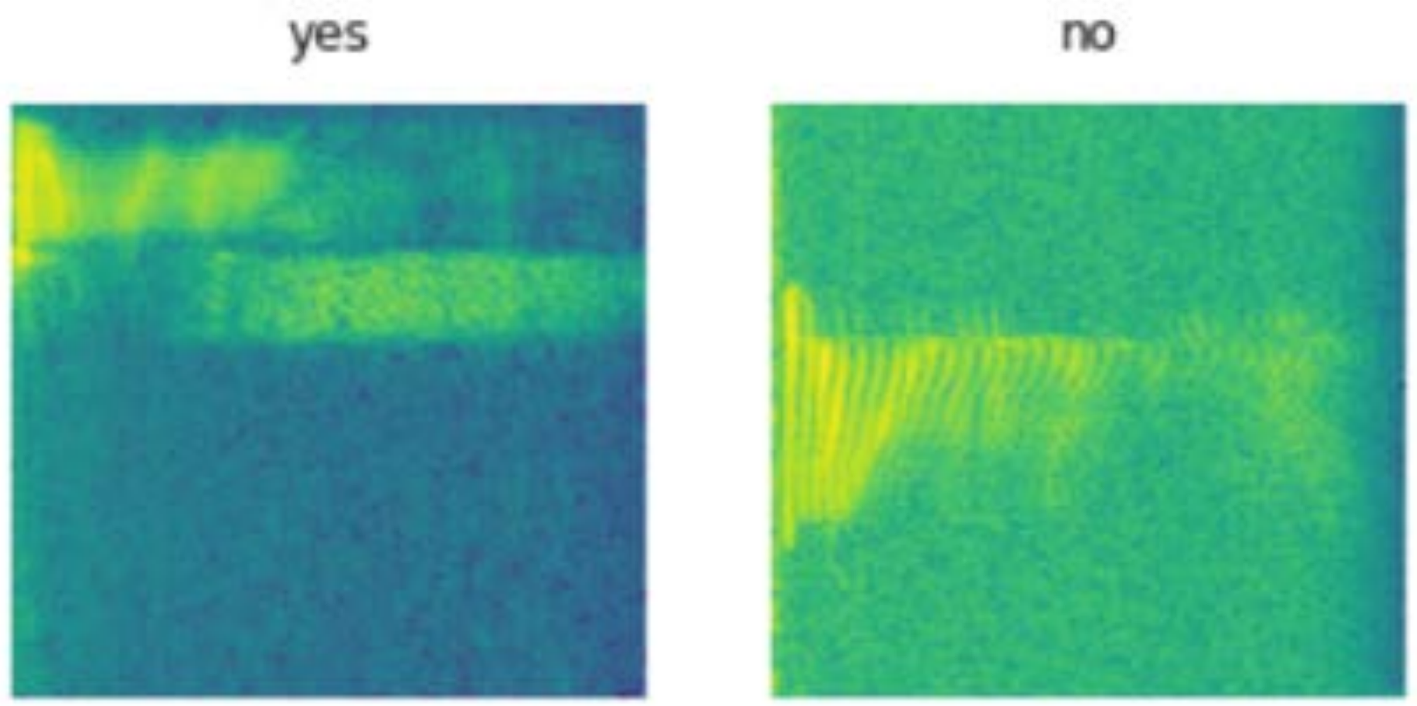
Applications

We focus on large-scale datasets to reflect real-life use cases.

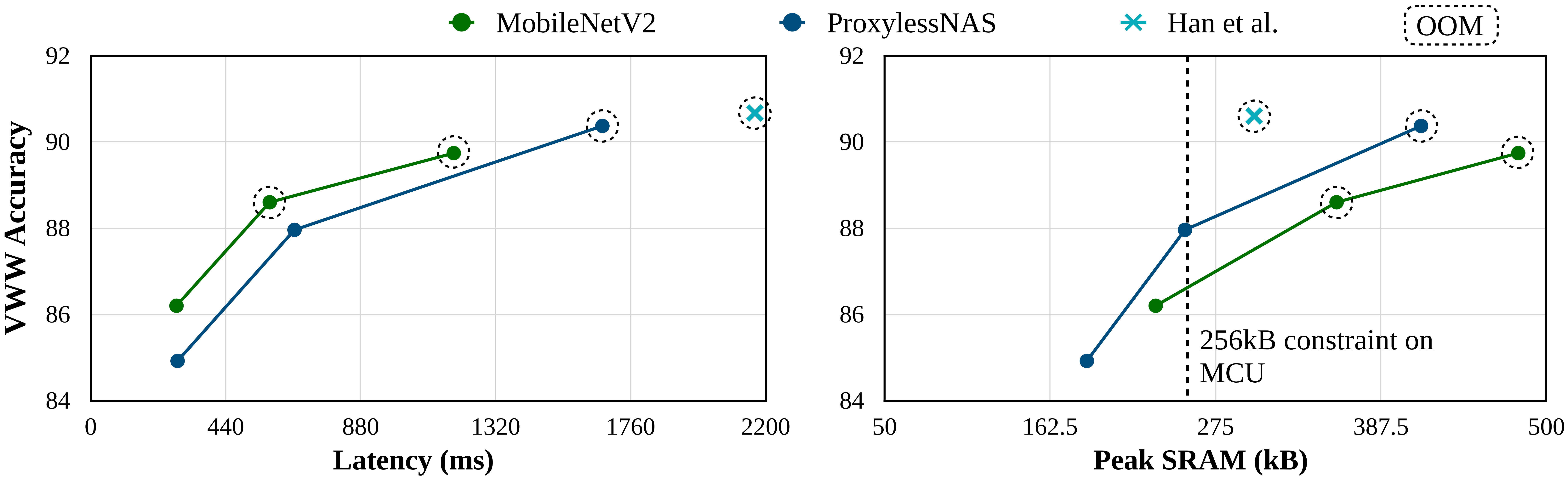
Datasets:

(1) ImageNet-1000
(2) Wake Words
- Visual: Visual Wake Words
- Audio: Google Speech Commands

(a) 'Person' (b) 'Not-person'



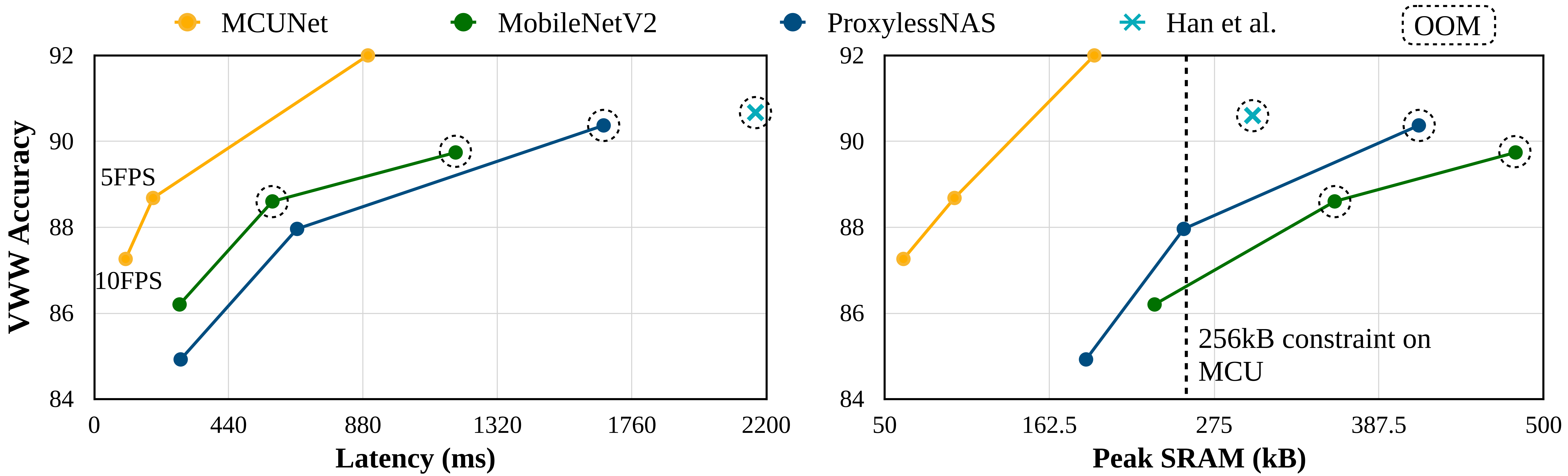
Visual Wake Words (VWW)



(a) Trade-off: accuracy *vs.* measured latency

(b) Trade-off: accuracy *vs.* peak memory

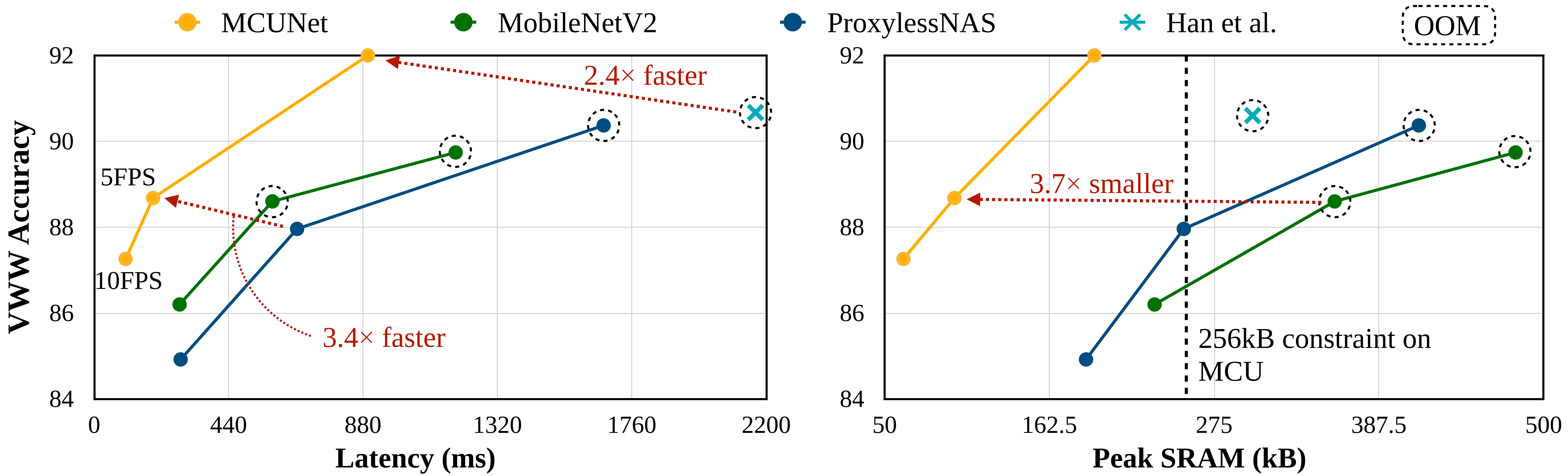
Visual Wake Words (VWW)



(a) Trade-off: accuracy *vs.* measured latency

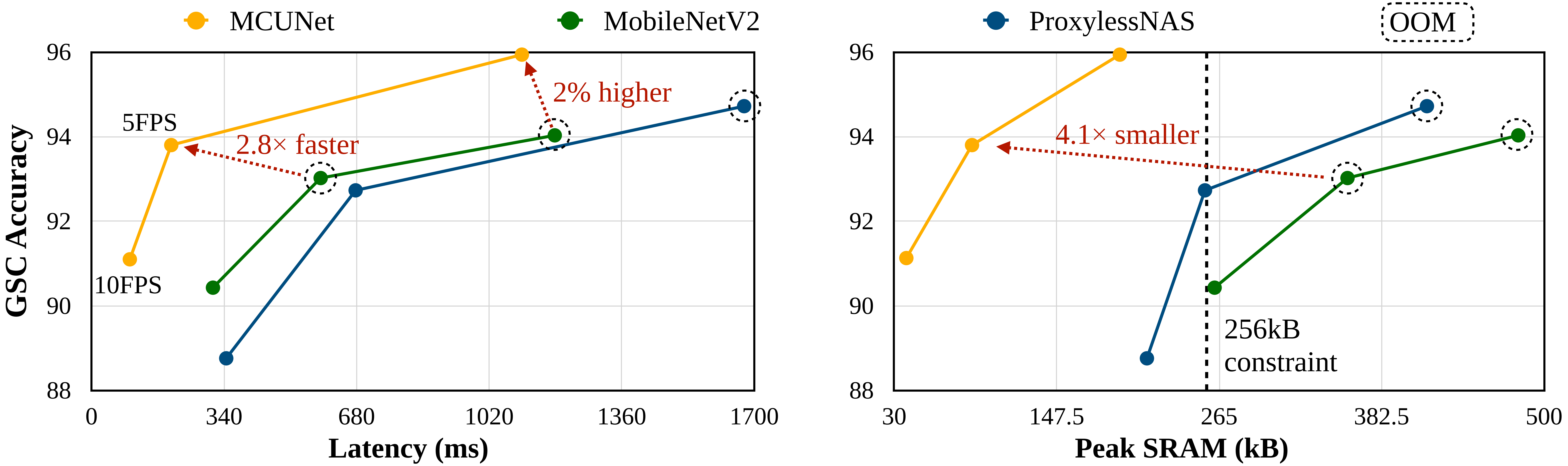
(b) Trade-off: accuracy *vs.* peak memory

Visual Wake Words (VWW)



(a) Trade-off: accuracy *vs.* measured latency
(b) Trade-off: accuracy *vs.* peak memory

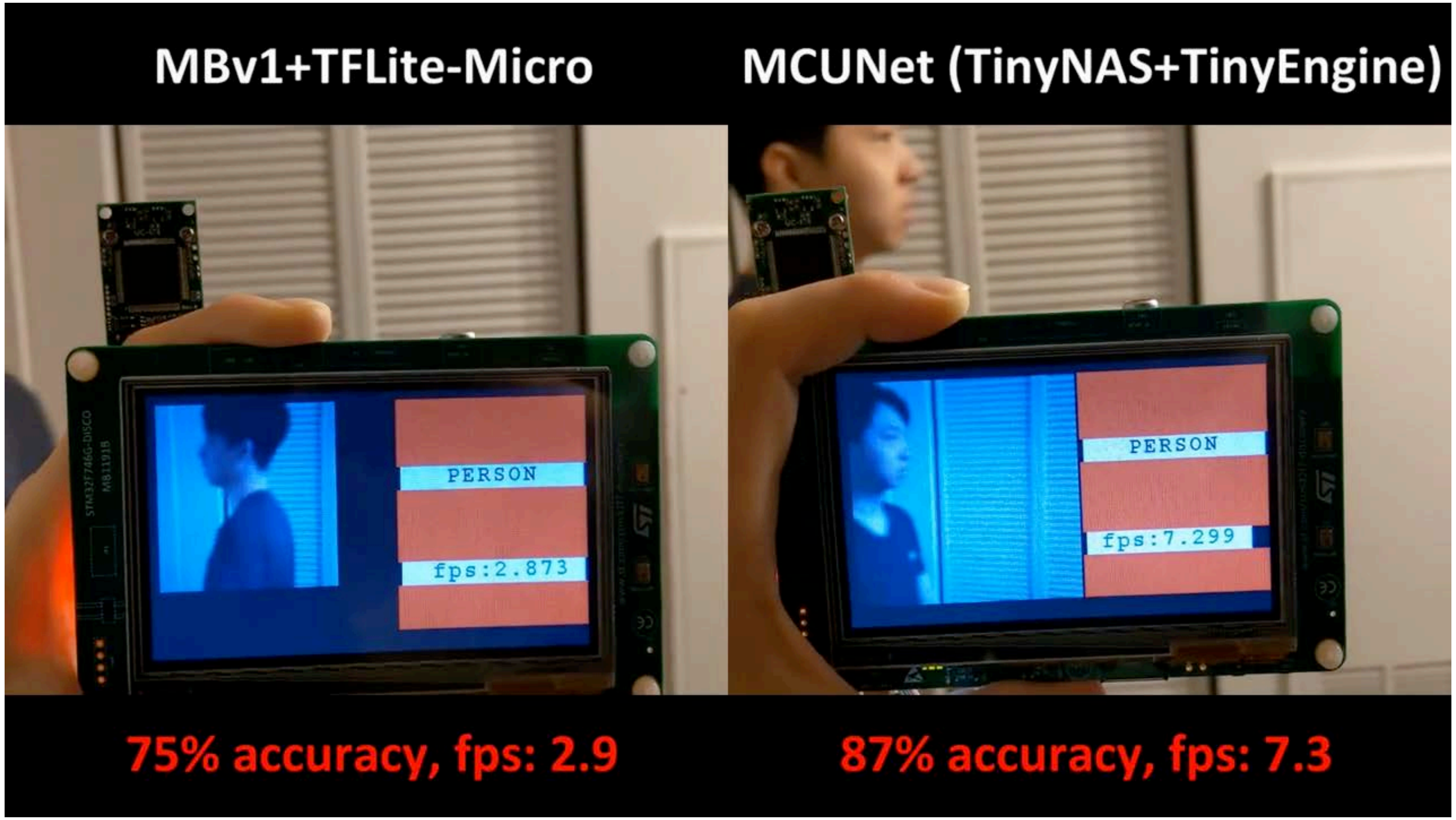
Audio Wake Words (Speech Commands)



(a) Trade-off: accuracy *vs.* measured latency

(b) Trade-off: accuracy *vs.* peak memory

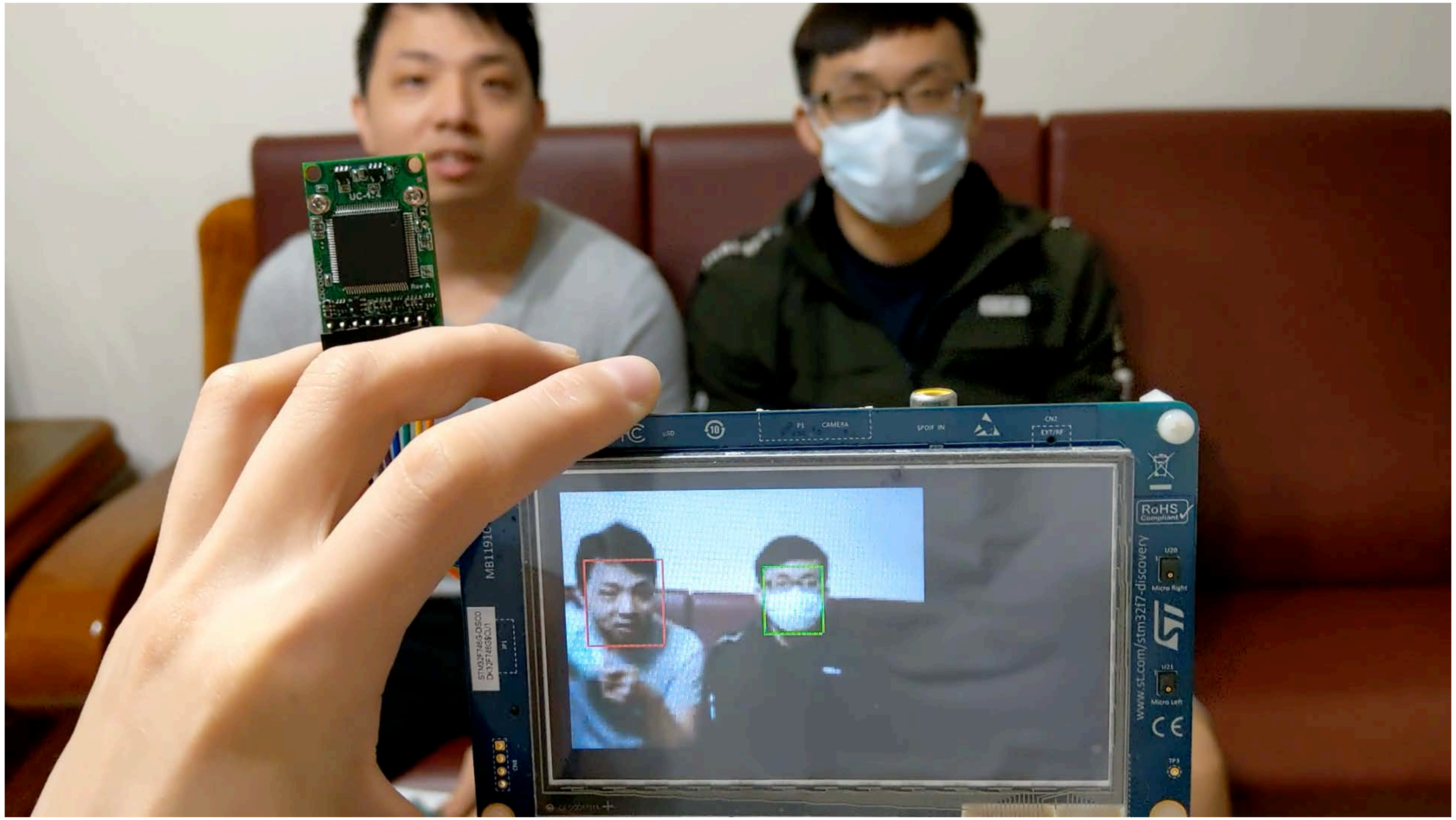
Demo: Visual Wake Words on MCU



- Detecting if there is person

- STM32F746
- 320KB SRAM
- 1MB Flash
- ARM Cortex-M7 @216MHz

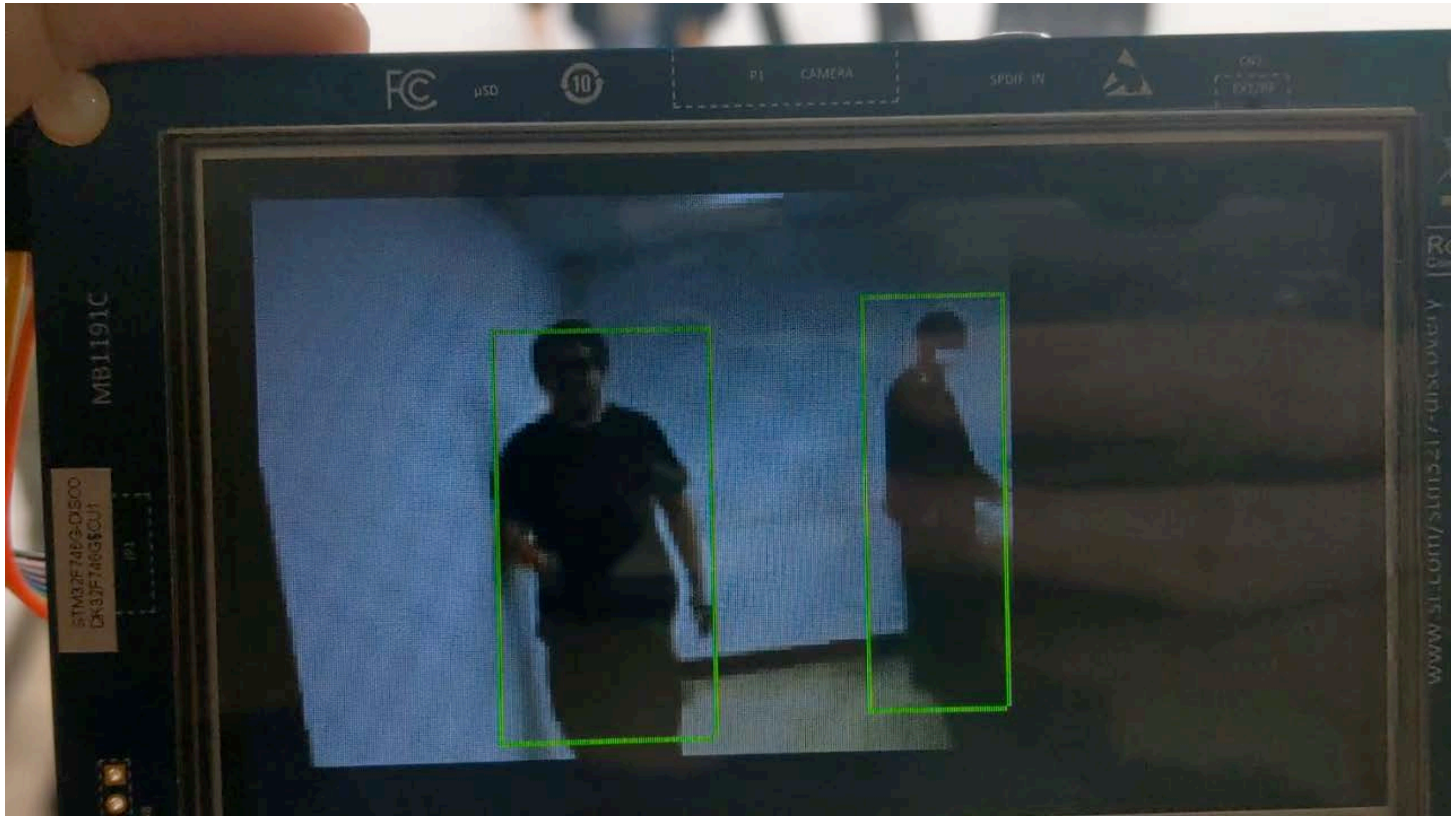
Demo: Face Mask Detection on MCU



- Detecting faces & masks

- STM32F746
- 320KB SRAM
- 1MB Flash
- ARM Cortex-M7 @216MHz

Demo: Person Detection on MCU



- Detecting persons

- STM32F746
- 320KB SRAM
- 1MB Flash
- ARM Cortex-M7 @216MHz

Grocery Shelf Detection

Model size: 37KB (compared with MobileNet-v2: 3.5MB)
Computation: 352MOPs on 608x608 input resolution.

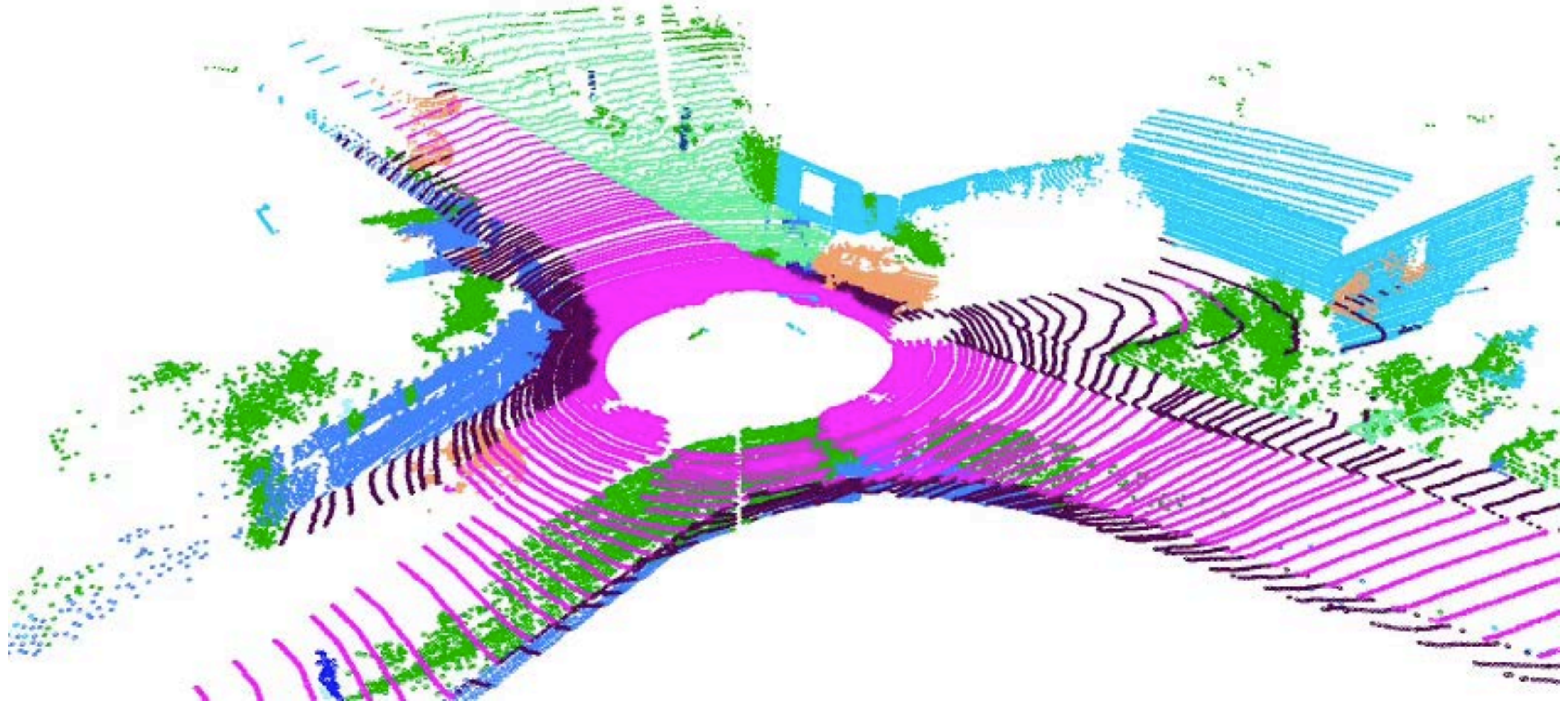
TinyML for Point Cloud

SPVNAS, ECCV'20

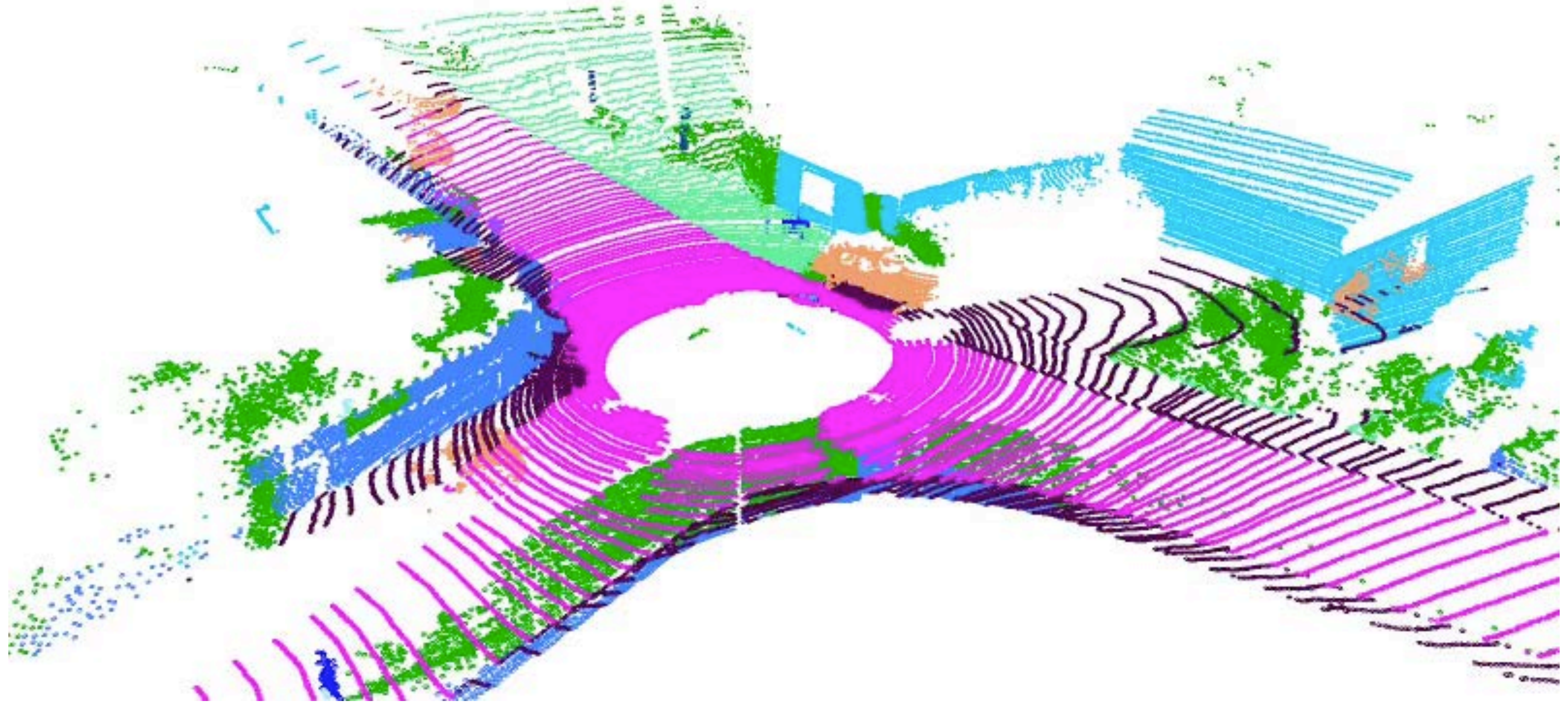
AR/VR: a whole backpack of computer

Self-driving: a whole trunk of GPU

Mobile phone: limited battery



MinkowskiNet: **3.4** FPS



SPVNAS (Ours): **9.1** FPS

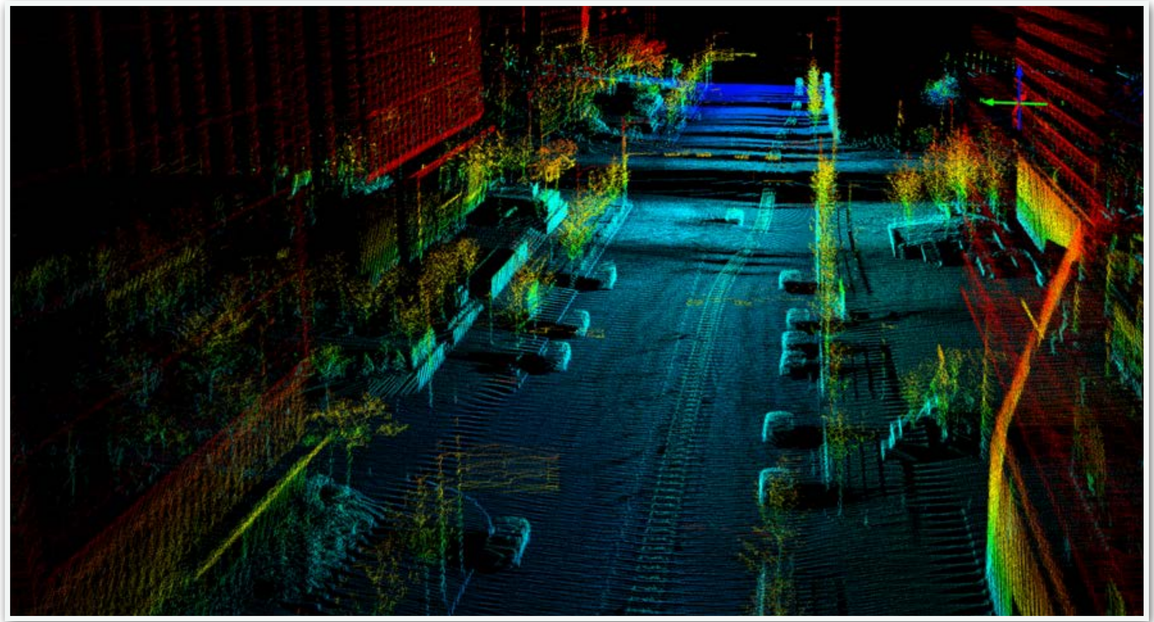
accuracy ranks 1st on the SemanticKitti leaderboard

Approach	Paper	Code	mIoU	Classes (IoU)
SPVNAS			67.0	
TORNADONet			63.1	
KPRNet			63.1	
Cylinder3D			61.8	
FusionNet			61.3	
SalsaNext			59.5	
KPConv			58.8	
SqueezeSegV3			55.9	

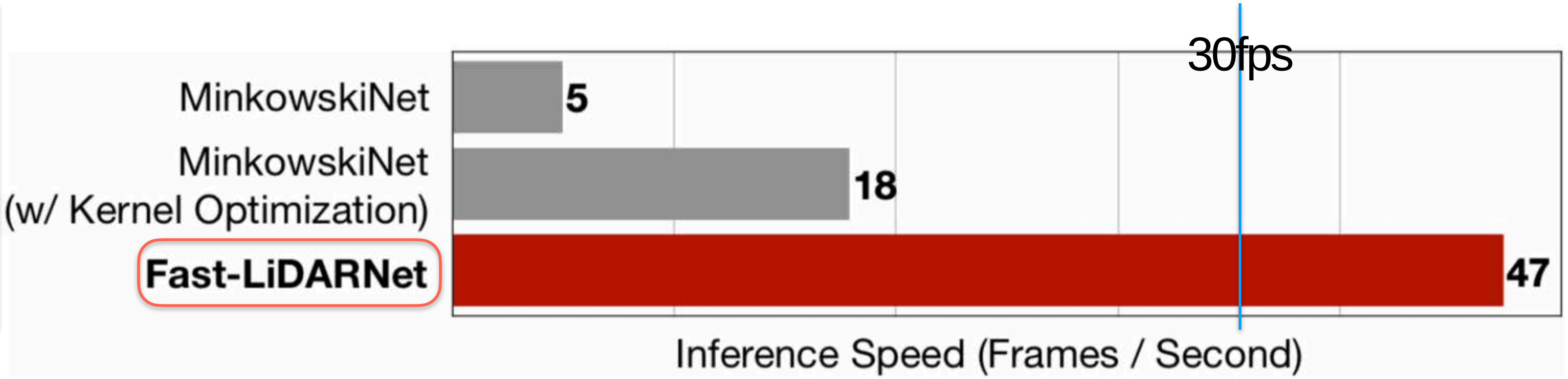
TinyML for Driving

Liu et al. ICRA’21.

3D LiDAR Sensor



3D Point Cloud: 2M points/s

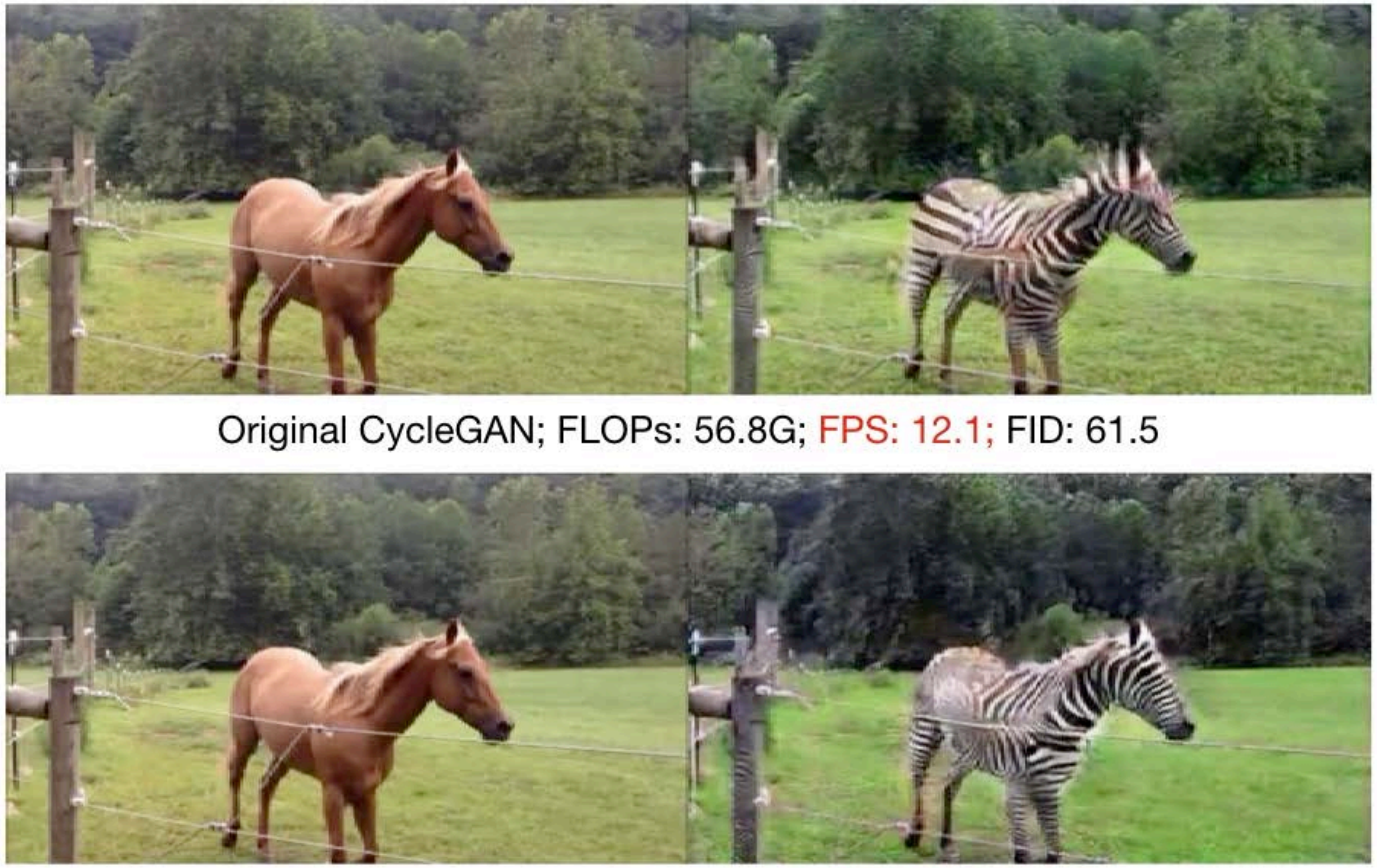


Demo:

TinyML for GAN

Accelerating Horse2zebra by GAN Compression

Demo:



Original CycleGAN; FLOPs: 56.8G; FPS: 12.1; FID: 61.5

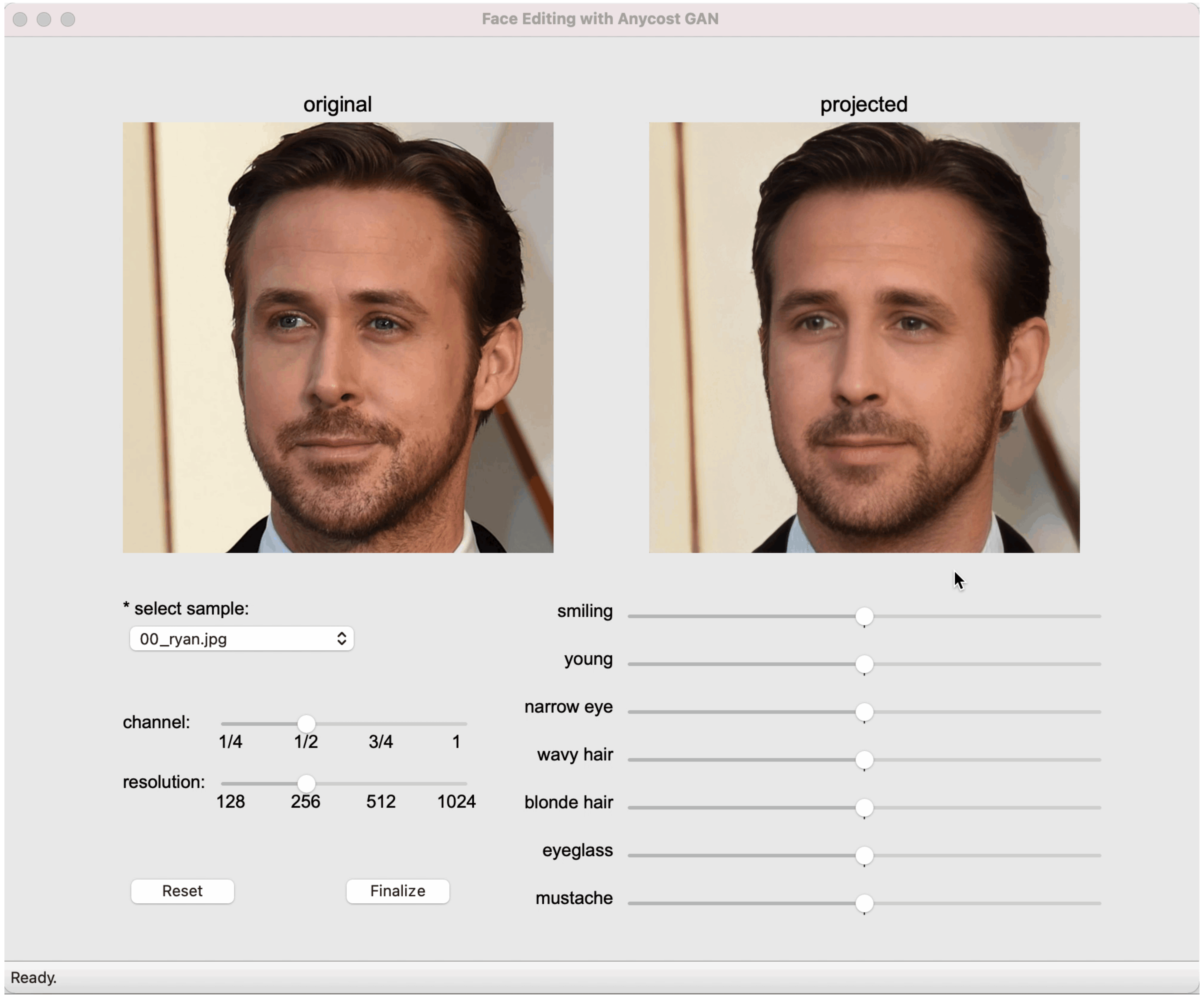
GAN Compression; FLOPs: 3.50G (16.2x); FPS: 40.0 (3.3x); FID: 53.6

TinyML for GANs

AnyCost GAN, CVPR'21

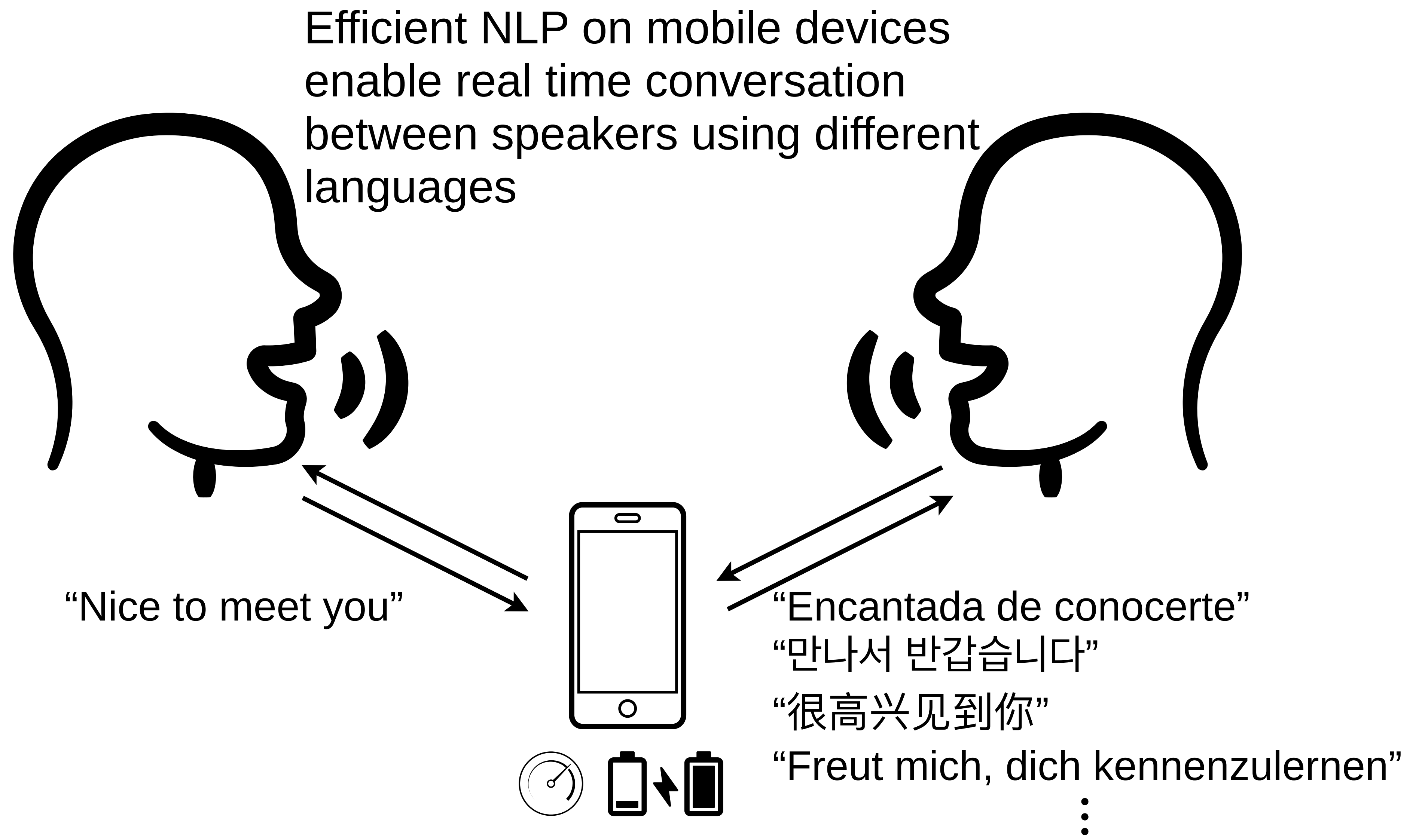
TinyML for GANs

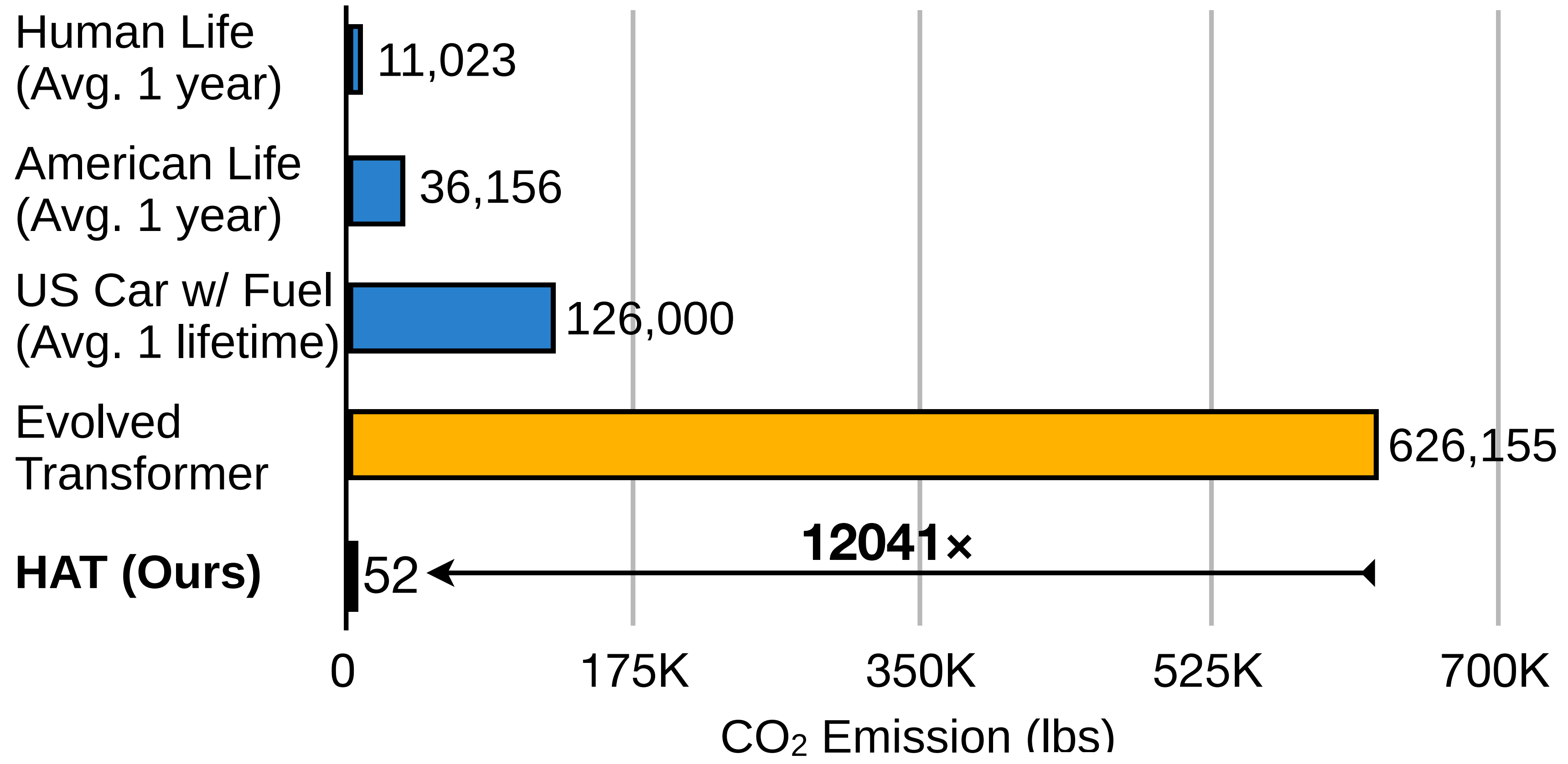
AnyCost GAN, CVPR'21

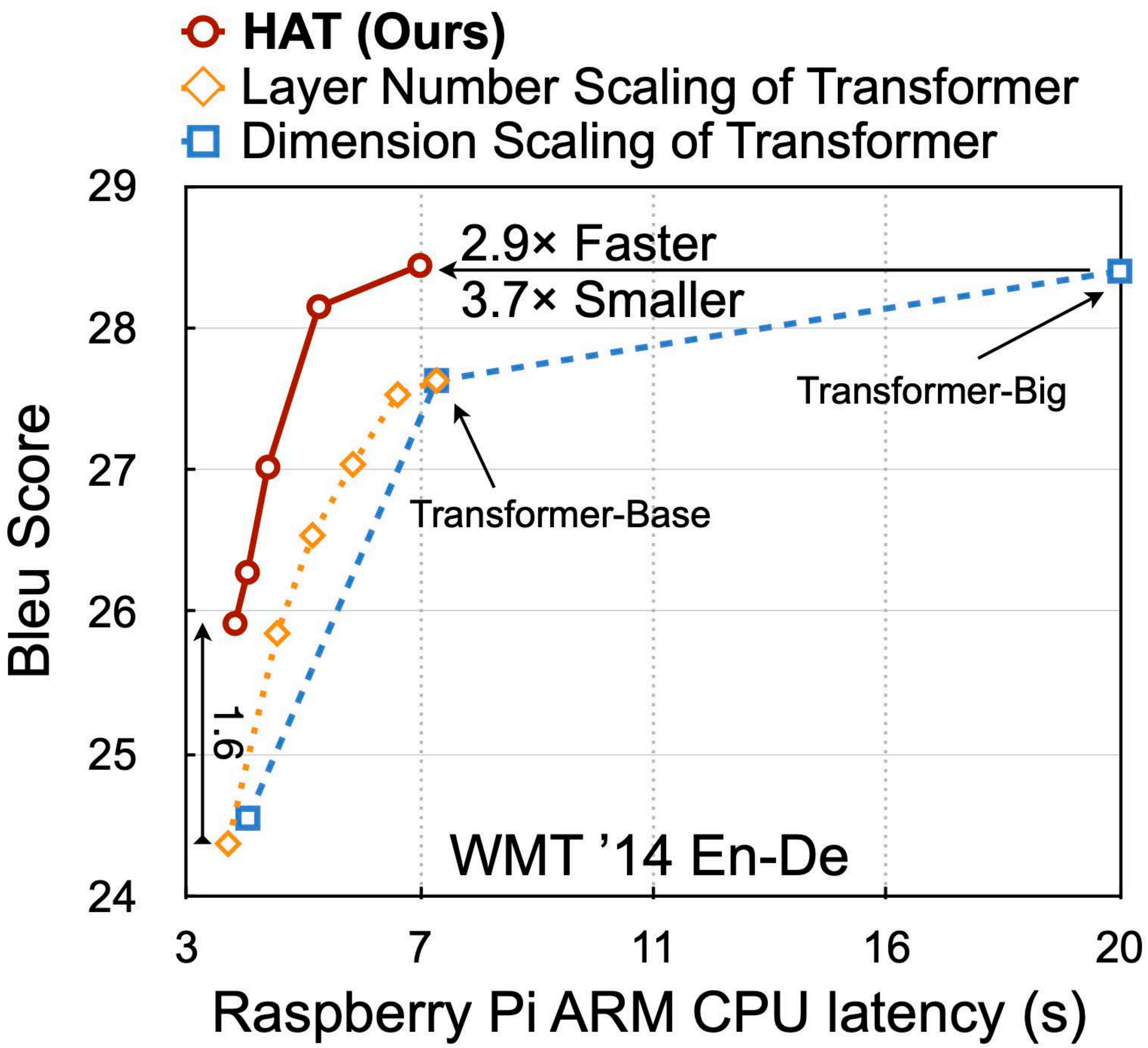


TinyML for NLP

HAT, ACL'20
SpAtten, HPCA'21

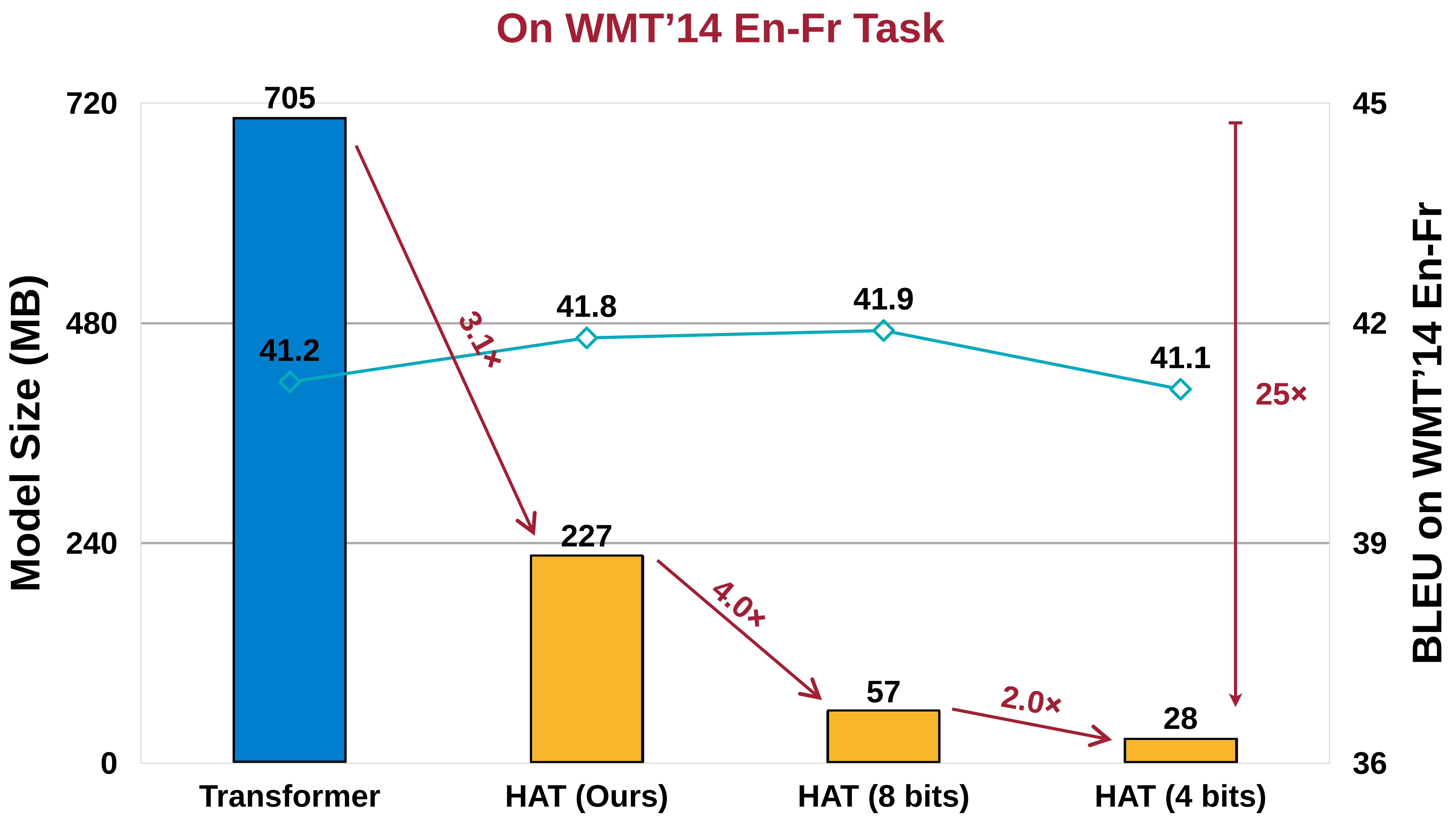






TinyML for NLP

HAT, ACL'20
SpAtten, HPCA'21



- HAT is orthogonal to general model compression techniques

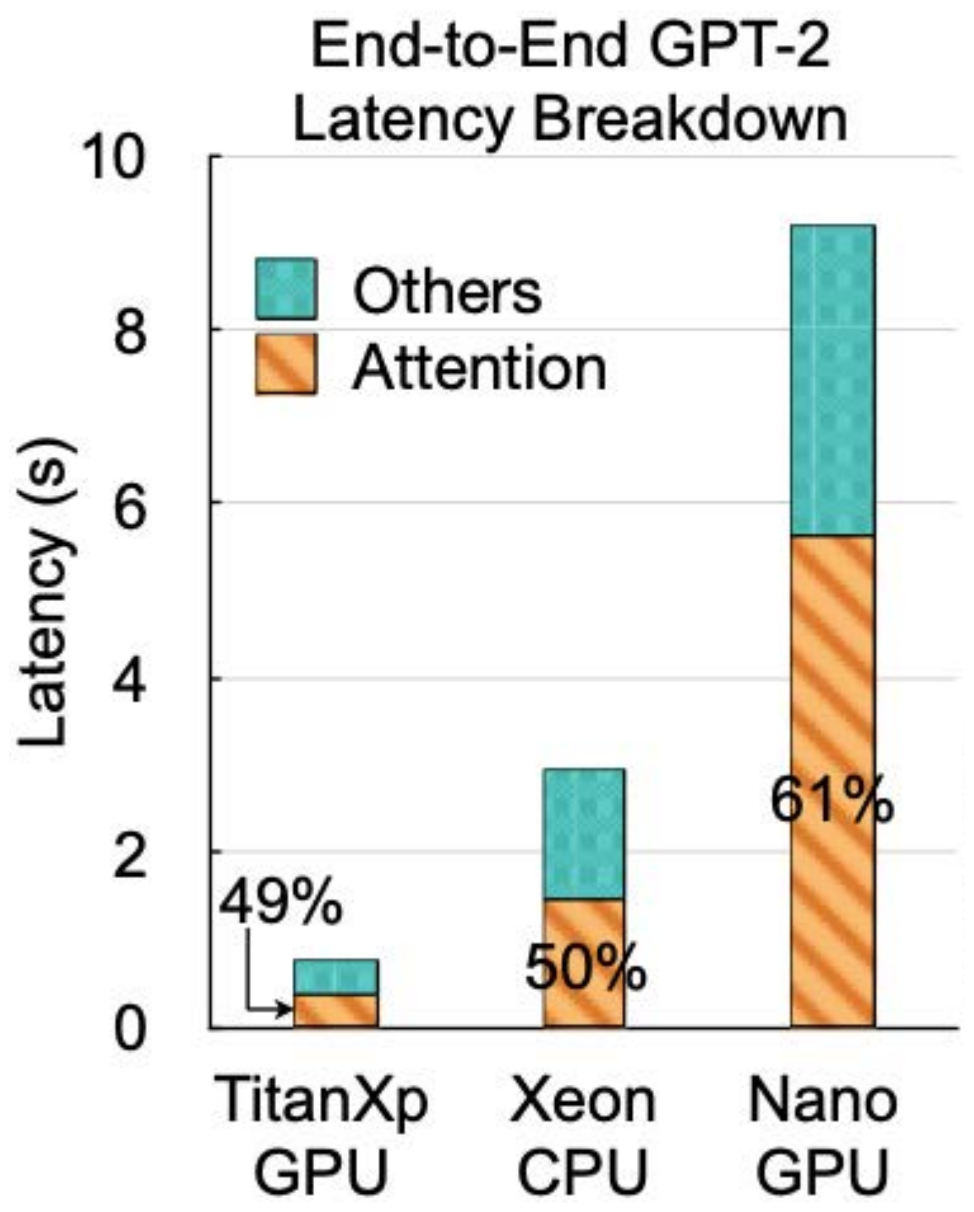
TinyML for NLP

HAT, ACL'20
SpAtten, HPCA'21

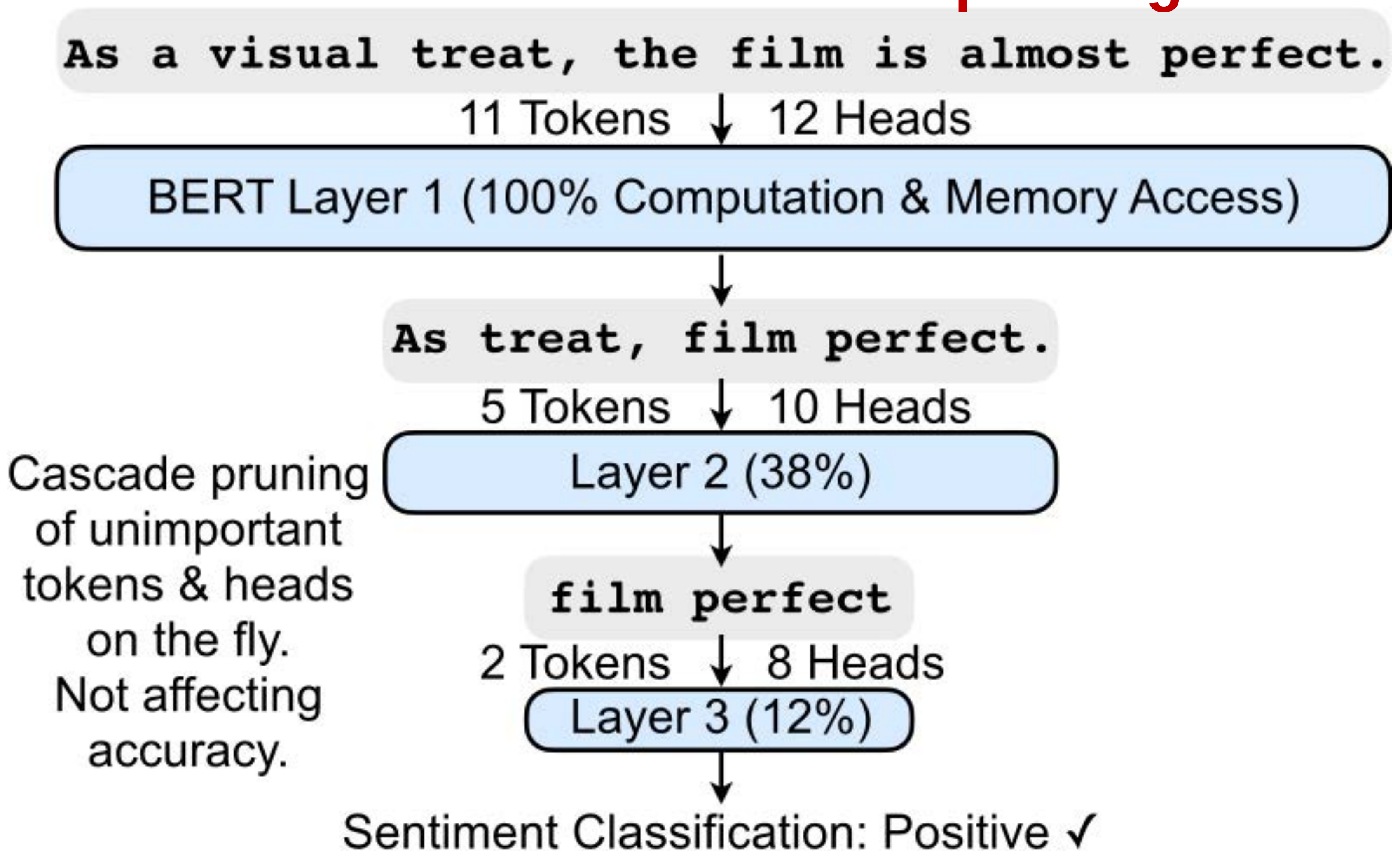
- **Motivation: Attention layer** in natural language processing models is the bottleneck for end-to-end performance.
- **Main idea: reduce redundancy**

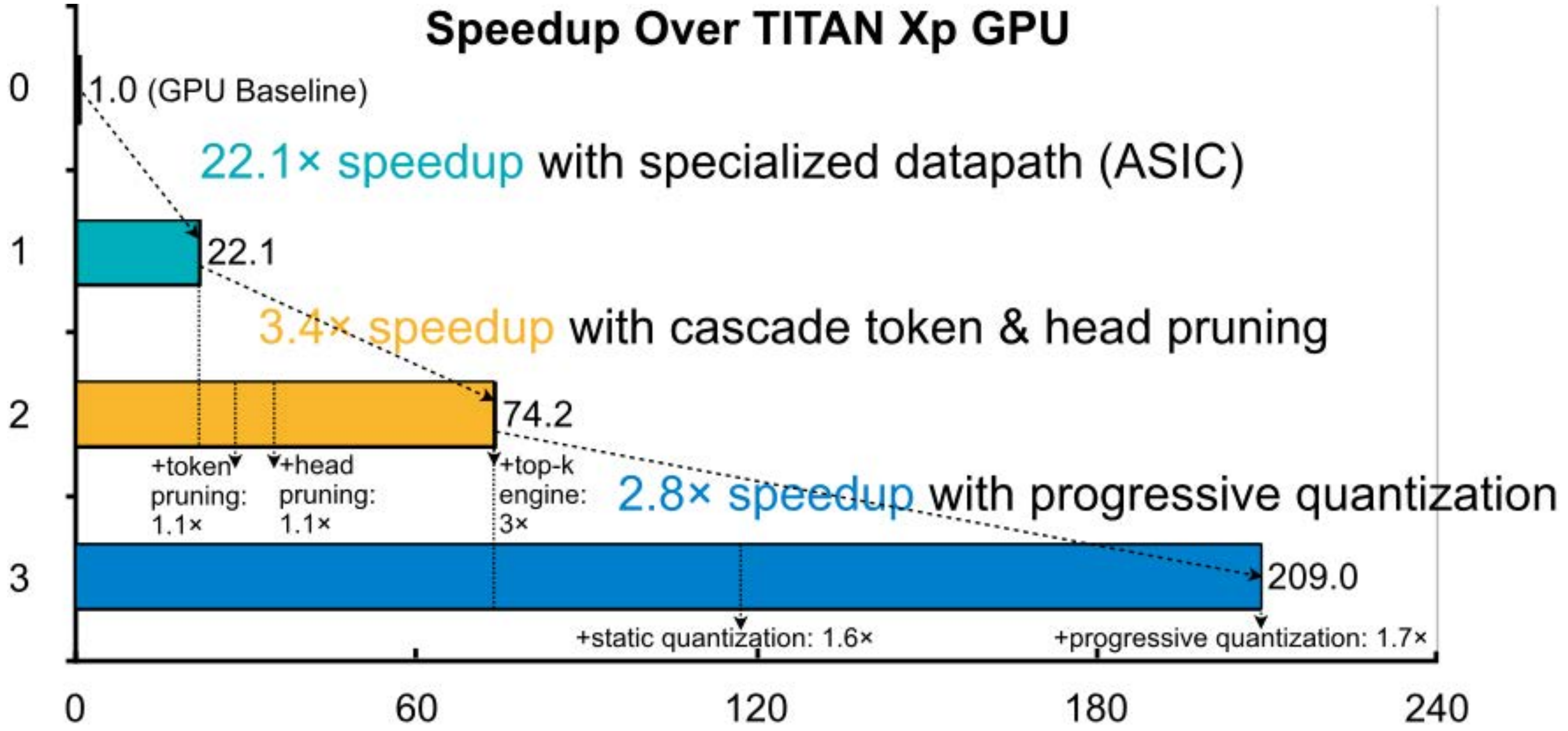
1. **Cascade Token and head pruning**
2. **Progressive quantization:** progressively fetch MSB and LSB

Attention operation is the bottleneck



Cascade token and head pruning





TinyML for Video Recognition

TSM, ICCV 2019

I3D:
Latency: **164.3** ms/Video   Something-V1 Acc.: **41.6**%

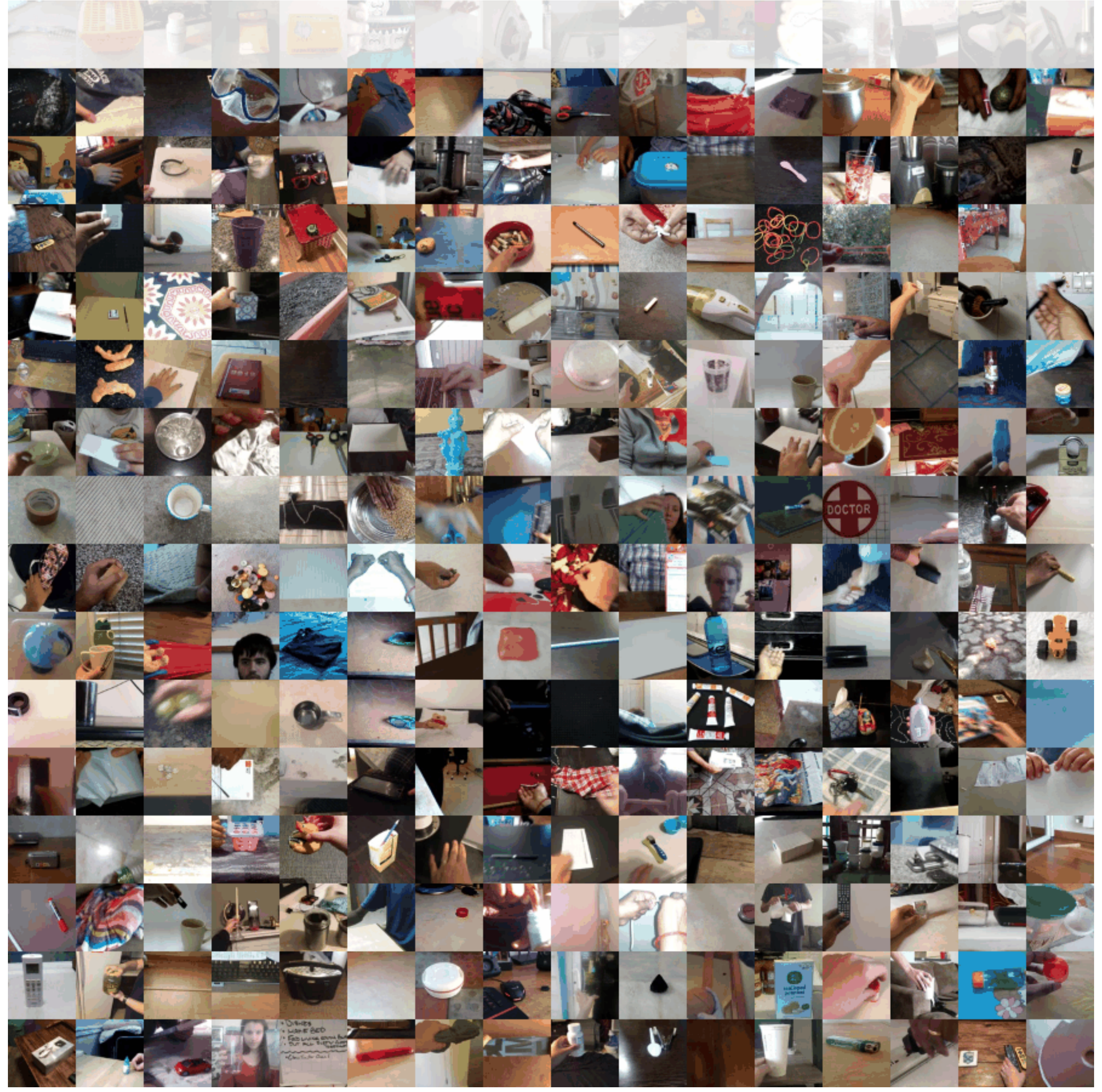
TSM:
Latency: **17.4** ms/Video   Something-V1 Acc.: **43.4**%

Speed-up: 9x

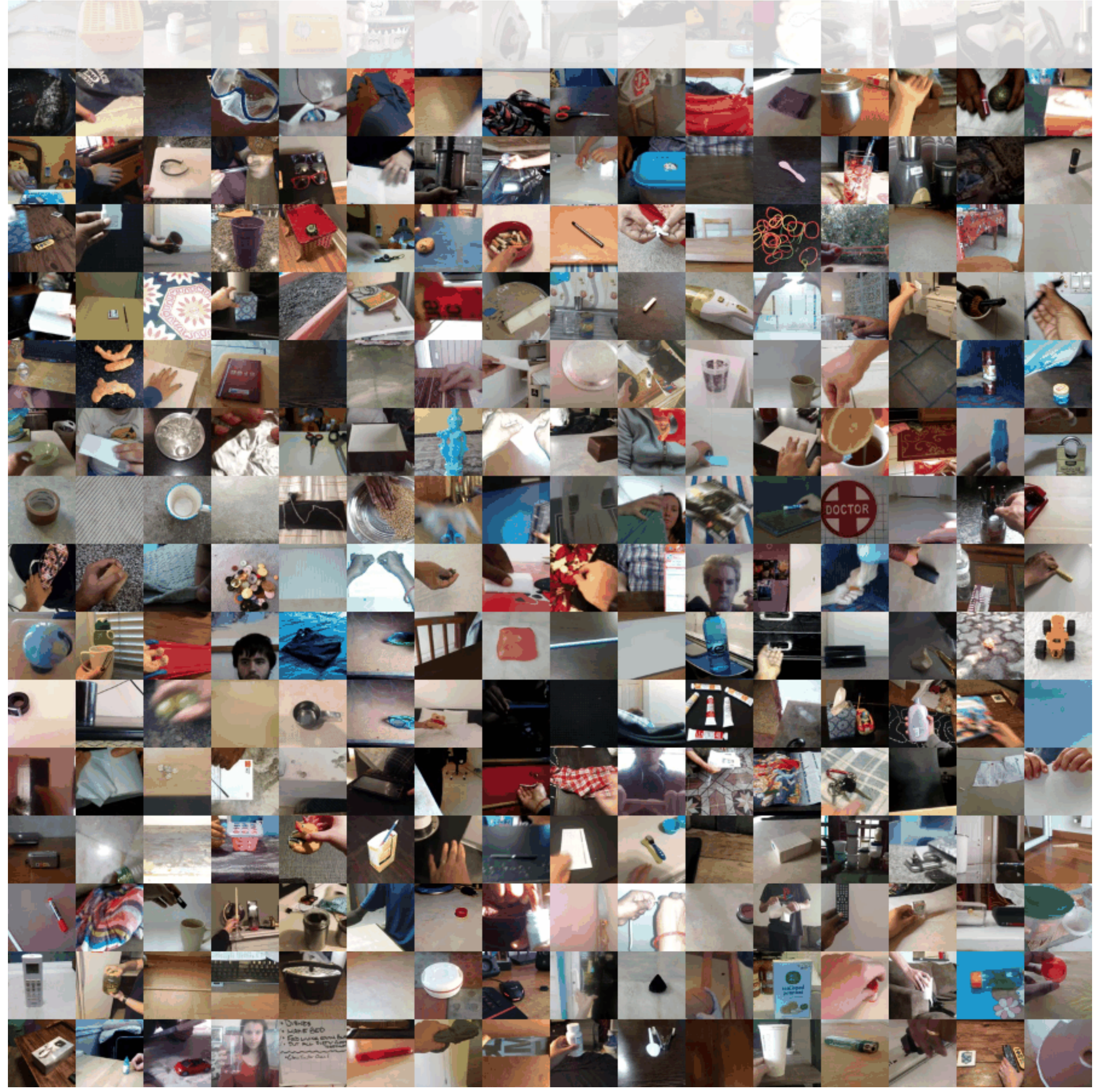
TinyML for Video Recognition

TSM, ICCV 2019

I3D:
Throughput: **6.1** video/s
Something-V1 Acc.: **41.6**%



TSM:
Throughput: **77.4** video/s
Something-V1 Acc.: **43.4**%

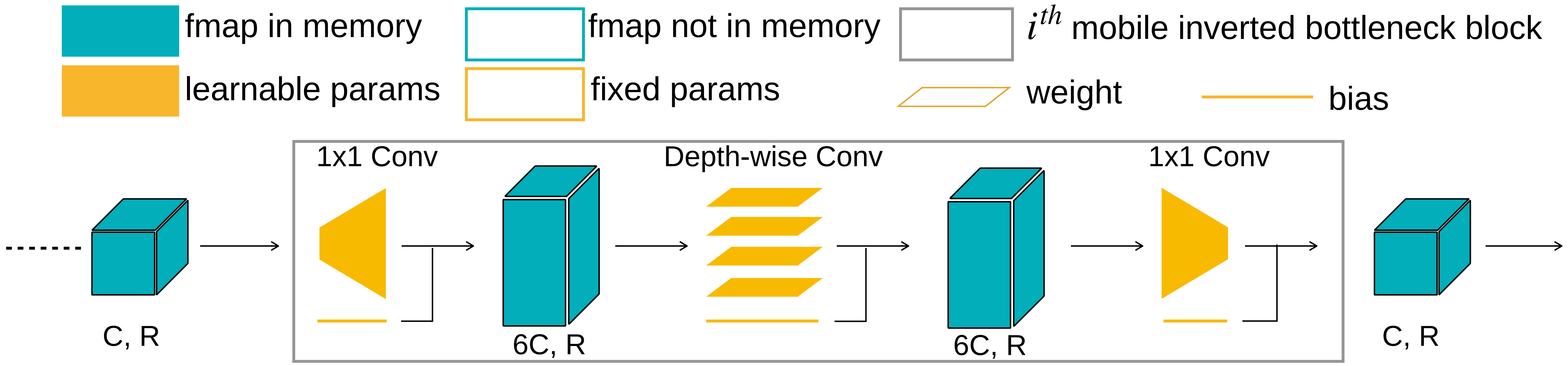


12.7x higher throughput

Tiny Transfer Learning

- Customization: AI systems need to continually adapt to new data collected from the sensors.
- Security: Data cannot leave devices because of security and regularization.
- We can reduce the training memory **from 300MB to 16MB**

Weight update is Memory-expensive; Bias update is Memory-efficient



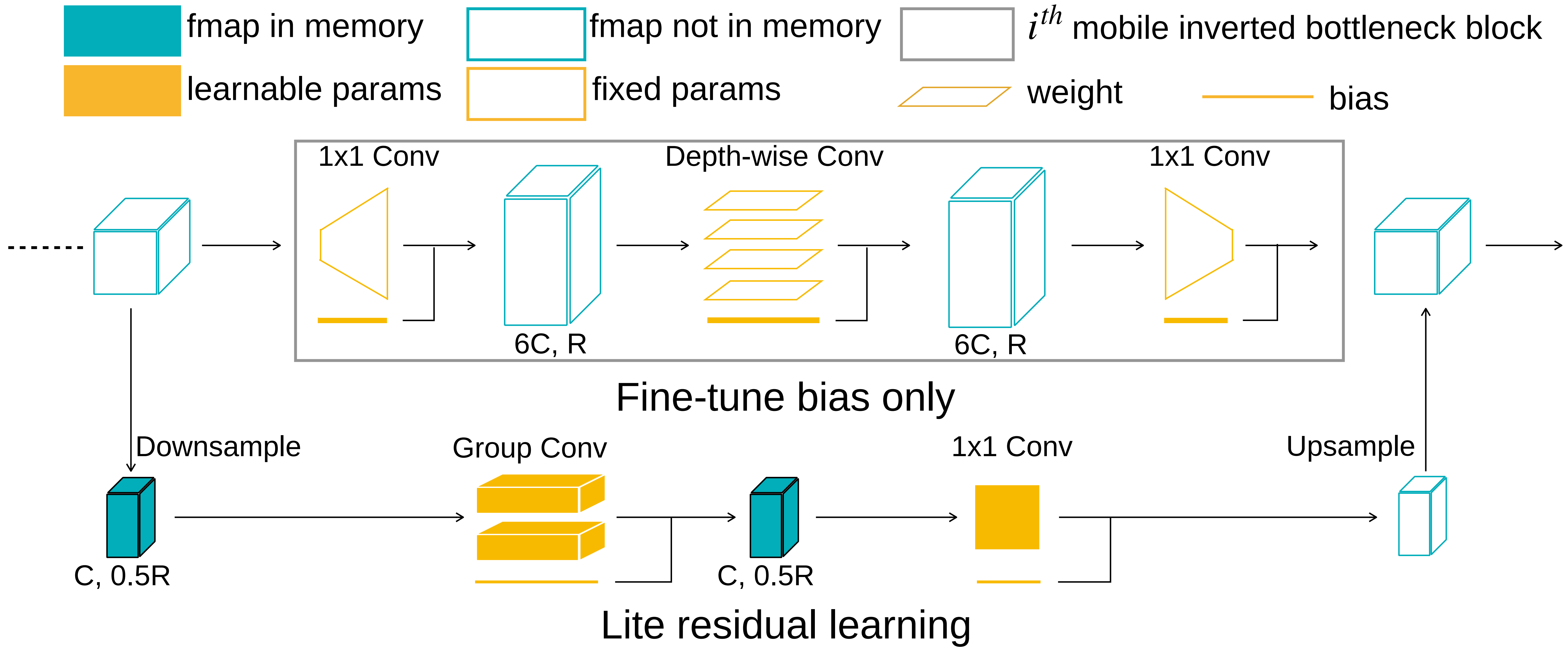
Fine-tune the full network (Conventional)

Forward: $\mathbf{a}_{i+1} = \mathbf{a}_i \mathbf{W}_i + \mathbf{b}_i$

Backward: $\dfrac{\partial L}{\partial \mathbf{W}_i} = \mathbf{a}_i^T \dfrac{\partial L}{\partial \mathbf{a}_{i+1}}, \qquad \dfrac{\partial L}{\partial \mathbf{b}_i} = \dfrac{\partial L}{\partial \mathbf{a}_{i+1}} = \dfrac{\partial L}{\partial \mathbf{a}_{i+2}} \mathbf{W}_{i+1}^T$

- Updating weights requires storing intermediate activations
- Updating biases does not

TinyTL: Lite Residual Learning



- Add lite residual modules (small memory overhead) to increase model capacity
  - (1/6 channel, 1/2 resolution, 2/3 depth)

Data-Efficient GAN

Train GAN with only 100 Images

Without our technique:

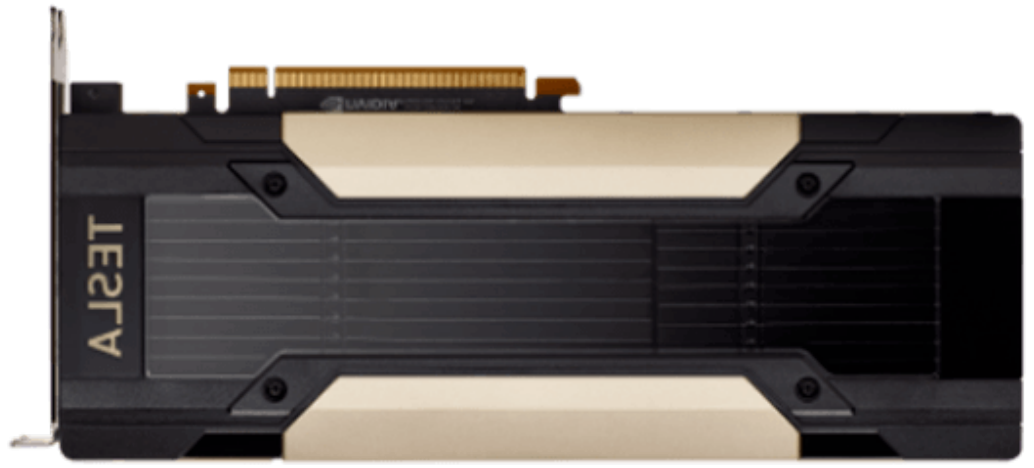
With our technique:

Train GANs with only 100 Images

Smooth interpolation, generalize well

https://github.com/mit-han-lab/data-efficient-gans

Summary: TinyML and Efficient Deep Learning



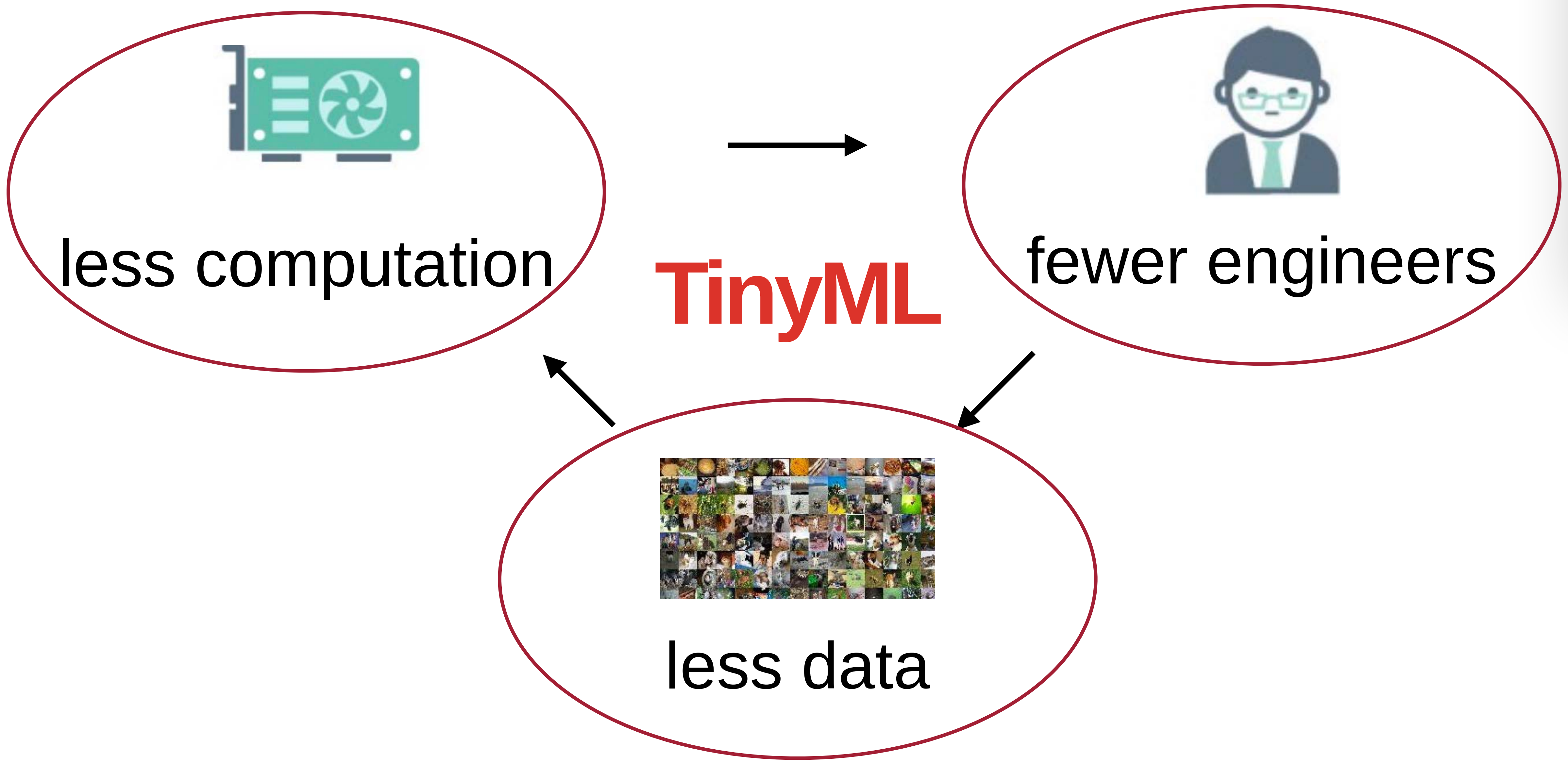
Cloud AI → **Mobile AI** → **Tiny AI**

ResNet | MobileNet | **MCUNet**

Project Page: **http://tinyml.mit.edu**

Hardware, AI and Neural-nets

Make AI Efficient, with **Tiny** Resource



github.com/mit-han-lab

youtube.com/c/MITHANLab

songhan.mit.edu