tinyML® EMEA

Enabling Ultra-low Power Machine Learning at the Edge

tinyML EMEA Technical Forum 2021 Proceedings

June 7 – 10, 2021

Virtual Event

TINY ML

www.tinyML.org

Qualcomm June 8, 2021 @QCOMResearch

Qualcomm

TINY ML

The Model-Efficiency Pipeline

Enabling deep learning inference at the edge

Bert Moons,
Engineer, Senior
Qualcomm Technologies Netherlands B.V.

Qualcomm AI Research

Advancing research to make AI ubiquitous

IoT

Mobile

Automotive

Cloud

Perception
Object detection, speech recognition, contextual fusion

Reasoning
Scene understanding, language understanding, behavior prediction

Action
Reinforcement learning for decision making

Power efficiency | Personalization | Efficient learning

We are creating platform innovations to scale AI across the industry

Qualcomm Research Netherlands

qualcomm.com/careers

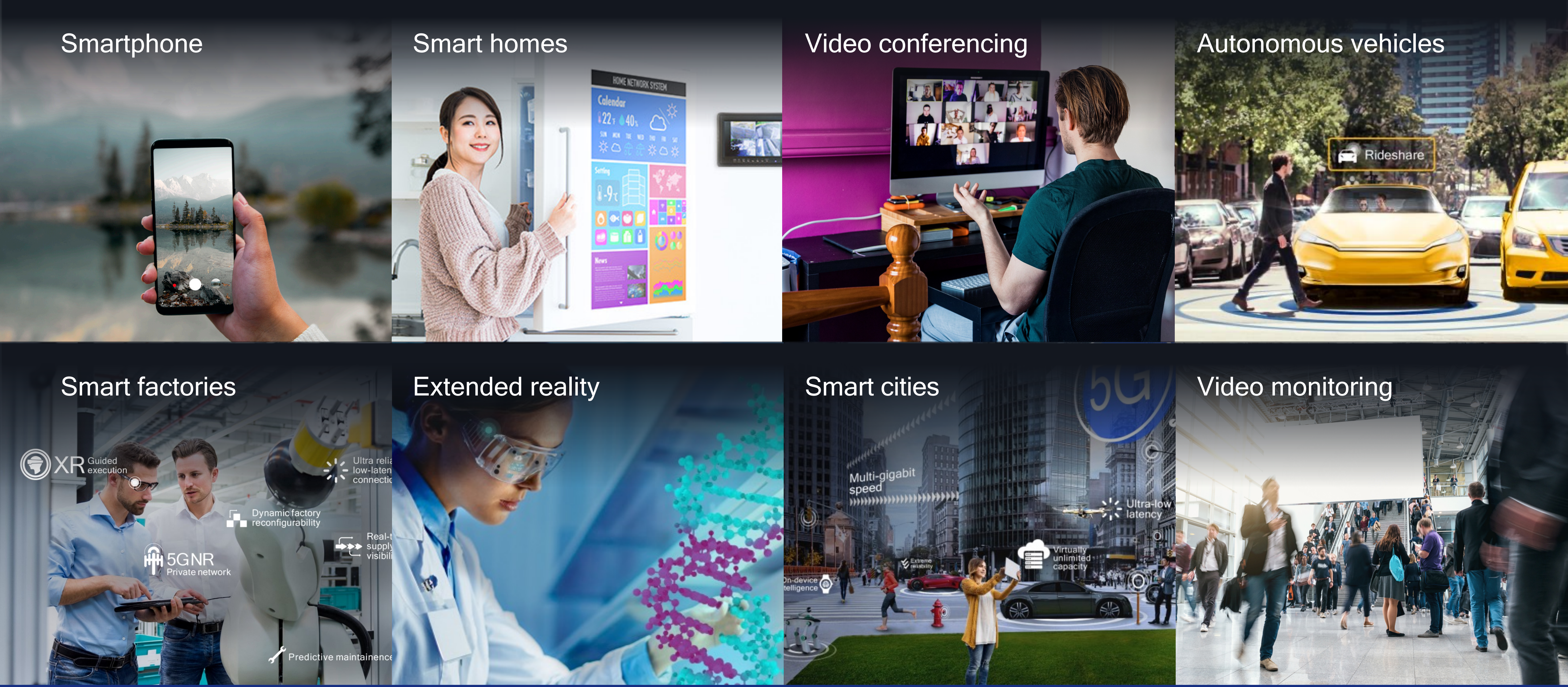
search for Amsterdam

Agenda

- Energy-Efficient machine learning and the computational budget gap
- The Model-Efficiency Pipeline reduces the cost of on-device inference

Qualcomm Innovation Center, Inc. open sources through AI Model Efficiency Toolkit (AIMET)

- What’s next in energy-efficient AI



AI is being used all around us

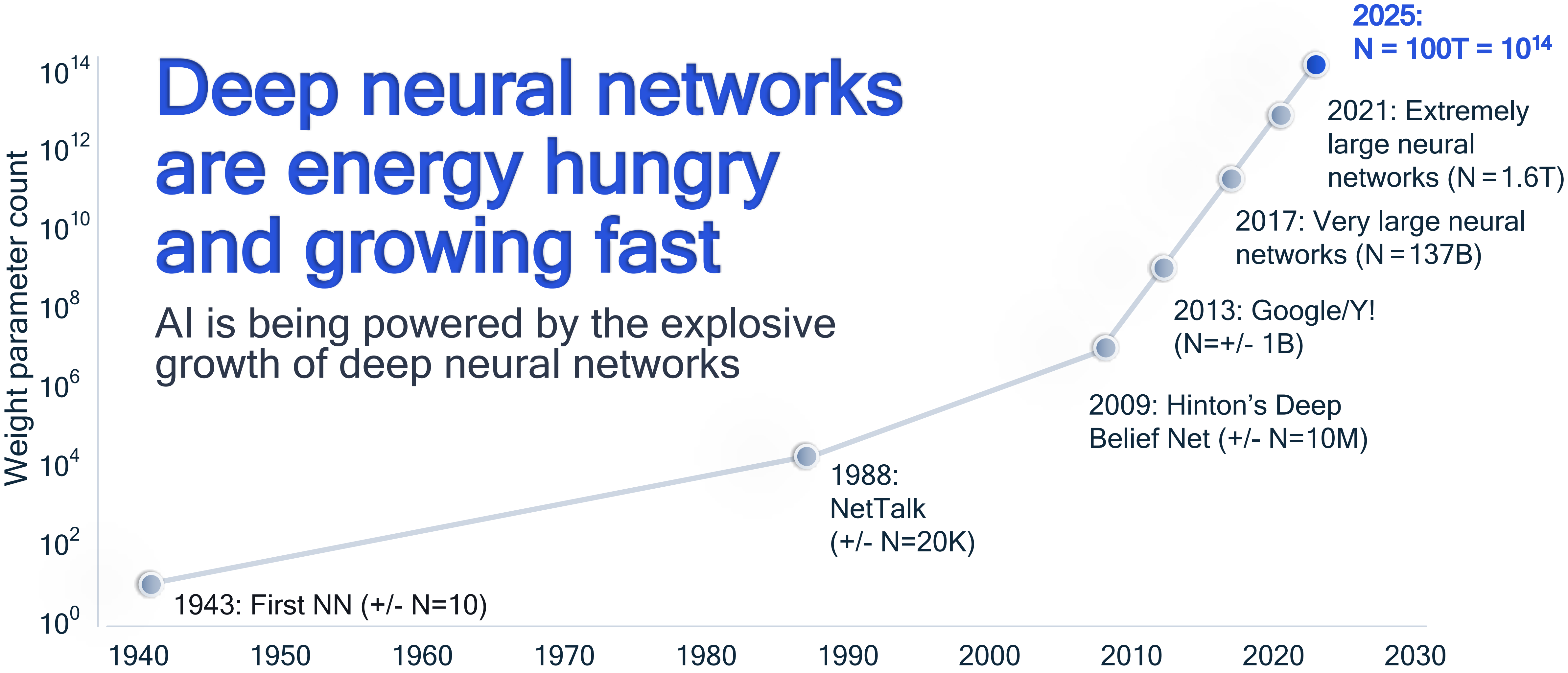
increasing productivity, enhancing collaboration, and transforming industries

AI video analysis is on the rise

Trend toward more cameras, higher resolution, and increased frame rate across devices

Deep neural networks are energy hungry and growing fast

AI is being powered by the explosive growth of deep neural networks

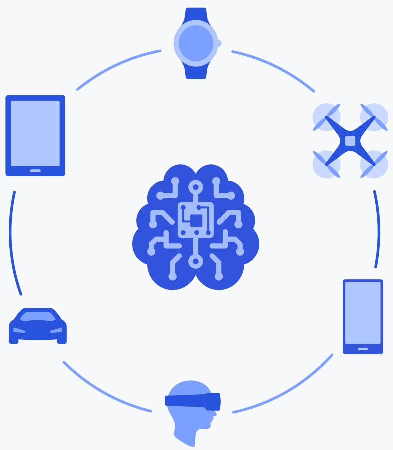


2025 | Increasingly large and complex neural networks for Natural Language Processing, Image and Video Processing

Source: Welling

The challenge of AI workloads

- Very compute intensive
- Large, complicated neural network models
- Complex concurrencies
- Real-time
- Always-on



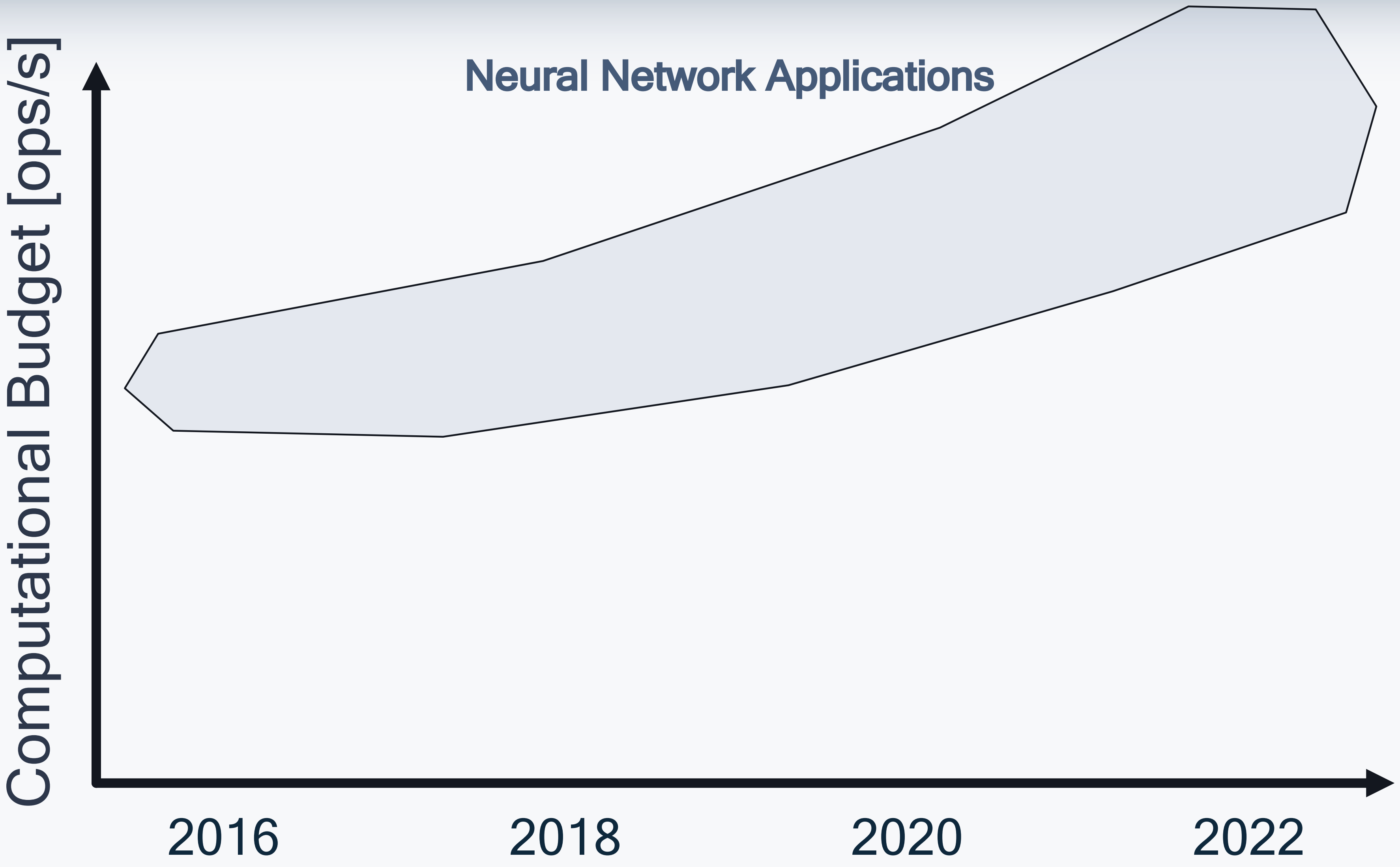
Power and thermal efficiency are essential for on-device AI

Constrained mobile environment

- Must be thermally efficient for sleek, ultra-light designs
- Requires long battery life for all-day use
- Storage/memory bandwidth limitations

The Deep Learning Budget Gap

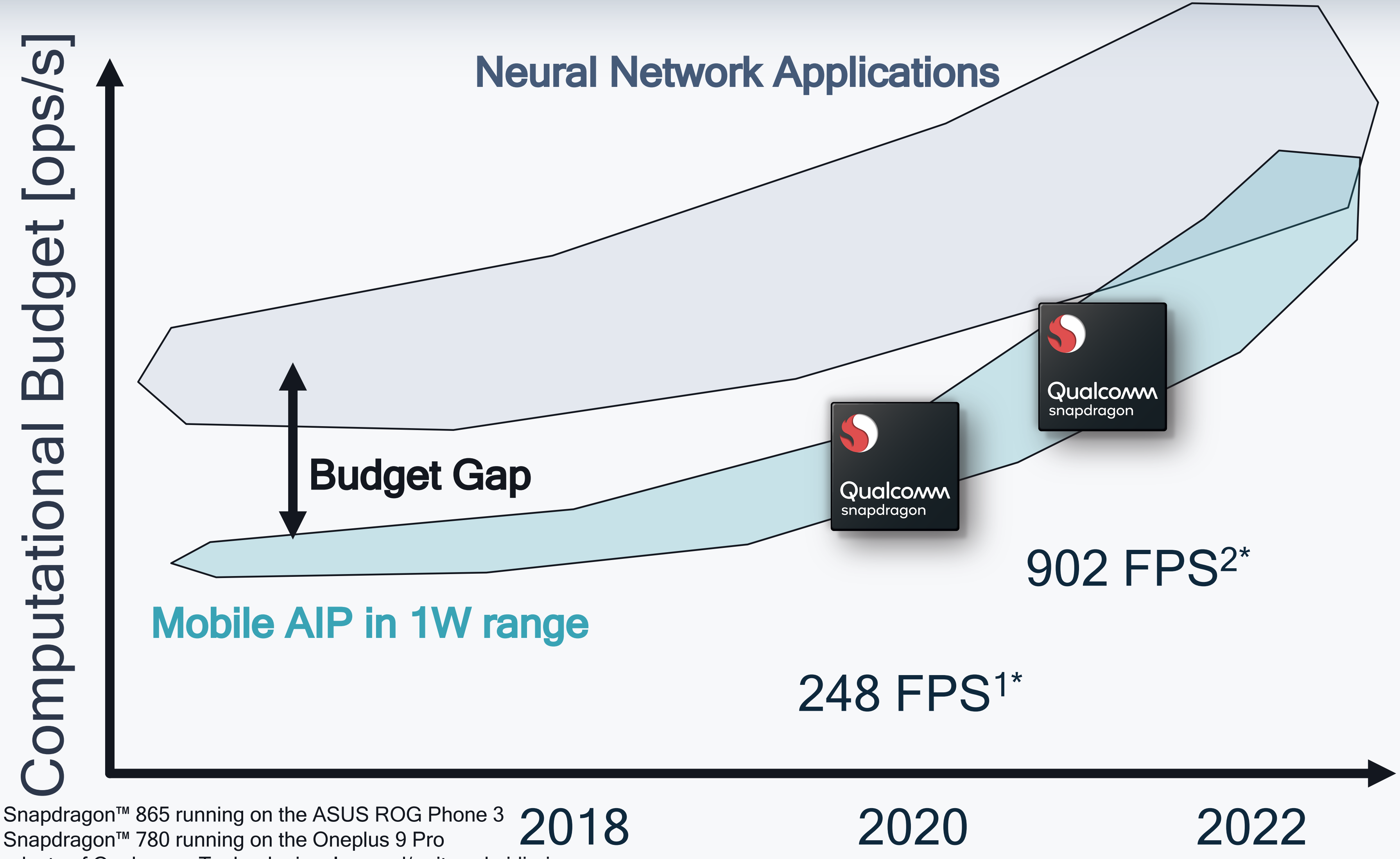
<u>Trend 1:</u>
Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, ...



The Deep Learning Budget Gap

<u>Trend 1:</u>
Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, …

<u>Trend 2:</u>
Faster, more efficient <u>hardware platforms</u> close the Budget Gap

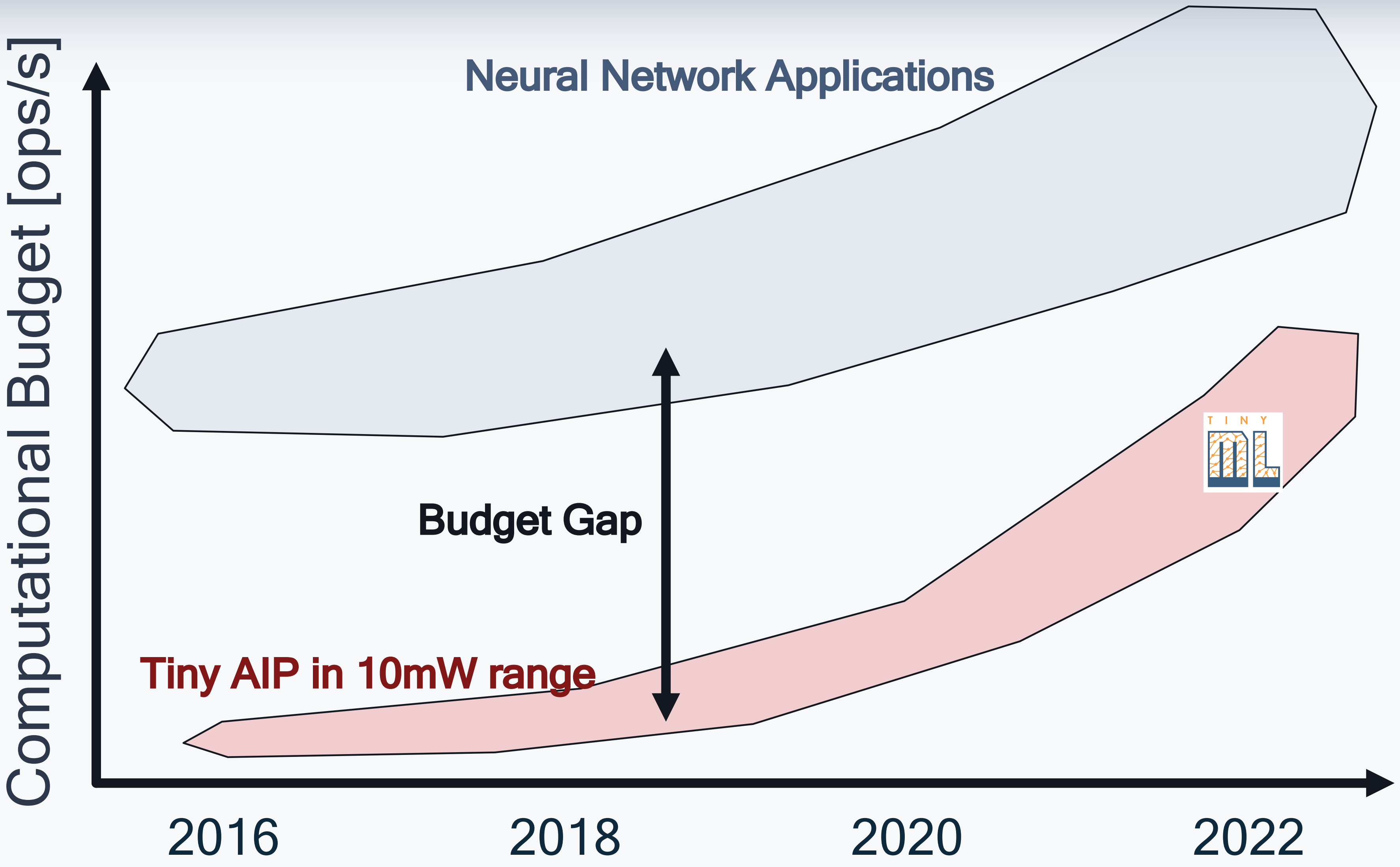


1: Qualcomm® Hexagon™ 698 DSP in the Qualcomm® Snapdragon™ 865 running on the ASUS ROG Phone 3
2: Qualcomm® Hexagon™ 780 DSP in the Qualcomm® Snapdragon™ 780 running on the Oneplus 9 Pro
Qualcomm Hexagon and Qualcomm Snapdragon are products of Qualcomm Technologies, Inc. and/or its subsidiaries.
*: MobileNetEdge on mlperf https://mlcommons.org/en/inference-mobile-10/

The Deep Learning Budget Gap

<u>Trend 1:</u>
Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, …

<u>Trend 2:</u>
Faster, more efficient <u>hardware platforms</u> close the Budget Gap

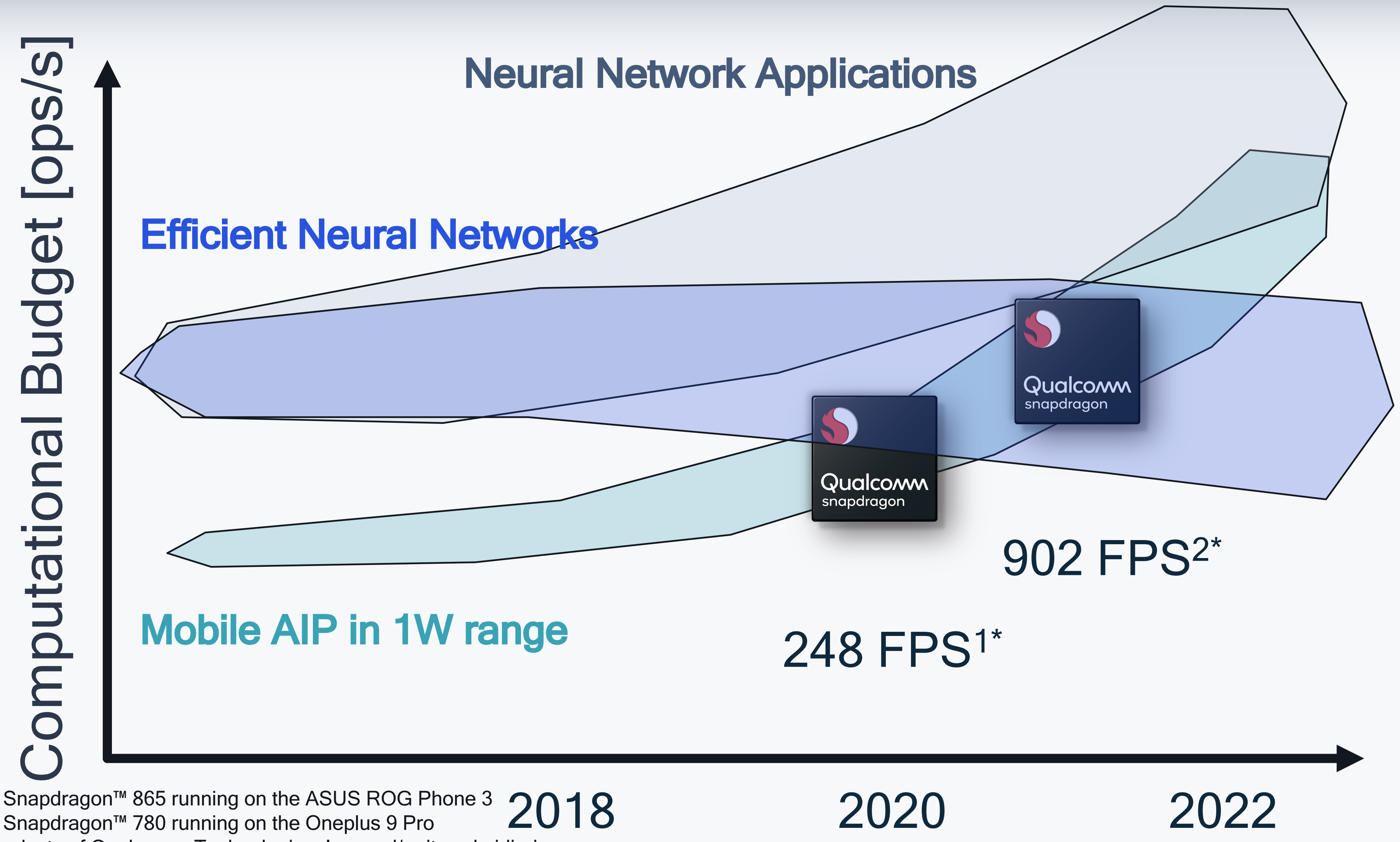


The Deep Learning Budget Gap

Trend 1:
Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, …

Trend 2:
Faster, more efficient hardware platforms close the Budget Gap

Trend 3:
Faster, optimized Neural Networks and Applications close the Budget Gap



1: Qualcomm® Hexagon™ 698 DSP in the Qualcomm® Snapdragon™ 865 running on the ASUS ROG Phone 3
2: Qualcomm® Hexagon™ 780 DSP in the Qualcomm® Snapdragon™ 780 running on the Oneplus 9 Pro
Qualcomm Hexagon and Qualcomm Snapdragon are products of Qualcomm Technologies, Inc. and/or its subsidiaries.
*: MobileNetEdge on mlperf https://mlcommons.org/en/inference-mobile-10/

Trend 3: The Model-Efficiency Pipeline

The Model-Efficiency Pipeline

Multiple axes to shrink AI models and run them efficiently on hardware

Neural architecture search

Pruning and Model Compression

Accurate Quantization

Neural Architecture Search: automated design of on-device optimal networks

Training networks from scratch is expensive!

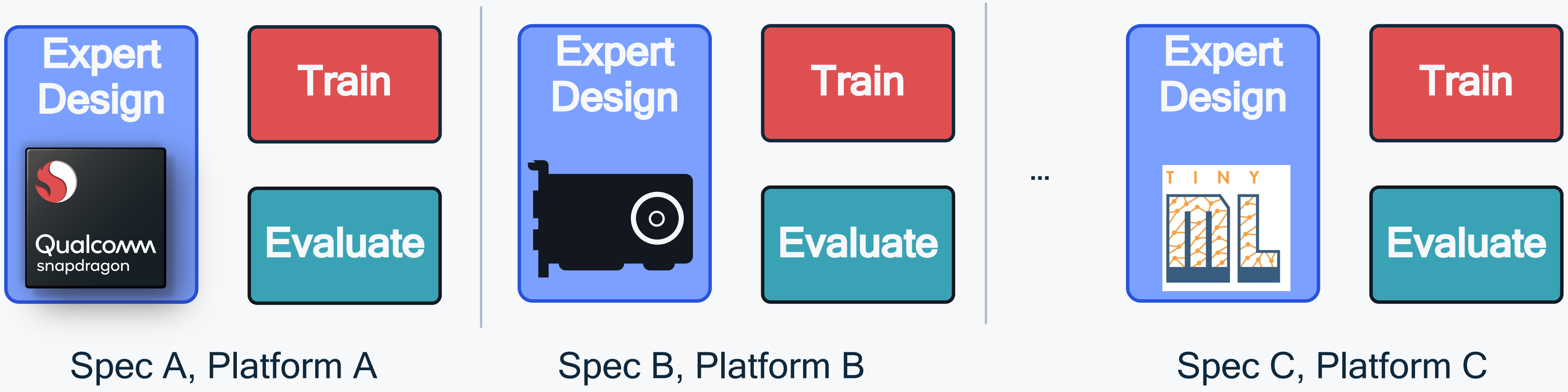
>2 GPU months to train a single SotA network on ImageNet
~4k USD per network using commercial cloud services

Neural Architecture Search: automated design of on-device optimal networks

Training networks from scratch is expensive!

Manual network design requires training many networks from scratch for every device

>2 GPU months to train a single SotA network on ImageNet
~4k USD per network using commercial cloud services



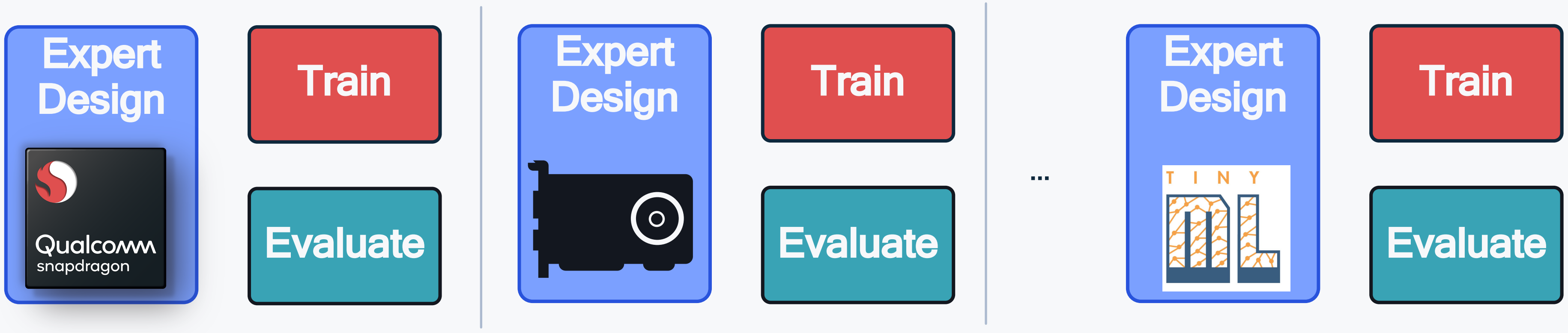
Neural Architecture Search: automated design of on-device optimal networks

Training networks from scratch is expensive!

Manual network design requires training many networks from scratch for every device

Solution

>2 GPU months to train a single SotA network on ImageNet
~4k USD per network using commercial cloud services



Spec A, Platform A

Spec B, Platform B

Spec C, Platform C

Cheap, scalable Neural Architecture Search reduces design and training costs of networks optimized for specific devices

Existing NAS solutions do not address all the challenges

Lack diverse search

Hard to search in diverse spaces, with different block-types, attention, and activations

Repeated training for every new scenario

High cost

Brute force search is expensive

>40,000 epochs per platform

Do not scale

Repeated training for every device

>40,000 epochs per platform

Unreliable hardware models

Requires differentiable cost-functions

Repeated training phase for every new device

Introducing new AI research

DONNA

Distilling Optimal Neural Network Architectures

Efficient NAS with hardware-aware optimization

Finds pareto-optimal architectures in terms of accuracy-latency at low cost

Diverse search to find the best models

Supports diverse spaces with different cell-types, attention, and activation functions (ReLU, Swish, etc.)

Low cost

Low start-up cost equivalent to training 2-10 networks from scratch

Scalable

Scales to many hardware devices at minimal cost

Reliable hardware measurements

Uses direct hardware measurements instead of a potentially inaccurate hardware model

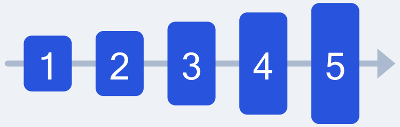
Distilling Optimal Neural Networks: Rapid Search in Diverse Spaces (Moons, Bert, et al., arXiv 2020)

DONNA 4-step process Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “”Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

A Define reference and search space once

Define backbone:
- Fixed channels
- Head and Stem



Varying parameters:
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types

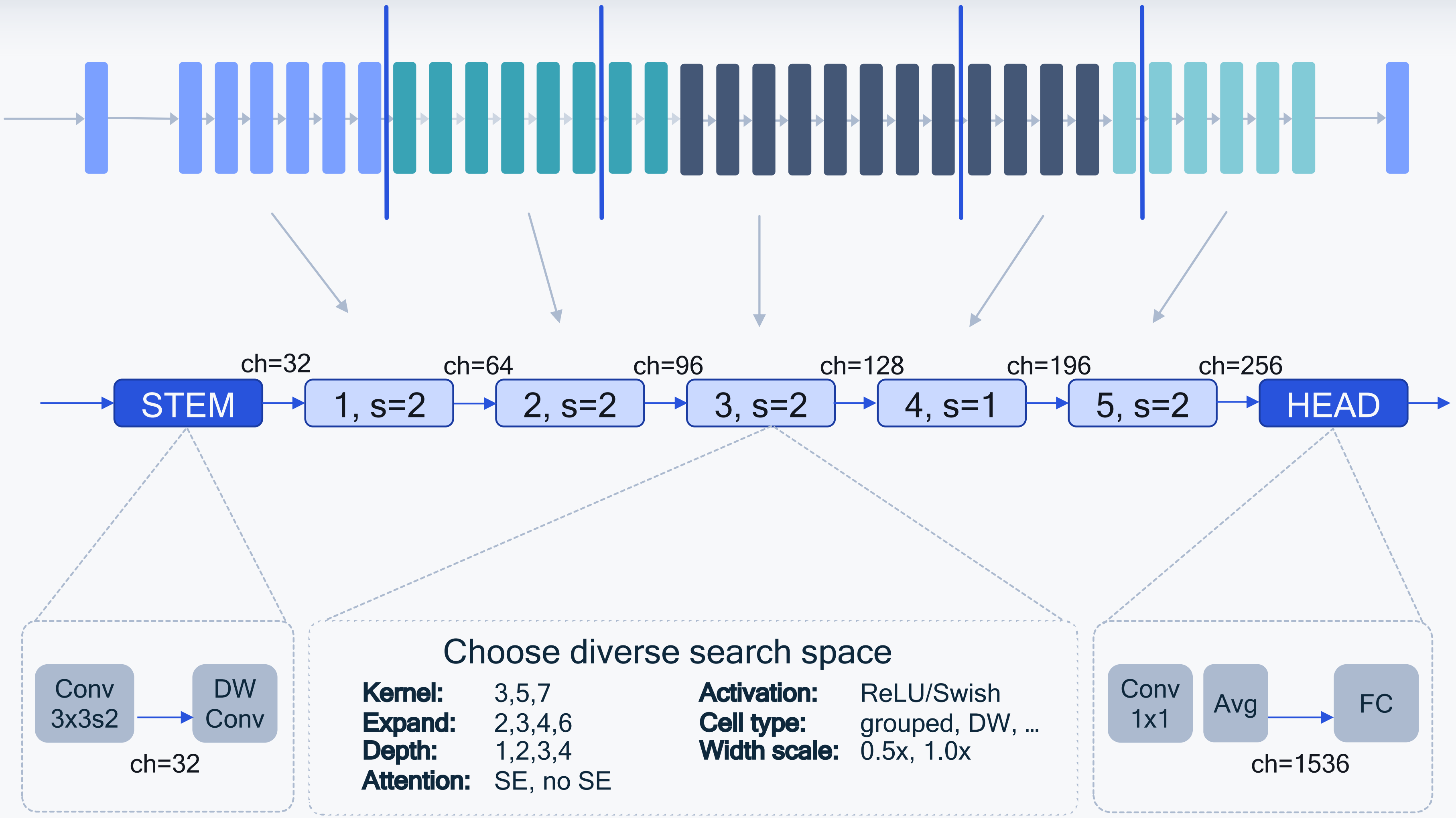
Define reference architecture and search-space once

A diverse search space is essential for finding optimal architectures with higher accuracy

Select reference architecture
The largest model in the search-space

Chop the NN into blocks
Fix the STEM, HEAD, # blocks, strides, # channels at block-edge

Choose search space
Diverse factorized hierarchical search space, including variable cell-types, kernel-size, expansion-rate, depth, # channels, activation, attention



Ch: channel; SE: Squeeze-and-Excitation

Define reference architecture and search-space once

Some example blocks in the shared search space: BasicBlocks, ShiftNets, MobileConv, Squeeze-and-Excitation

Choose search space

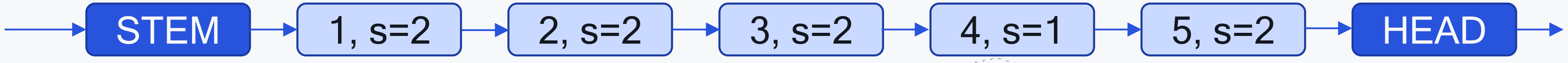
Examples of variable cell types that can be combined in a single search space

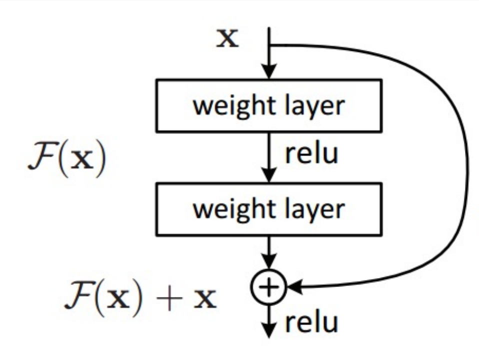
[1] K. He, "Deep Residual Learning for image recognition", CVPR16

[2] M. Sandler, "MobileNEtV2: Inverted Residuals and Linear Bottlenecks", CVPR18"

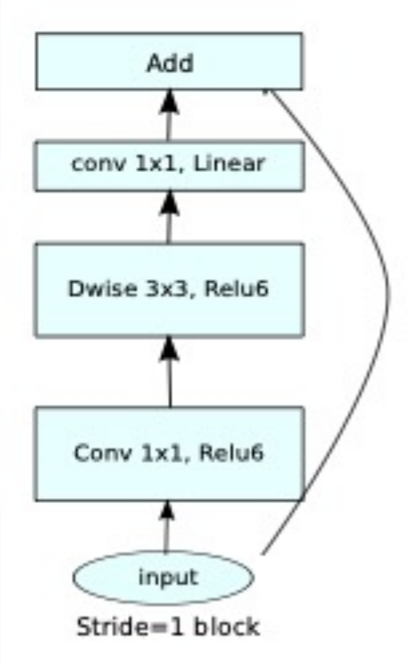
[3] A. Dosovitskiy, "An image is Worth 16x16 words: transformers for image recognition at scale", ICLR21

[4] W. Chen, "All you need is a few shifts: Designing efficient convolutional neural networks for image classification", CVPR19

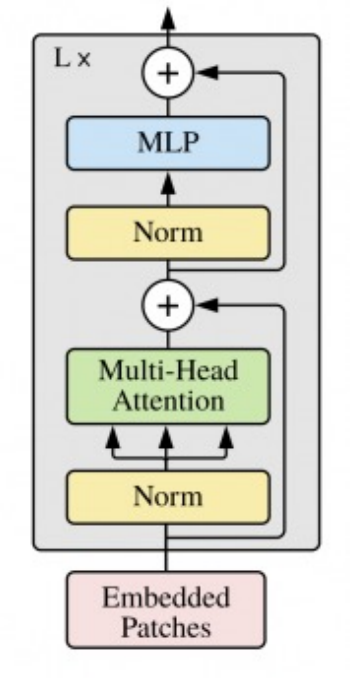




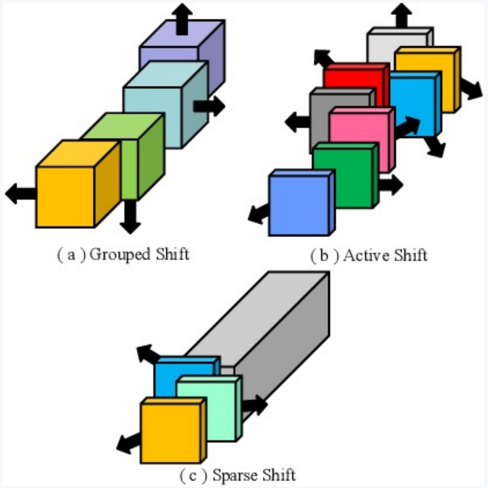
ResNet-Style BasicBlock [1]



MobileNet-Style Inverted Bottleneck [2]



Vision-Transformer [3]



ShiftNets [4]

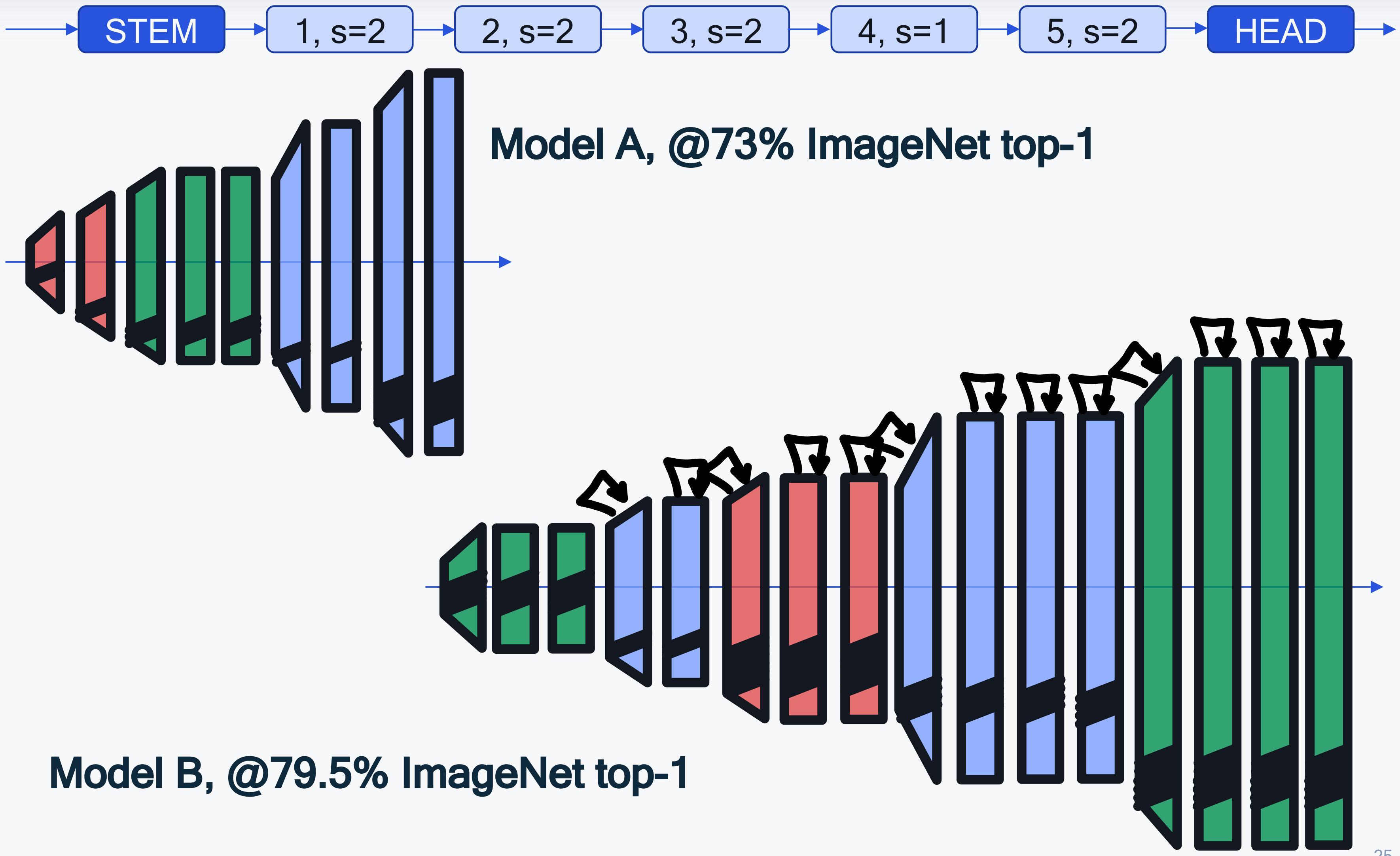
Ch: channel; SE: Squeeze-and-Excitation

Define reference architecture and search-space once

Some example blocks in the shared search space: BasicBlocks, ShiftNets, MobileConv, Squeeze-and-Excitation

Choose search space

Two example models, pareto-optimal on a desktop GPU



Distilling Optimal Neural Networks: Rapid Search in Diverse Spaces (Moons, Bert, et al., arXiv 2020)

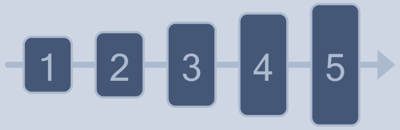
DONNA 4-step process

Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “”Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

A Define reference and search space once

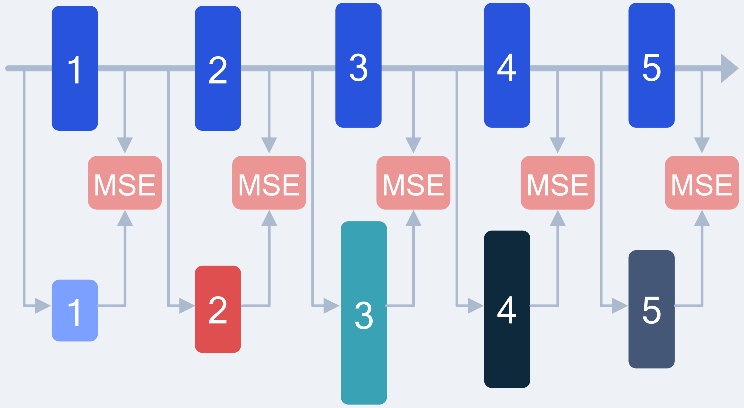
Define backbone:
- Fixed channels
- Head and Stem



Varying parameters:
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types

B Build accuracy model via Knowledge Distillation (KD) once

Approximate ideal projections of a reference model through KD



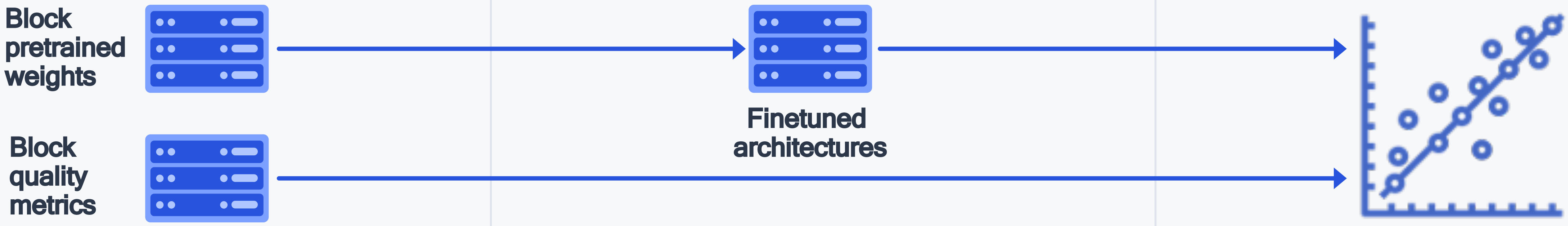
Use quality of blockwise approximations to build accuracy model

Build accuracy predictor via Blockwise Knowledge Distillation once

Low-cost hardware-agnostic training phase

Block library

Pretrain all blocks in search-space through blockwise knowledge distillation



- Fast block training
- Trivial parallelized training
- Broad search space

Architecture library

Quickly finetune a representative set of architectures

- Finetune sampled networks
- Fast network training
- Only 20-30 NN required

Accuracy predictor

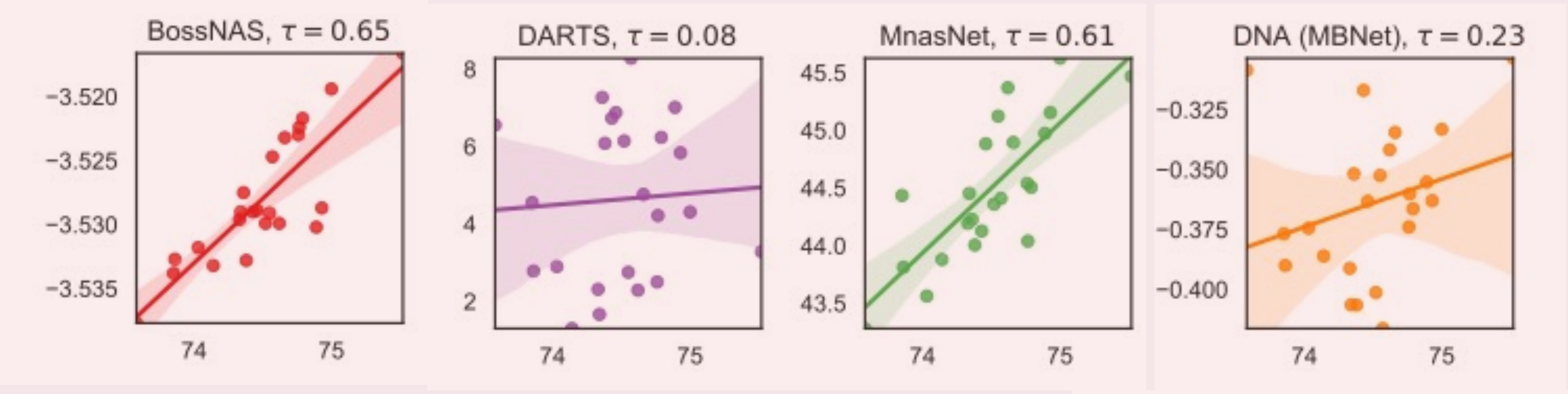
Fit linear regression model

- Linear Regression Model
- Accurate predictions
- Up to 10x improved ranking vs DARTS

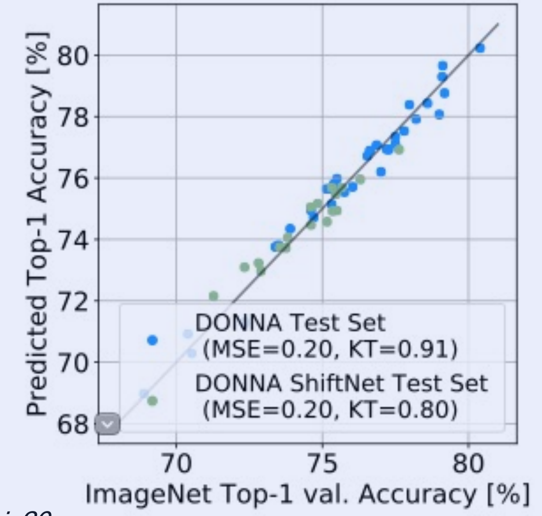
Build accuracy predictor via BKD once

Low-cost hardware-agnostic training phase

State-of-the-art references achieve up to 0.65 KT ranking*



DONNA achieves up to 0.91 KT on basic test sets and reliably extends to test sets with previously unseen cell-types: 0.8KT



Accuracy predictor

Fit linear regression model

Linear Regression Model

Accurate predictions

Up to 10x improved ranking vs DARTS

DONNA = Bert Moons et al, “”Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

** Changlin Li, et al, “BossNAS: Exploring Hybrid CNN-transformers with Block-wisely Self-supervised Neural Architecture Search, Arxiv 2021*

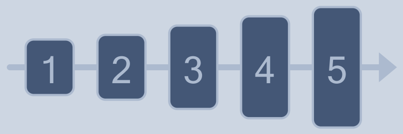
DONNA 4-step process

Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “”Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

A Define reference and search space once

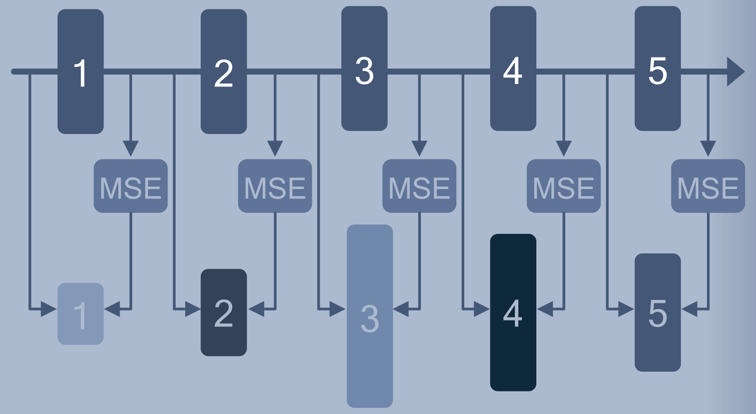
Define backbone:
- Fixed channels
- Head and Stem



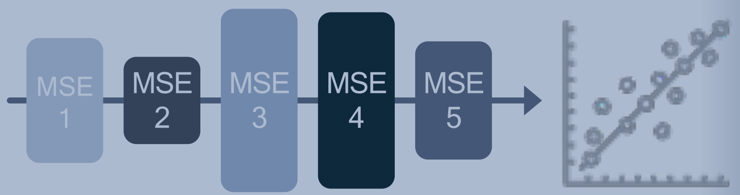
Varying parameters:
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types

B Build accuracy model via Knowledge Distillation (KD) once

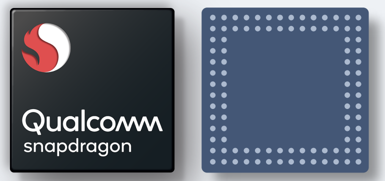
Approximate ideal projections of a reference model through KD



Use quality of blockwise approximations to build accuracy model



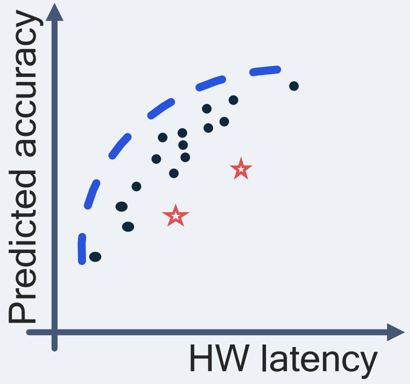
C Evolutionary search in 24h



+

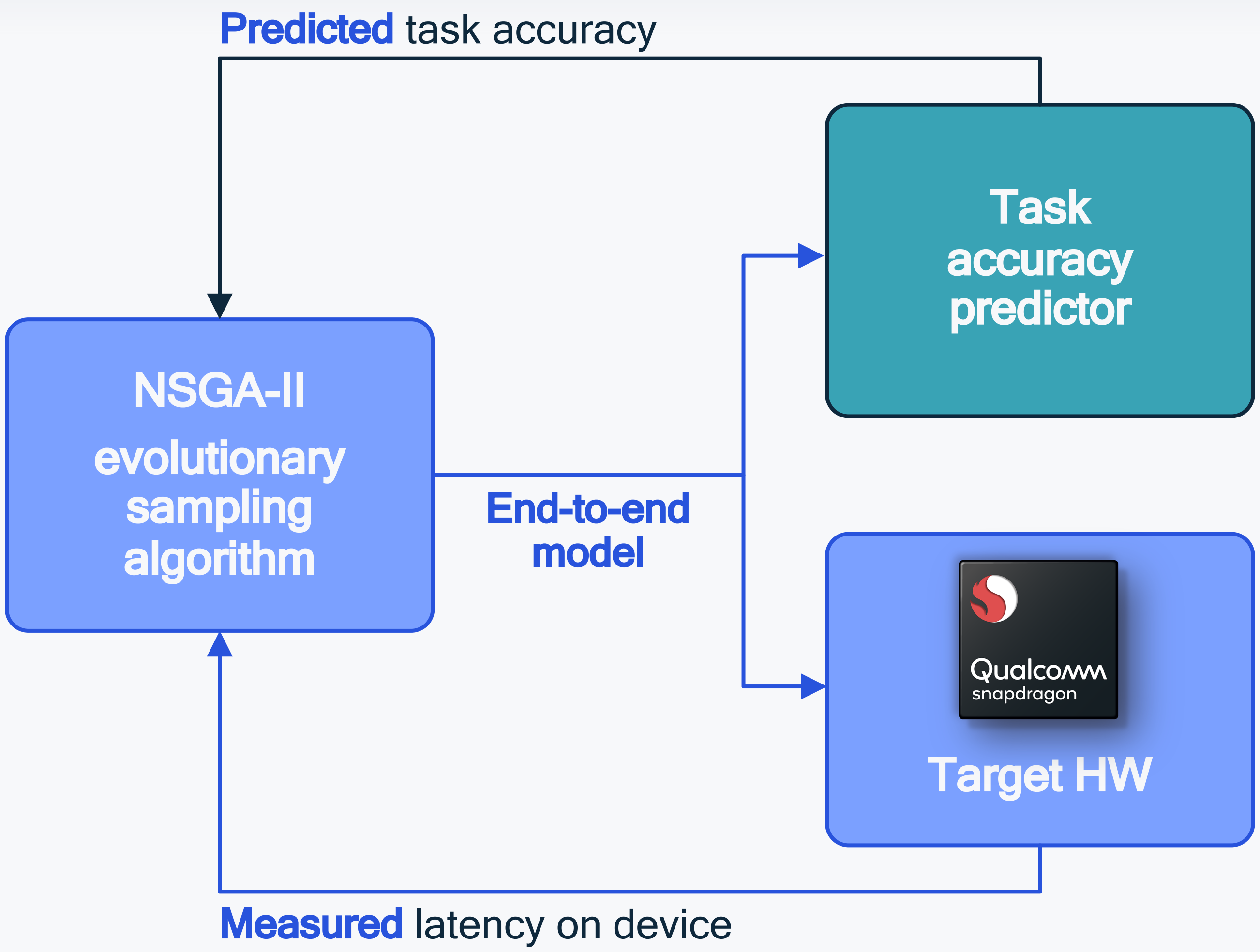
different compiler versions, different image sizes

Scenario-specific search



Evolutionary search with real hardware measurements

Scenario-specific search allows users to select optimal architectures for real-life deployments



Quick turnaround time

Results in +/- 1 day using one measurement device

Accurate scenario-specific search

Captures all intricacies of the hardware platform and software – e.g. run-time version or devices

NSGA: Non-dominated Sorting Genetic Algorithm

DONNA 4-step process

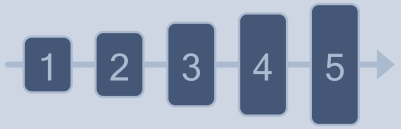
Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “”Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

A Define reference and search space once

Define backbone:

- Fixed channels
- Head and Stem

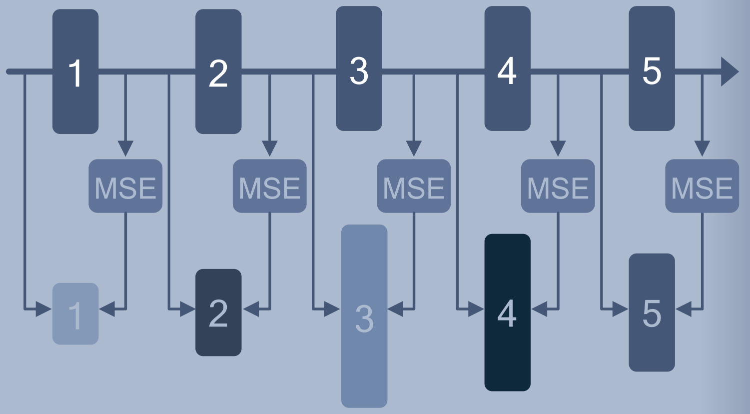


Varying parameters:

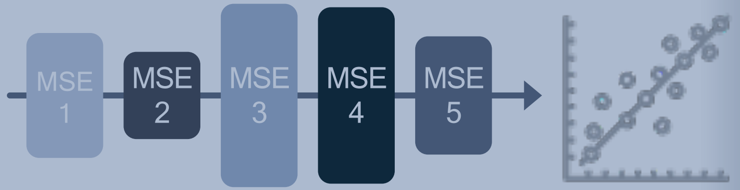
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types

B Build accuracy model via Knowledge Distillation (KD) once

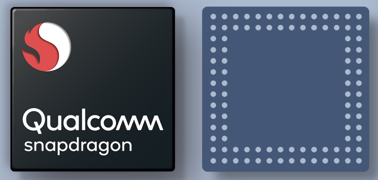
Approximate ideal projections of a reference model through KD



Use quality of blockwise approximations to build accuracy model

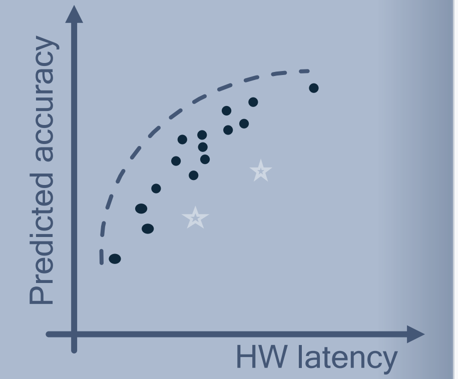


C Evolutionary search in 24h

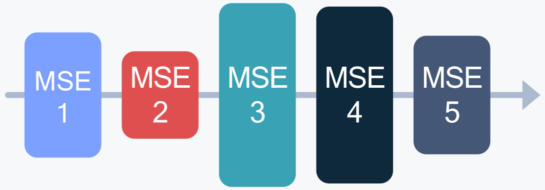


different compiler versions, different image sizes

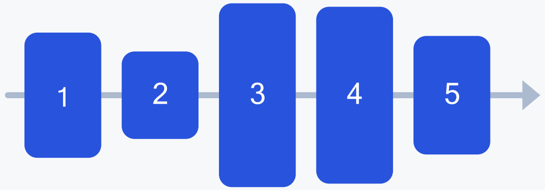
Scenario-specific search



D Sample and finetune

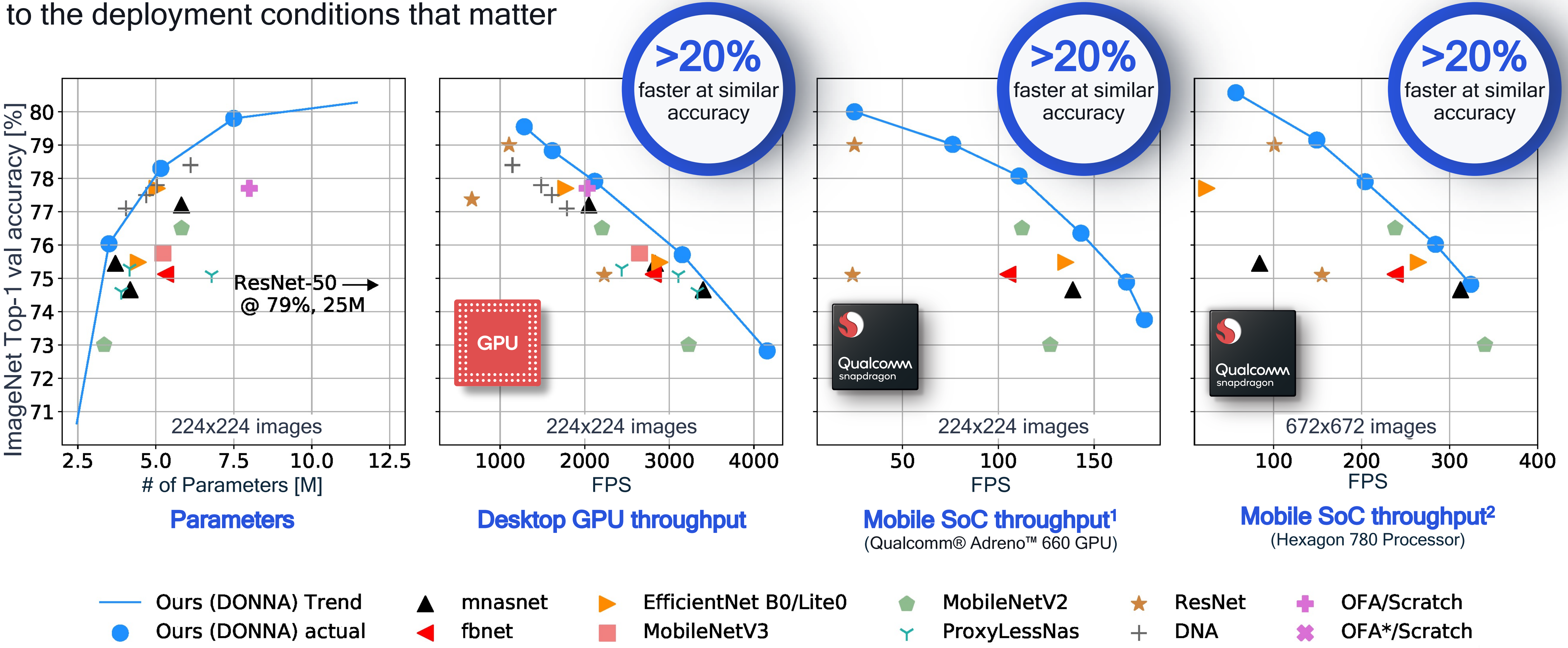


Use KD-initialized blocks from step B to finetune any network in the search space in **15-50 epochs instead of 450**



DONNA finds state-of-the-art networks for on-device scenarios

Quickly optimize and make tradeoffs in model accuracy with respect to the deployment conditions that matter



Parameters | **Desktop GPU throughput** | **Mobile SoC throughput[1]** (Qualcomm® Adreno™ 660 GPU) | **Mobile SoC throughput[2]** (Hexagon 780 Processor)

Qualcomm Adreno 660 GPU in the Snapdragon 888 running on the Samsung Galaxy S21. 2: Qualcomm Hexagon 780 Processor in the Snapdragon 888 running on the Samsung Galaxy S21.

Fs_e: From-Scratch-Equivalent training cost, 450 epochs

DONNA efficiently finds optimal models over diverse scenarios

Cost of training is a handful of architectures*

Method	Granularity	Macro-diversity	Search-cost / scenario 1 scenario, 10 models/scenario [FS_e]	Search-cost / scenario ∞ scenarios, 10 models/ scenario [FS_e]
OFA	Layer-level	Fixed	2.7+**10**×[0.05-0.15]	0.5 - 1.5
DNA	Layer-level	Fixed	1.5+**10**×1	10
MNasNet	Block-level	Variable	90+**10**×1	100
This Work	Block-level	Variable	9+**10**×0.1	1

Good OK Not good

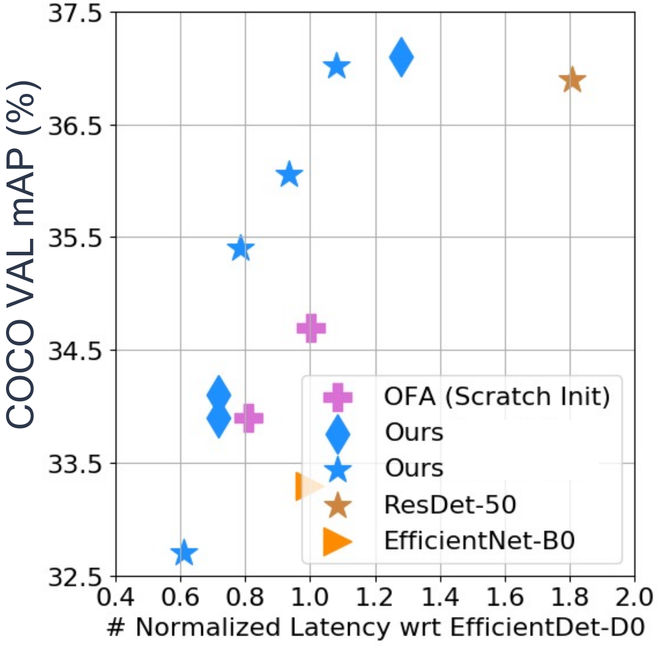
DONNA == MnasNet-level diversity at 100x lower cost

OFA = Han Cai, et al, "Once For All: Train One Network and Specialize it for Efficient deployment", ICLR2020
DNA = Changlin Li, "Blockwisely Supervised Neural Architecture Search with Knowledge Distillation", CVPR20
MNasNet = MingXing Tan, et al, "MNasNet: Platform-Aware Neural Architecture Search for Mobile", CVPR19
This work = Bert Moons et al, ""Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

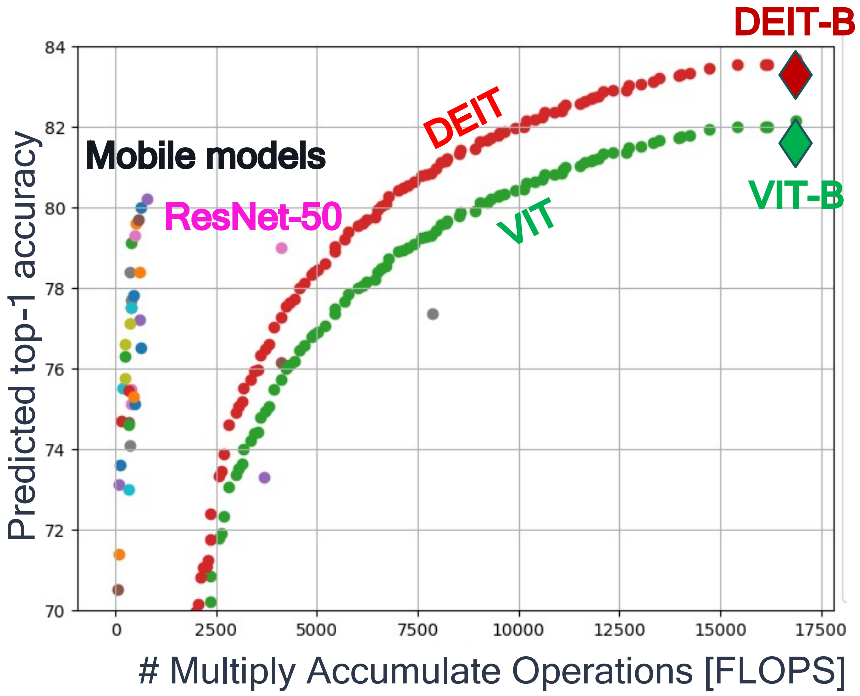
*Training 1 model from scratch = 450 epochs

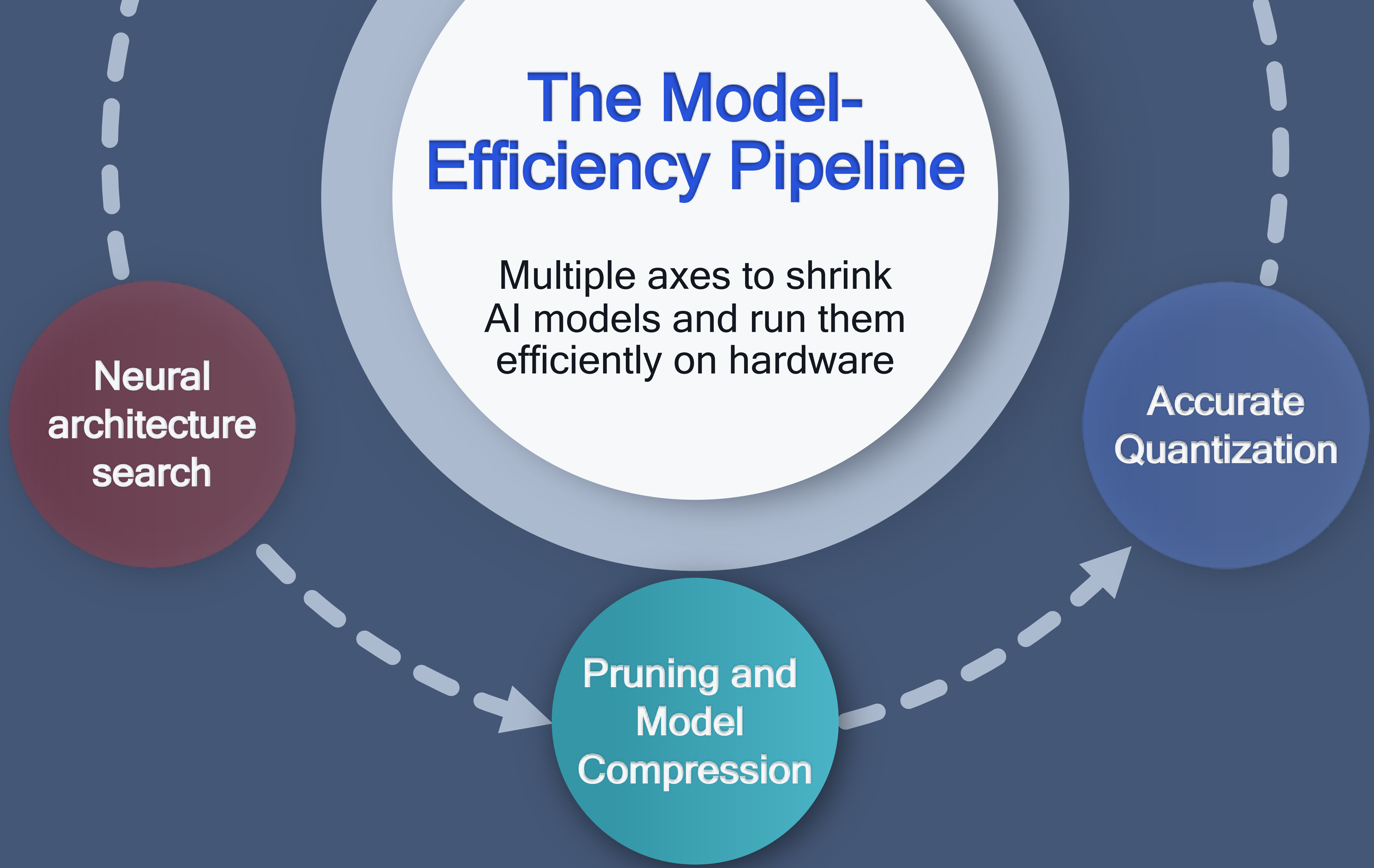
DONNA applies directly to downstream tasks and non-CNN neural architectures without conceptual code changes

Object Detection



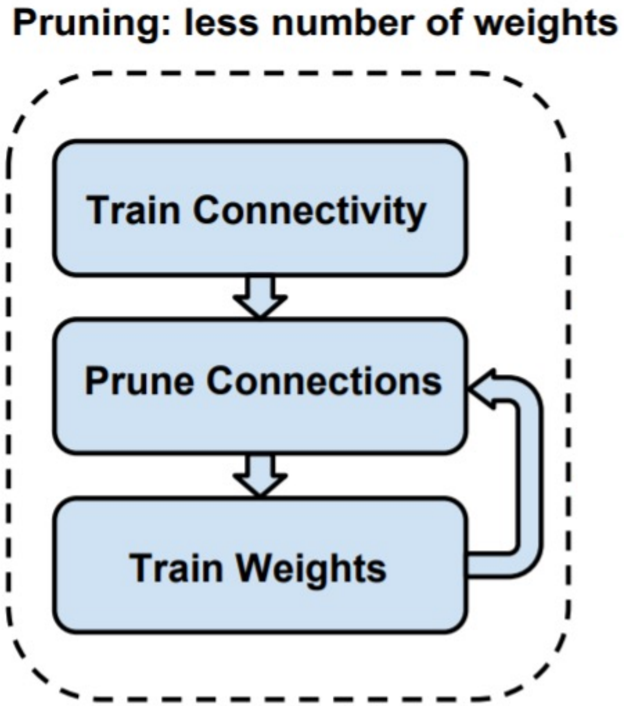
Vision Transformers

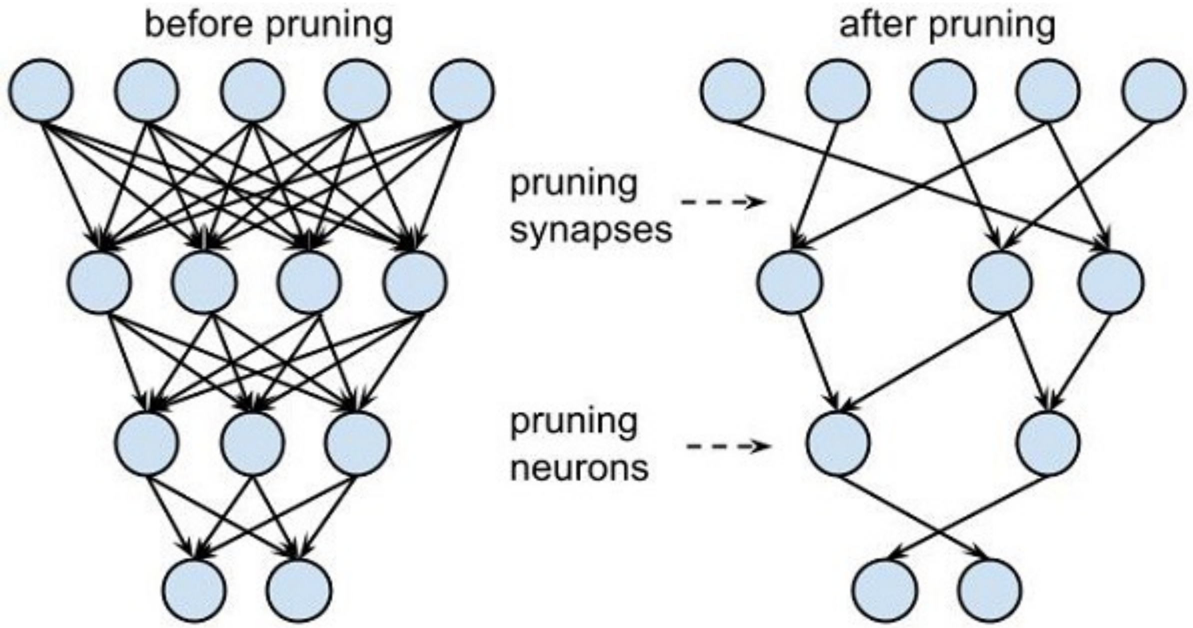



The Model-Efficiency Pipeline
Multiple axes to shrink AI models and run them efficiently on hardware
Neural architecture search
Pruning and Model Compression
Accurate Quantization

Unstructured Pruning of Neural Networks

Pruning removes unnecessary connections in the neural network.
Unstructured pruning is non-trivial to accelerate on parallel hardware.

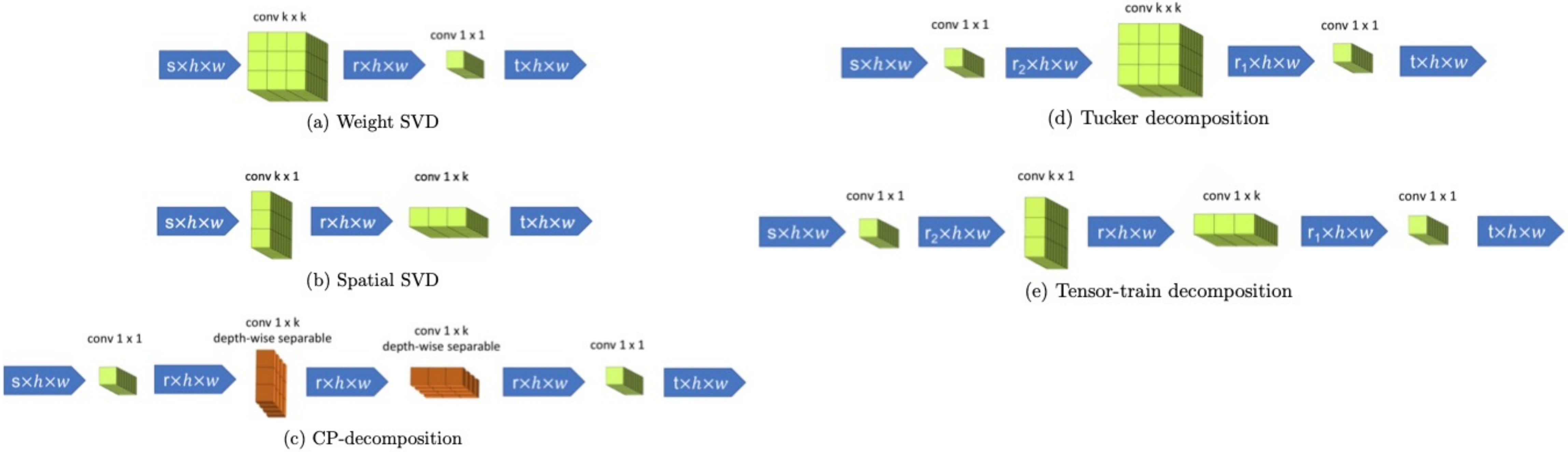




Song Han, et al, "Deep compression: compressing deep Neural Networks with Pruning, Trained Quantization and Huffman Coding ", NIPS2015

Structured compression through low rank approximations

Structured mathematical decompositions (SVD, CP, Tucker-II, Tensor-train,…) are easier to accelerate on parallel hardware

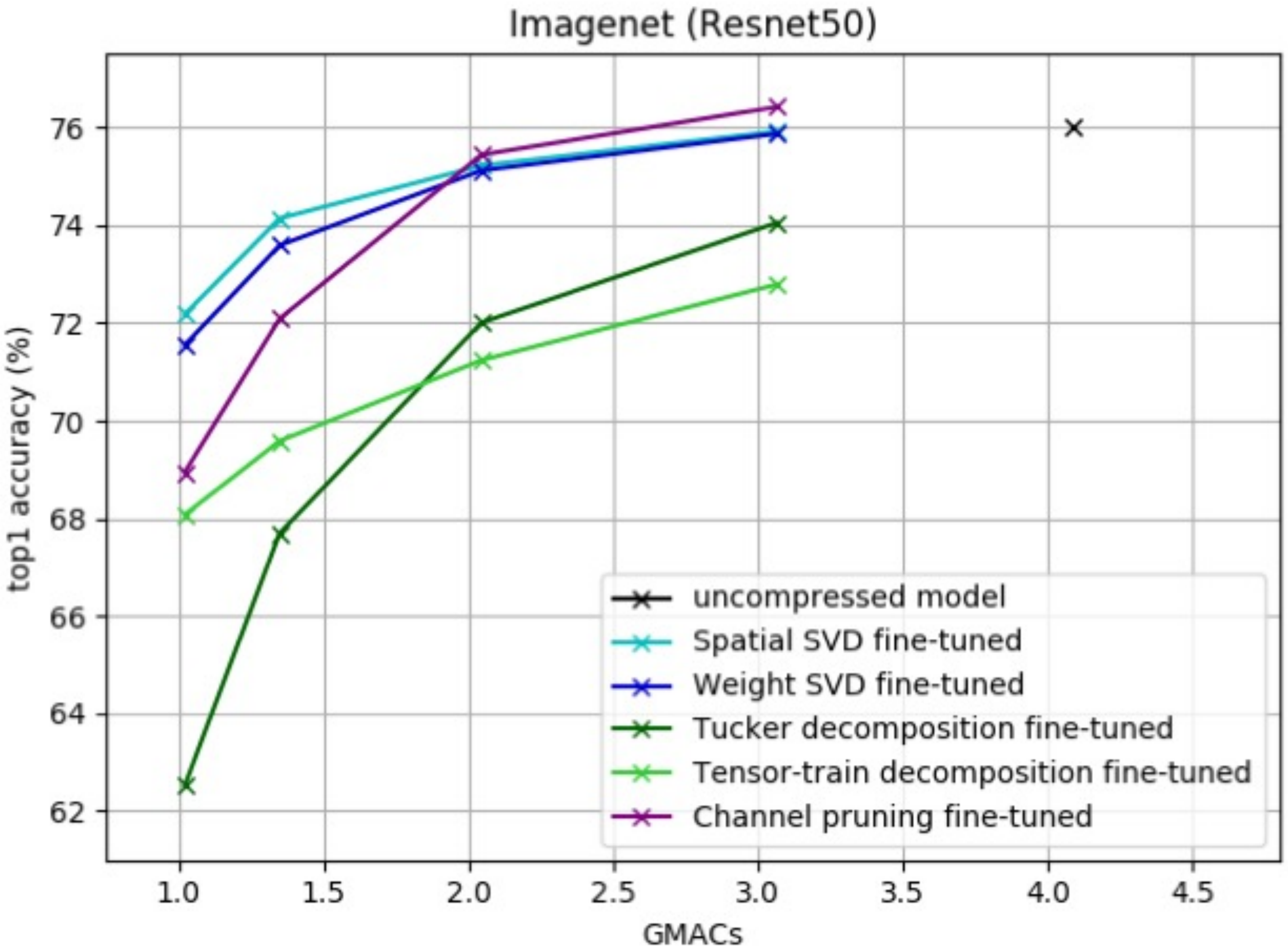


(a) Weight SVD
(b) Spatial SVD
(c) CP-decomposition
(d) Tucker decomposition
(e) Tensor-train decomposition

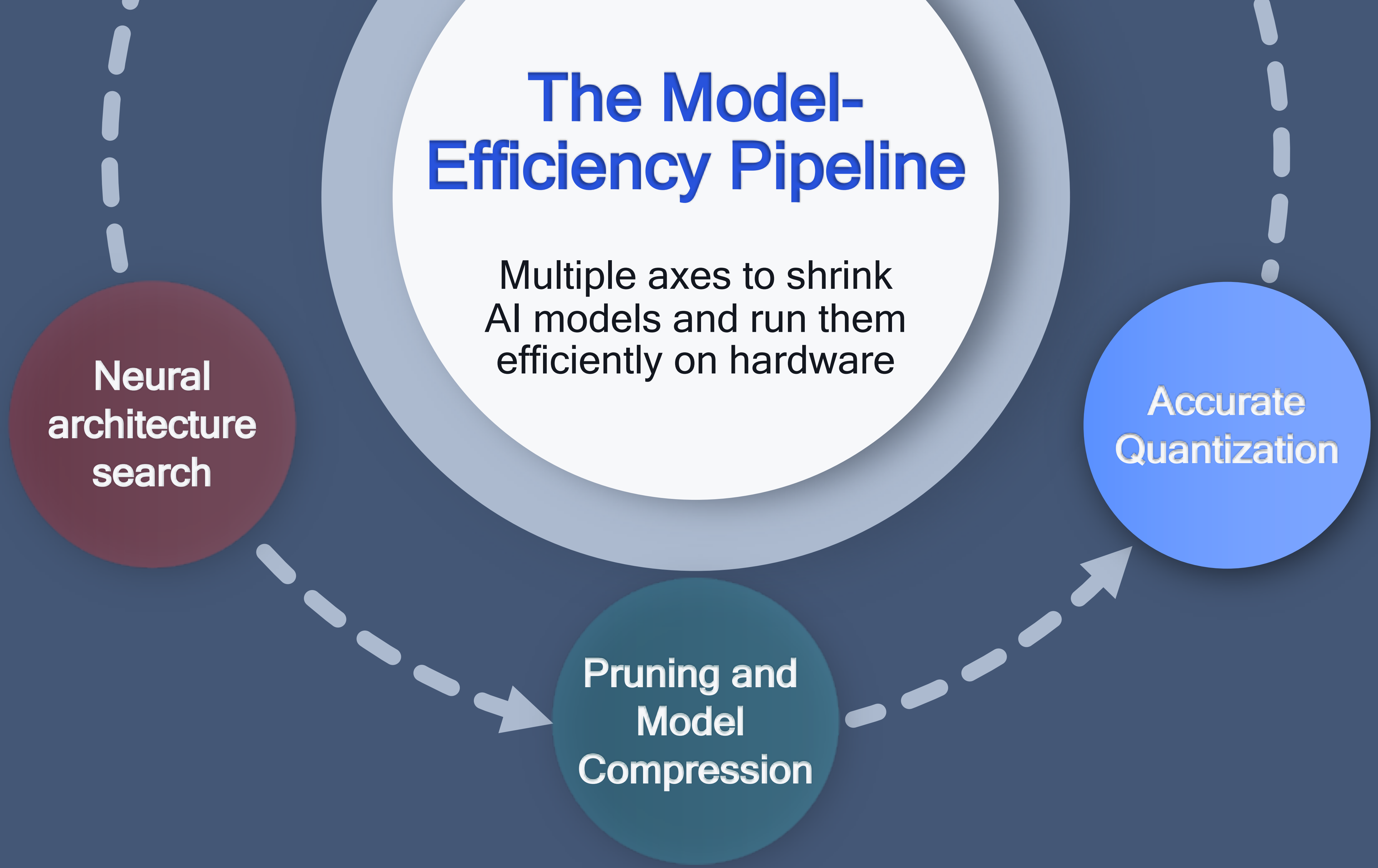
Andrey Kuzmin, et al, "Taxonomy and Evaluation of Structured Compression of Convolutional Neural Networks", Arxiv 2019

Structured compression through low rank approximations

- **(Structured) Channel Pruning** and **Spatial-SVD** typically work best.
- 50% compression @0.3% accuracy loss for ResNet-50



Andrey Kuzmin, et al, "Taxonomy and Evaluation of Structured Compression of Convolutional Neural Networks", Arxiv 2019


The Model-Efficiency Pipeline
Multiple axes to shrink AI models and run them efficiently on hardware
Neural architecture search
Pruning and Model Compression
Accurate Quantization

What is neural network quantization?

For any given trained neural network:

- Store weights in n bits
- Compute calculations in n bits

Benefits

- Reduced memory usage
- Reduced energy usage
- Lower latency

Quantization example

Pushing the limits of what's possible with quantization

Data-free quantization

Baseline training-free method with equalization and bias correction

- ☑ No training
- ☑ Data free

SOTA 8-bit results

<1% Accuracy drop for MobileNet V2 against FP32 model

Data-Free Quantization Through Weight Equalization and Bias Correction (Nagel, van Baalen, et al., ICCV 2019)

AdaRound

Outperform rounding to nearest

- ☑ No training
- ☑ Minimal unlabeled data

SOTA 4-bit weight results

<2.5% Accuracy drop for MobileNet V2 against FP32 model

Up or Down? Adaptive Rounding for Post-Training Quantization (Nagel, Amjad, et al., ICML 2020)

Bayesian bits

Automated mixed-precision

- ☑ Training required
- ☑ Training data required
- ☑ Jointly learns bit-width precision and pruning

SOTA mixed-precision results

<1% Accuracy drop for MobileNet V2 against FP32 model for mixed precision model with **computational complexity equivalent to a 4-bit weight model**

Bayesian Bits: Unifying Quantization and Pruning van Baalen, Louizos, et al., NeurIPS 2020)

SOTA: State-of-the-art

Adaround

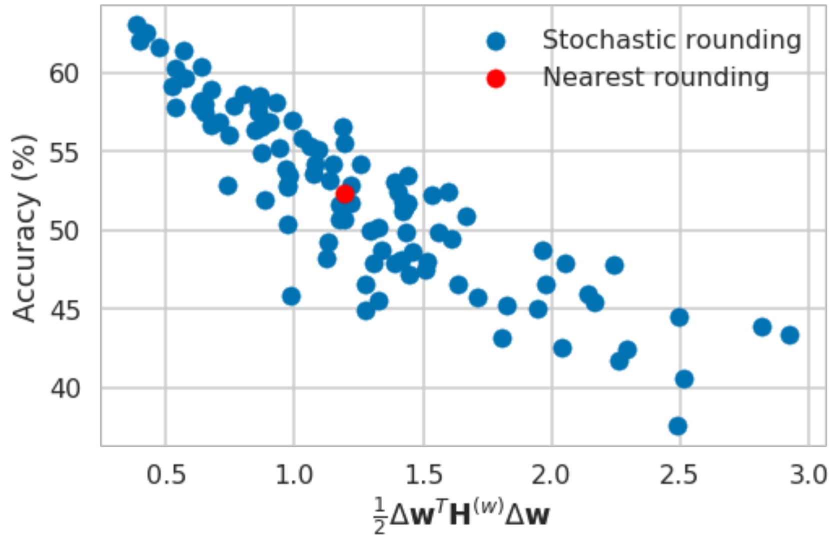
Up or Down? Adaptive Rounding for Post-Training Quantization (Nagel, Amjad, et al., ICML 2020)

- Traditional post-training weight quantization uses **rounding to nearest:**

- However, rounding-to-nearest is not optimal

Rounding Method	Accuracy (%)
Nearest	52.29
Floor / Ceil	00.10
Stochastic	52.06±5.52
Stochastic (best)	63.06

4-bit weight quantization of 1st layer of Resnet18 ,tested on ImageNet.



Up or Down?

How can we systematically find the best rounding choice?

AdaRound: learning to round

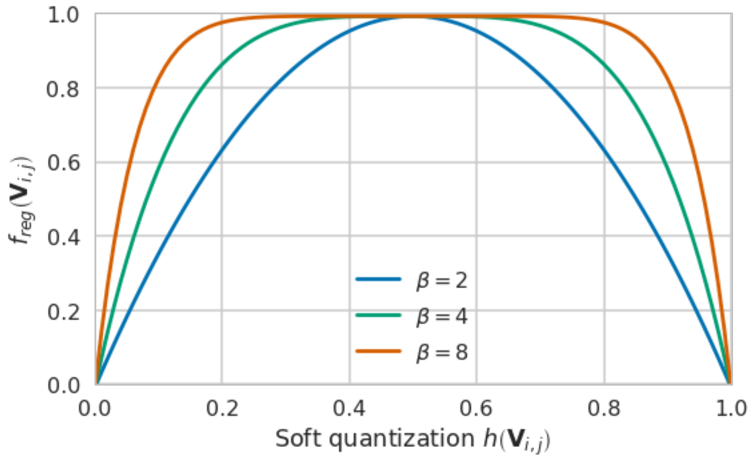
- Minimize per-layer L2 loss of output features

Regularizer forces V to be 0 or 1

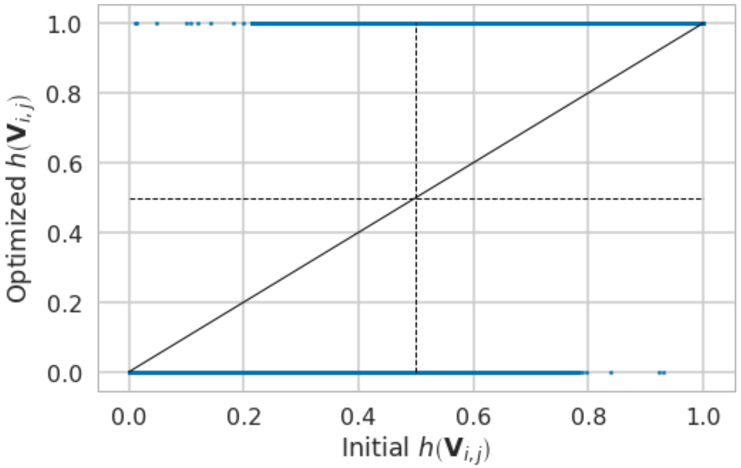
$$\underset{\mathbf{V}}{\arg\min} \left\| \mathbf{W}\mathbf{x} - \widetilde{\mathbf{W}}\mathbf{x} \right\|_F^2 + \boxed{\lambda f_{reg}\left(\mathbf{V}\right)}$$

$$\widetilde{\mathbf{W}} = \mathrm{s} \cdot clip\left(\boxed{\left\lfloor \frac{\mathbf{W}}{\mathrm{s}} \right\rfloor} + \boxed{h\left(\mathbf{V}\right)}, \mathrm{n}, \mathrm{p} \right)$$

round down + learned value between [0,1]



- Regularization: $$f_{reg}\left(\mathbf{V}\right) = \sum_{i,j} 1 - \left|2h\left(\mathbf{V}_{i,j}\right) - 1\right|^{\beta}$$



Comparison to literature

Setting a new SOTA for 4-bit post-training weight quantization

Optimization	#bits W/A	Resnet18	Resnet50	InceptionV3	MobilenetV2
Full precision	32/32	69.68	76.07	77.40	71.72
DFQ (Nagel et al., 2019)	8/8	69.7	-	-	71.2
DFQ (our impl.)	4/8	38.98	52.84	-	46.57
Bias corr (Banner et al., 2019)	4*/8	67.4	74.8	59.5	-
AdaRound w/ act quant	4/8	**68.55±0.01**	**75.01±0.05**	**75.72±0.09**	**69.25±0.06**†

Table 7. Comparison among different post-training quantization strategies in the literature. We report results for various models in terms of ImageNet validation accuracy (%). *Uses per channel quantization. †Using CLE (Nagel et al., 2019) as preprocessing.

Tools are open-sourced through AIMET

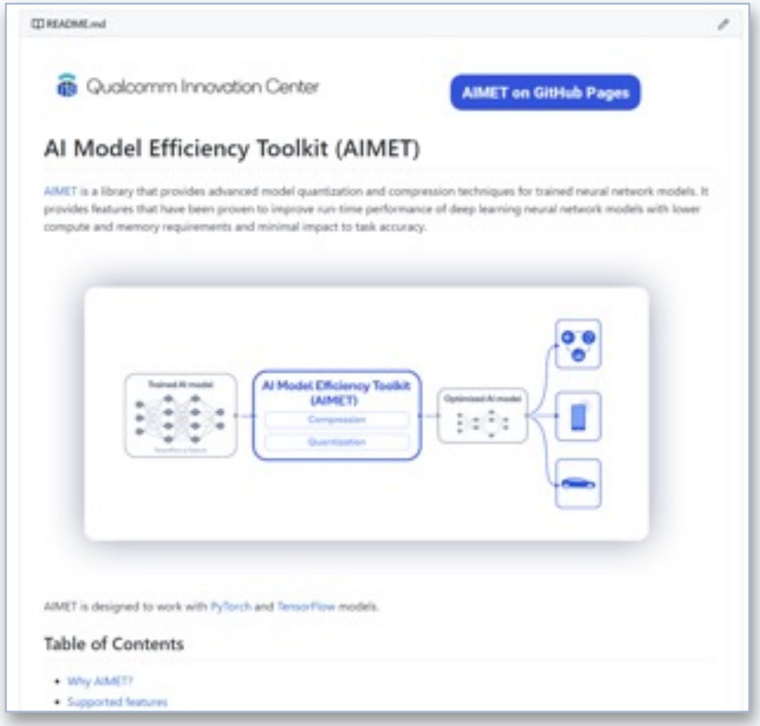
github.com/quic/aimet

github.com/quic/aimet-model-zoo

AIMET Model Zoo is a product of Qualcomm Innovation Center, Inc.

AIMET

State-of-the-art quantization and compression techniques



README.md

Qualcomm Innovation Center

AIMET on GitHub Pages

AI Model Efficiency Toolkit (AIMET)

AIMET is a library that provides advanced model quantization and compression techniques for trained neural network models. It provides features that have been proven to improve run-time performance of deep learning neural network models with lower compute and memory requirements and minimal impact to task accuracy.

AIMET is designed to work with PyTorch and TensorFlow models.

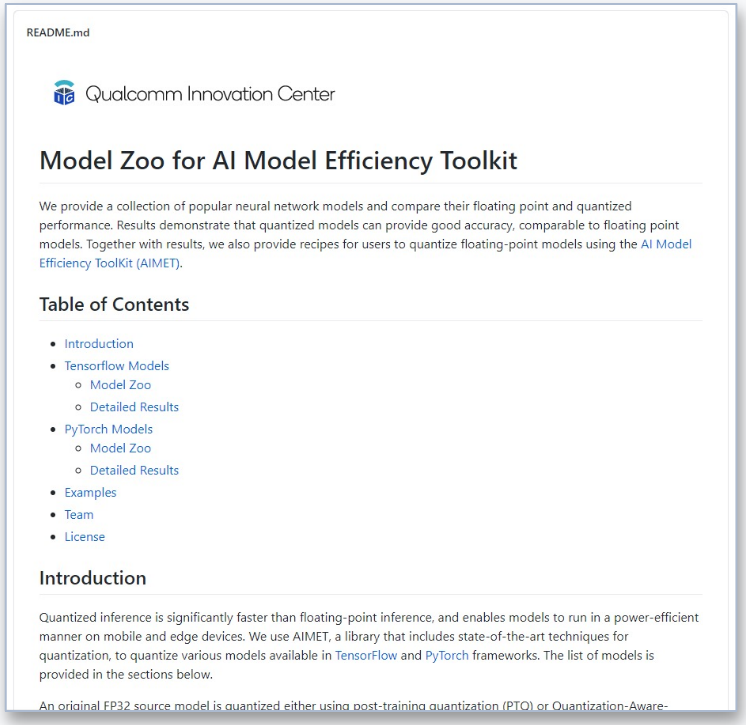
Table of Contents

- Why AIMET?
- Supported features

github.com/quic/aimet

AIMET Model Zoo

Accurate pre-trained 8-bit quantized models



README.md

Qualcomm Innovation Center

Model Zoo for AI Model Efficiency Toolkit

We provide a collection of popular neural network models and compare their floating point and quantized performance. Results demonstrate that quantized models can provide good accuracy, comparable to floating point models. Together with results, we also provide recipes for users to quantize floating-point models using the AI Model Efficiency ToolKit (AIMET).

Table of Contents

- Introduction
- Tensorflow Models
  - Model Zoo
  - Detailed Results
- PyTorch Models
  - Model Zoo
  - Detailed Results
- Examples
- Team
- License

Introduction

Quantized inference is significantly faster than floating-point inference, and enables models to run in a power-efficient manner on mobile and edge devices. We use AIMET, a library that includes state-of-the-art techniques for quantization, to quantize various models available in TensorFlow and PyTorch frameworks. The list of models is provided in the sections below.

An original FP32 source model is quantized either using post-training quantization (PTQ) or Quantization-Aware-

github.com/quic/aimet-model-zoo

Join our open-source projects

AIMET Model Zoo includes popular quantized AI models

Accuracy is maintained for INT8 models – less than 1% loss*

<1% Loss in accuracy*

Tensorflow

Model	FP32	INT8	Metric
ResNet-50 (v1)	75.21%	74.96%	Top-1 accuracy*
MobileNet-v2-1.4	75%	74.21%	Top-1 accuracy*
EfficientNet Lite	74.93%	74.99%	Top-1 accuracy*
SSD MobileNet-v2	0.2469	0.2456	mAP*
RetinaNet	0.35	0.349	mAP*
Pose estimation	0.383	0.379	mAP*
SRGAN	25.45	24.78	PSNR*

Pytorch

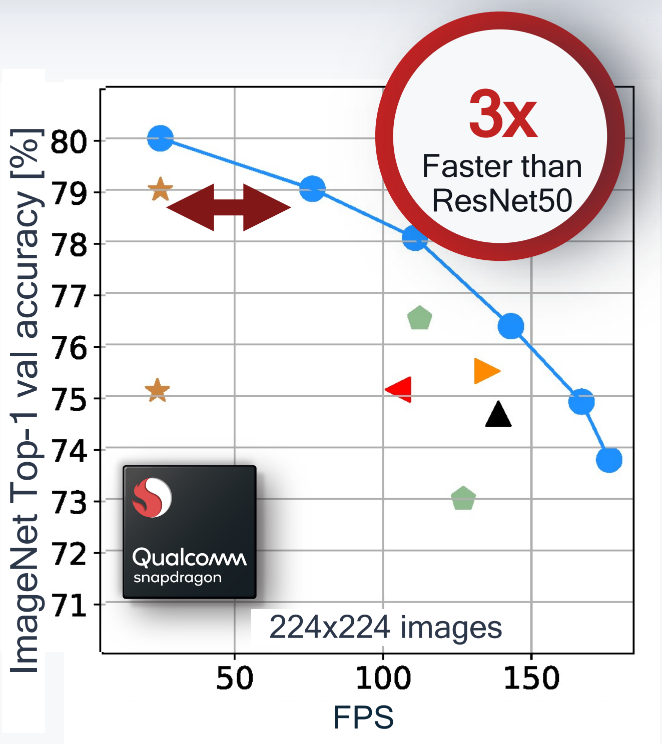
Model	FP32	INT8	Metric
MobileNetV2	7167%	71.14%	Top-1 accuracy*
EfficientNet-lite0	75.42%	74.44%	Top-1 accuracy*
DeepLabV3+	72.62%	72.22%	mIoU*
MobileNetV2-SSD-Lite	68.7%	68.6%	mAP*
Pose estimation	0.364	0.359	mAP*
SRGAN	25.51	25.5	PSNR
DeepSpeech2	9.92%	10.22%	WER*

*: Comparison between FP32 model and INT8 model quantized with AIMET.
For further details, check out: https://github.com/quic/aimet-model-zoo/

What's next in efficient on-device AI

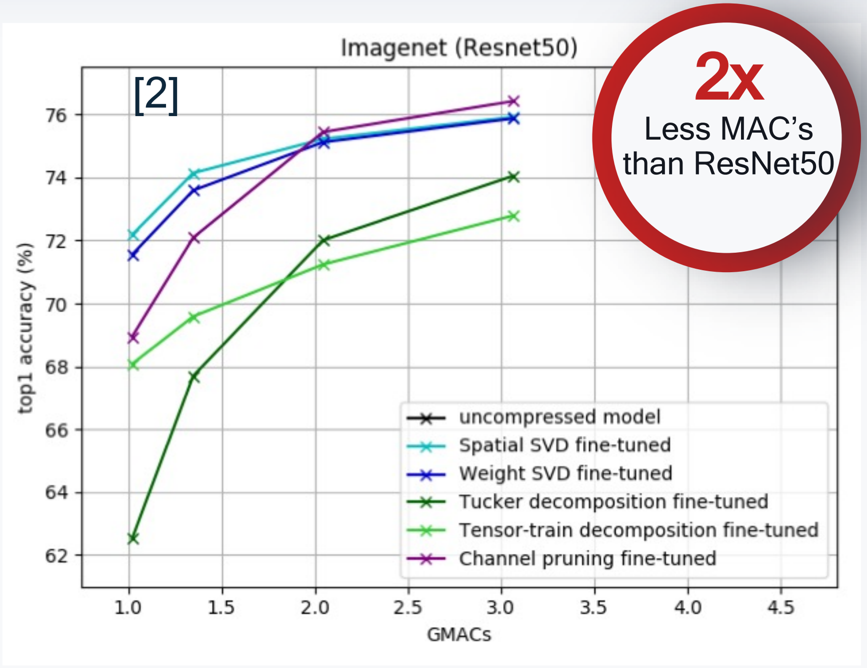
Ultimately limited gains from NAS, compression, uniform quantization

Current tools optimize existing architectures, leading to 1-3x gains over standard networks on device



Mobile SoC throughput[1]
(Adreno 660 GPU)

NAS



Compression

8-bit Integer — up to 16X

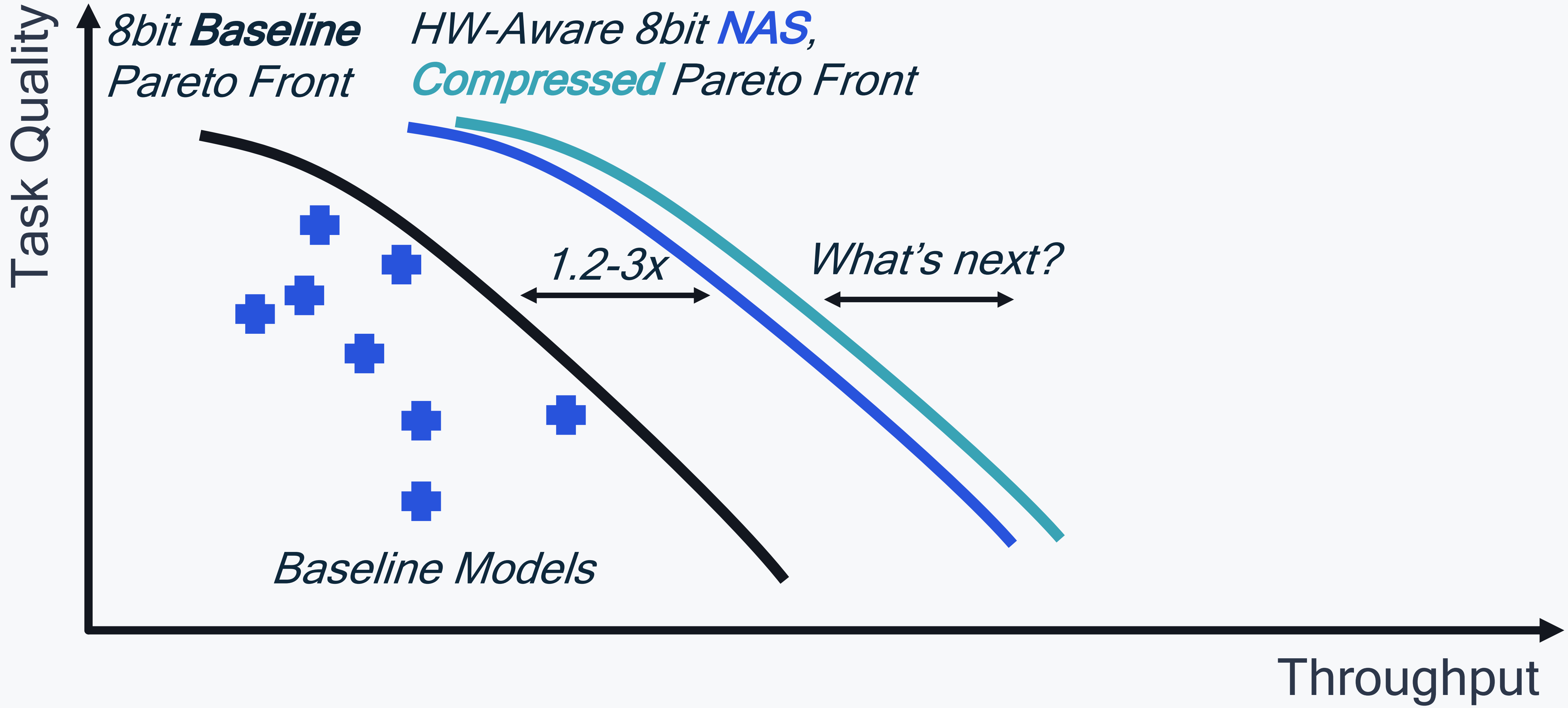
4-bit Integer — up to 64X

4-8b established

1 Qualcomm Adreno 660 GPU in the Snapdragon 888 running on the Samsung Galaxy S21

[2]Andrey Kuzmin, et al, "Taxonomy and Evaluation of Structured Compression of Convolutional Neural Networks", Arxiv 2019

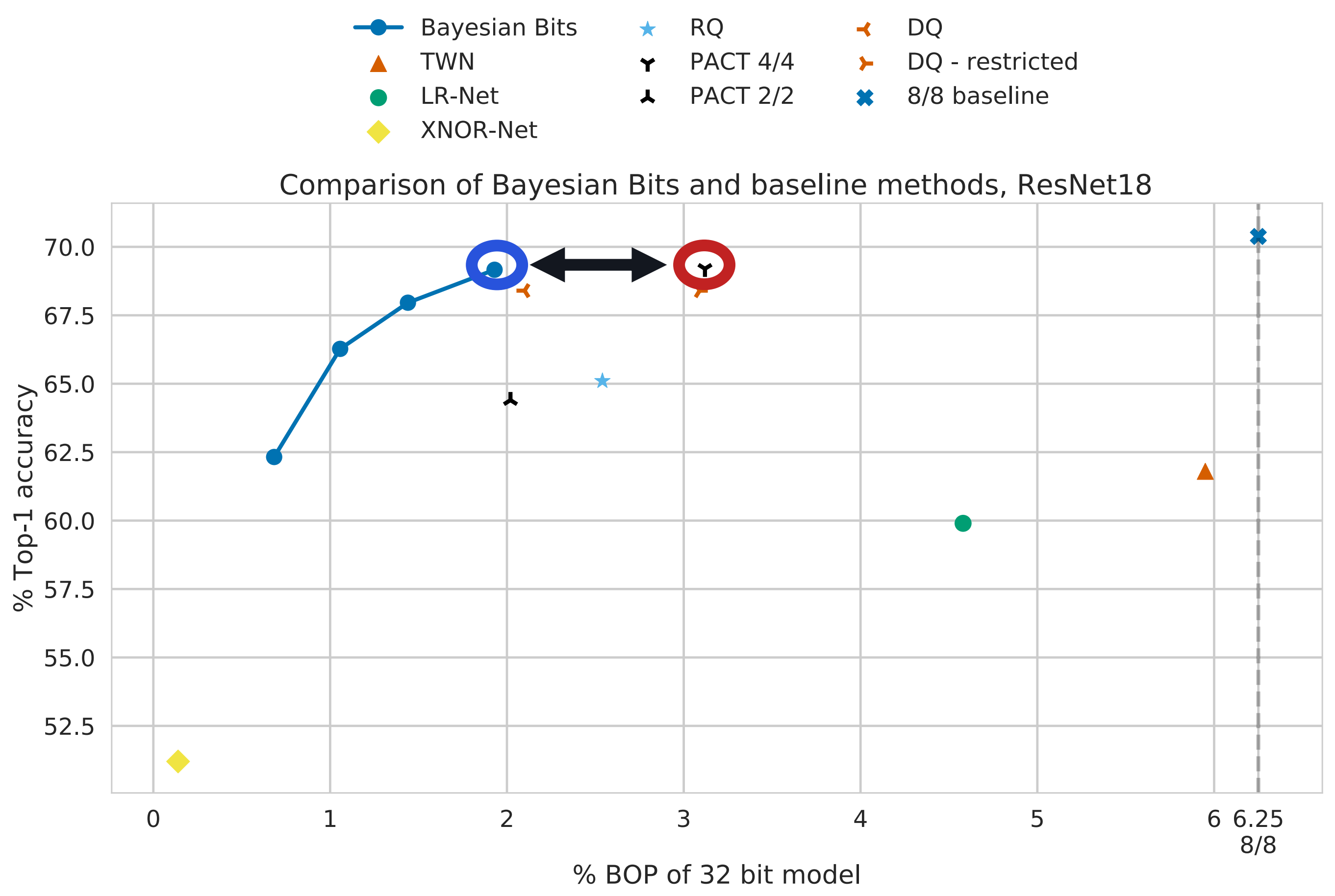
What's next in efficient AI models?



Mixed-Precision Quantized NAS

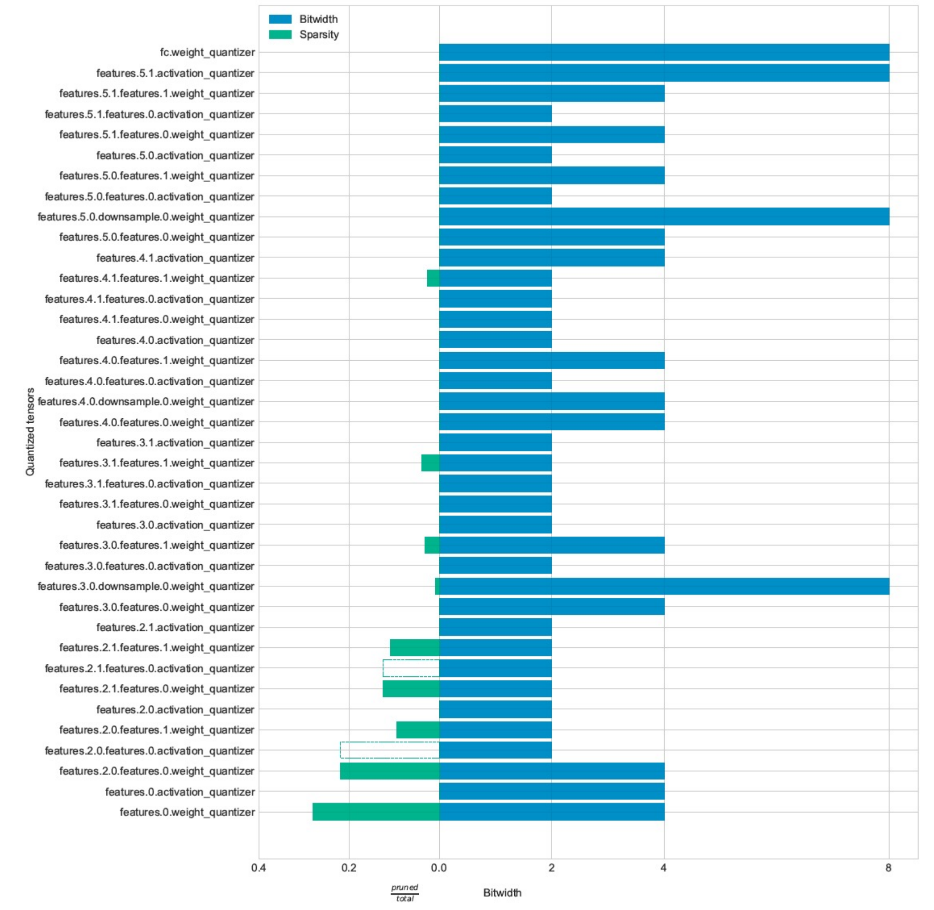
Mixed Precision outperforms uniform quantization

Bayesian Bits: Neural Networks can be optimized for mixed-precision.



During training, the network automatically finds the optimal trade-off between network complexity and accuracy

Mart Van Baalen, et ak "Bayesian Bits: Unifying Quantization and Pruning", Neurips 2020

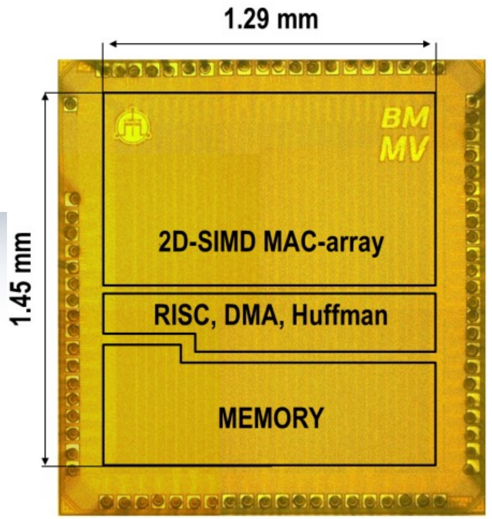


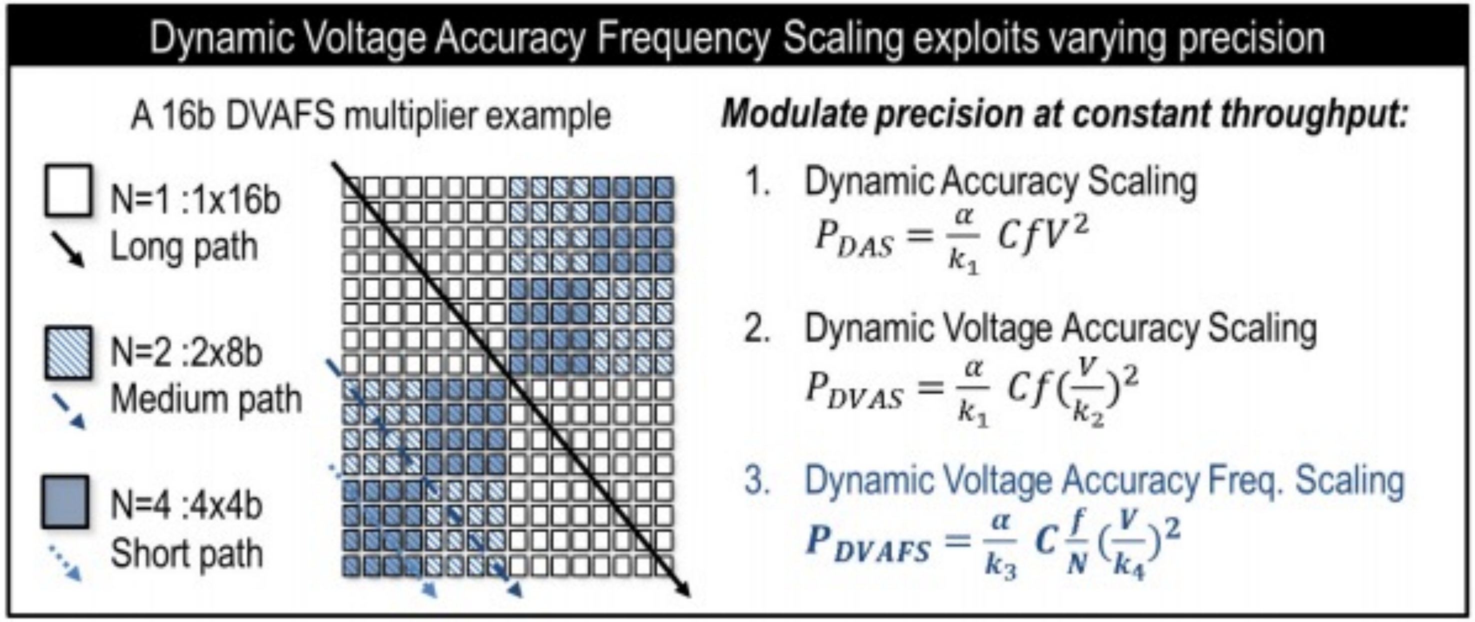
The result: Some layers are fine with 8 bits, while others are fine with 2 bits. And some layers are pruned (green)

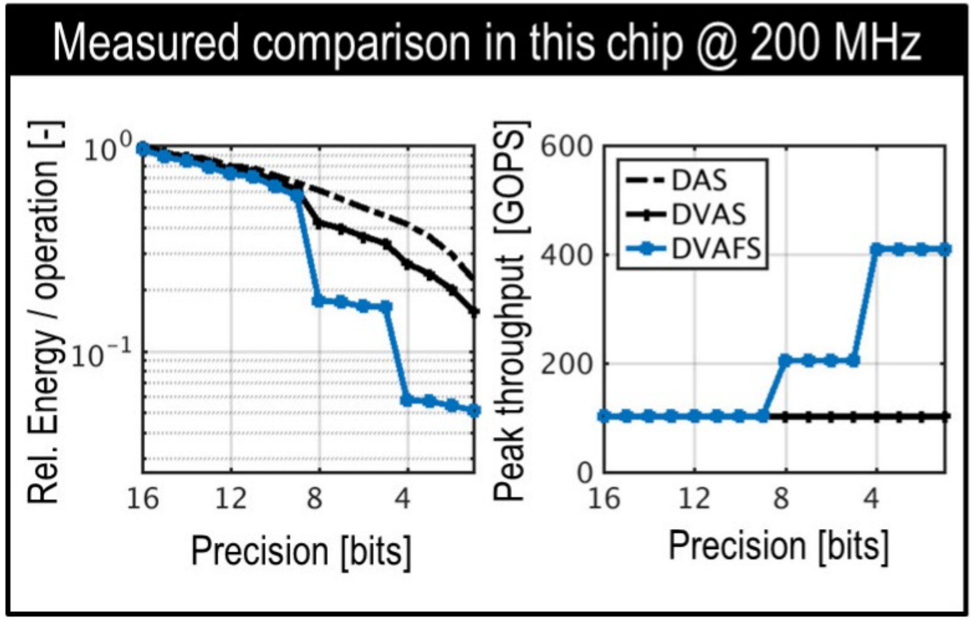
Many Ways to gain from mixed-precision

An academic system level example

- DVAFS: DVAS + subword parallelism   op/J **10x** @ 2b vs 8b



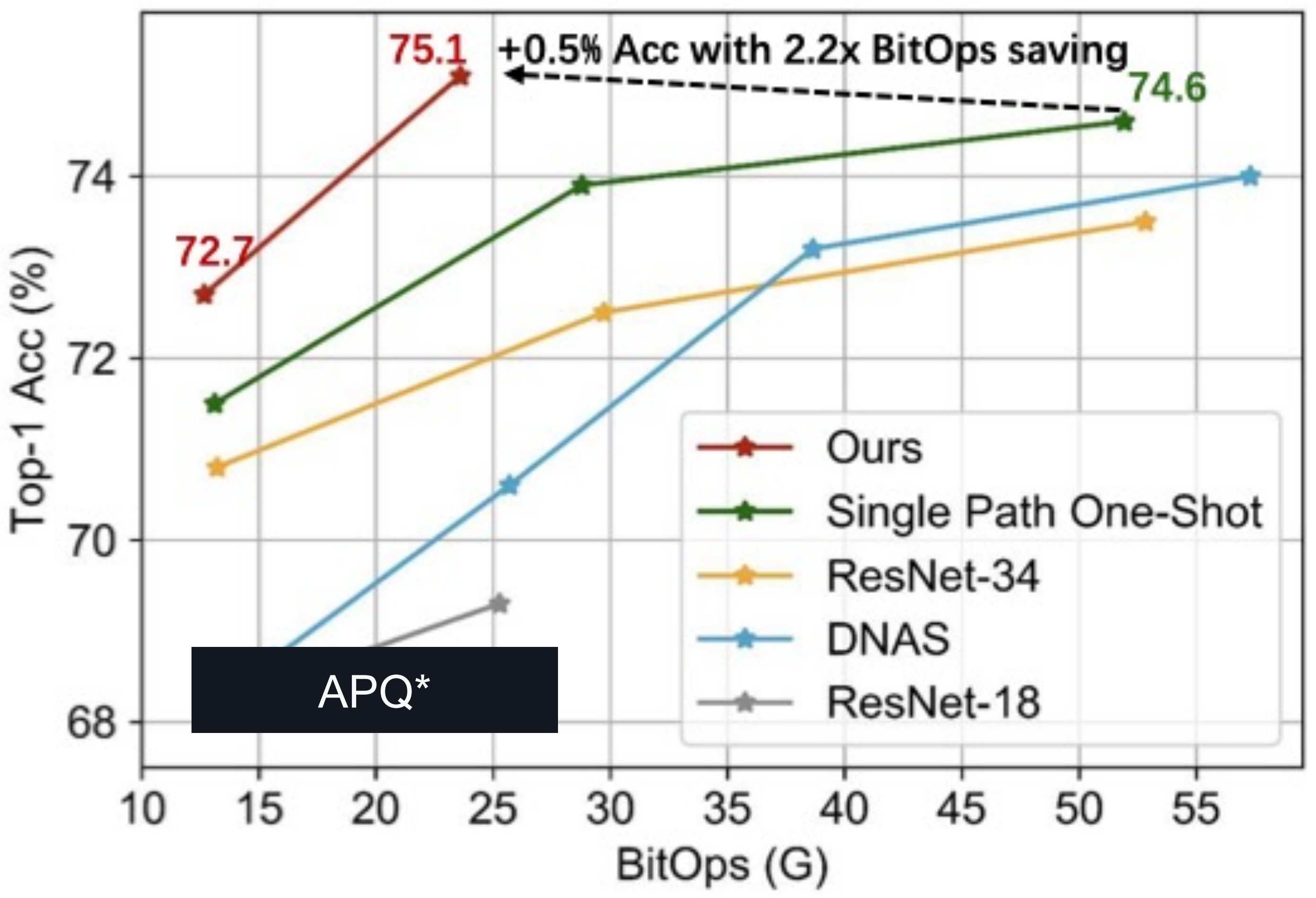




Bert Moons, et al, "Envision: a 0.26-to-10 TOPS/W Subword-Parallel Dynamic-Voltage-Accuracy-Frequency-Scalable Convolutional Neural Network Processor in 28nm FDSOI", ISSCC2017

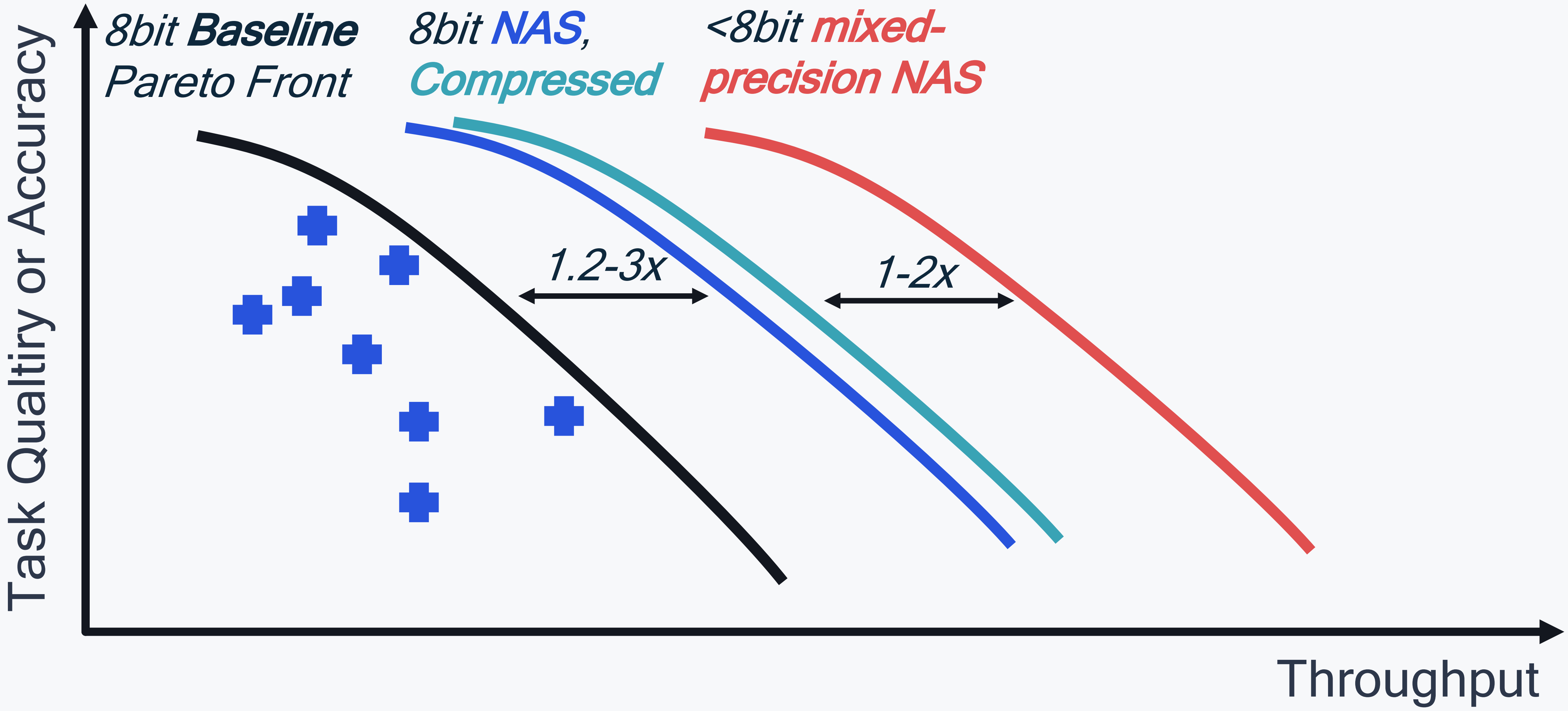
Mixed Precision Quantized NAS

- APQ builds on top of OFA
- +/- 1%, or 2.2x BOPS gains expected through joint NAS and Quantization



Tianzhe Wang, et al, “”APQ: Joint Search for Network Architecture, Pruning and Quantization Policy”, CVPR2020

What's next in efficient AI models?

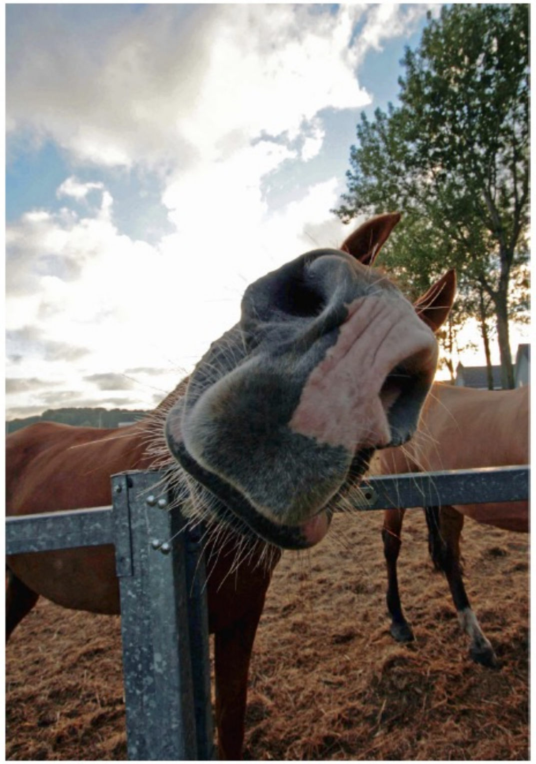


Conditional networks

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Classification: some samples are easier than others

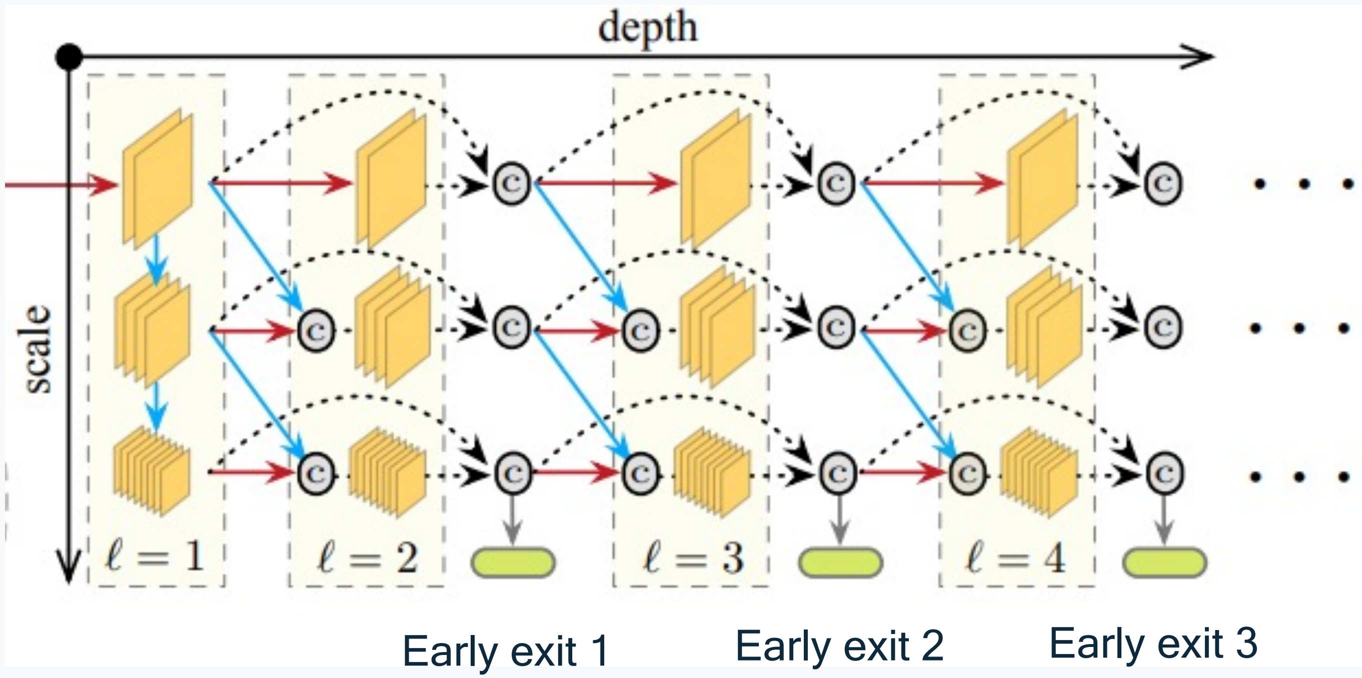



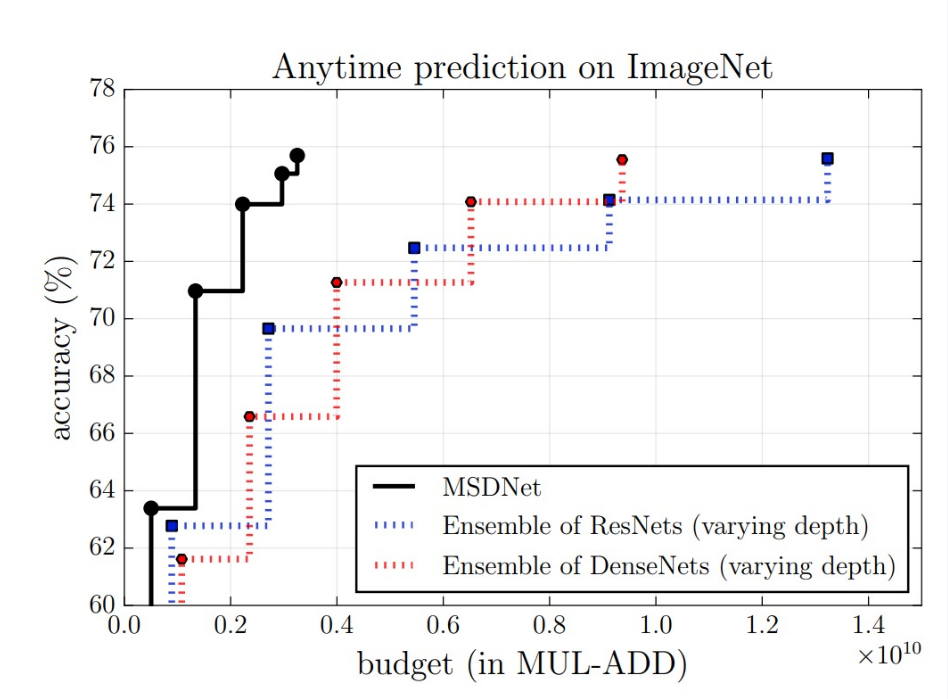
Copyright Pixel Addict and Doyle (CC BY-ND 2.0), no changes made
found through Huang, G, et al, "Multi-Scale Dense Networks for Resource Efficient Image Classification", ICLR2018

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Early exiting, some samples are easier than others





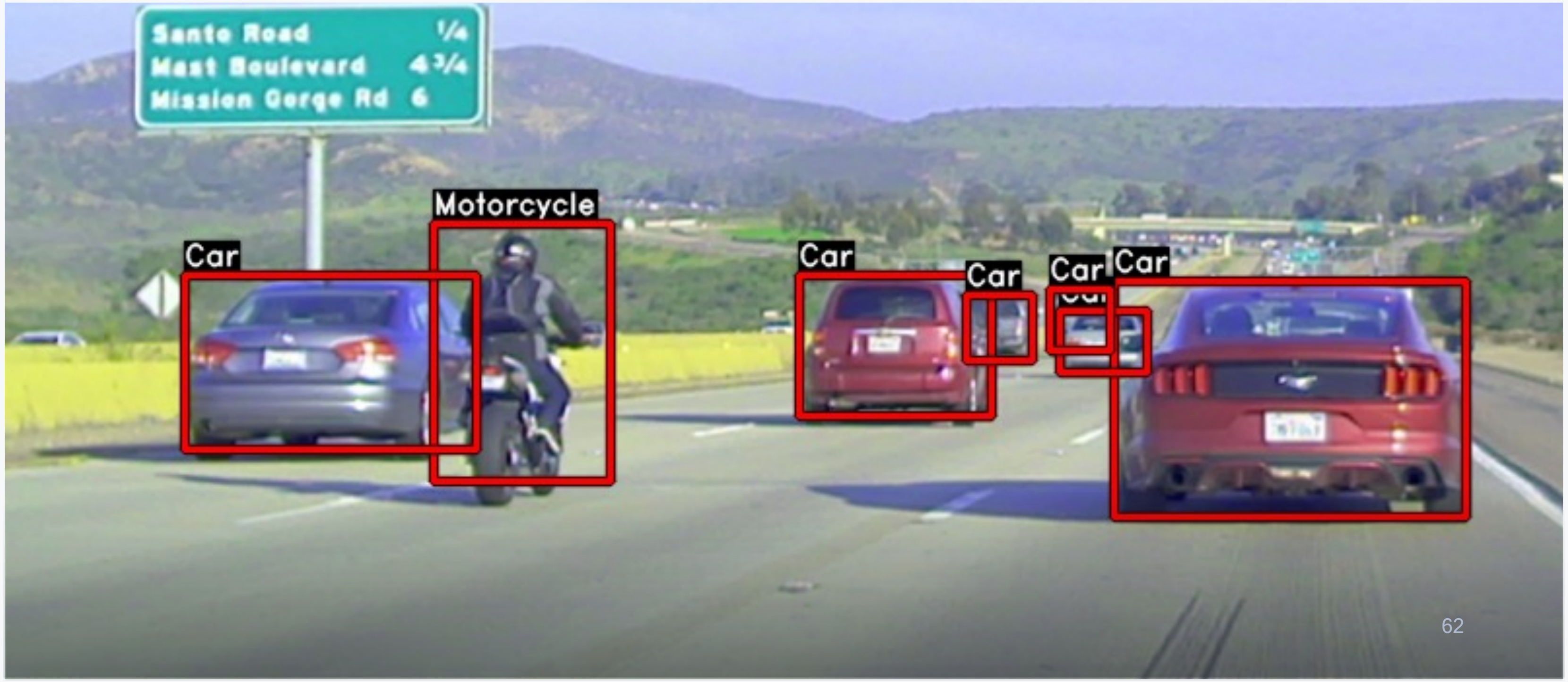
Huang, G, et al, "Multi-Scale Dense Networks for Resource Efficient Image Classification", ICLR2018

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Segmentation: **backgrounds** are **abundant** and **easy** to recognize

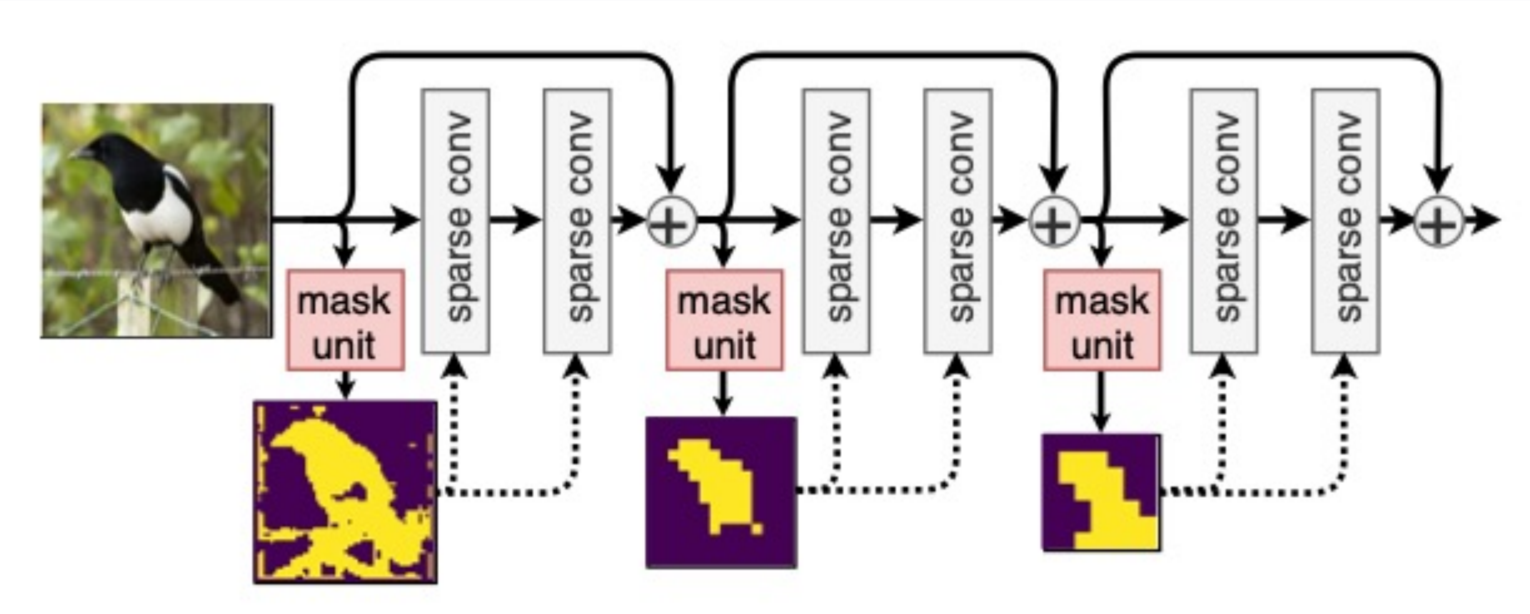
Detection: **Objects of interest** are relatively **rare**



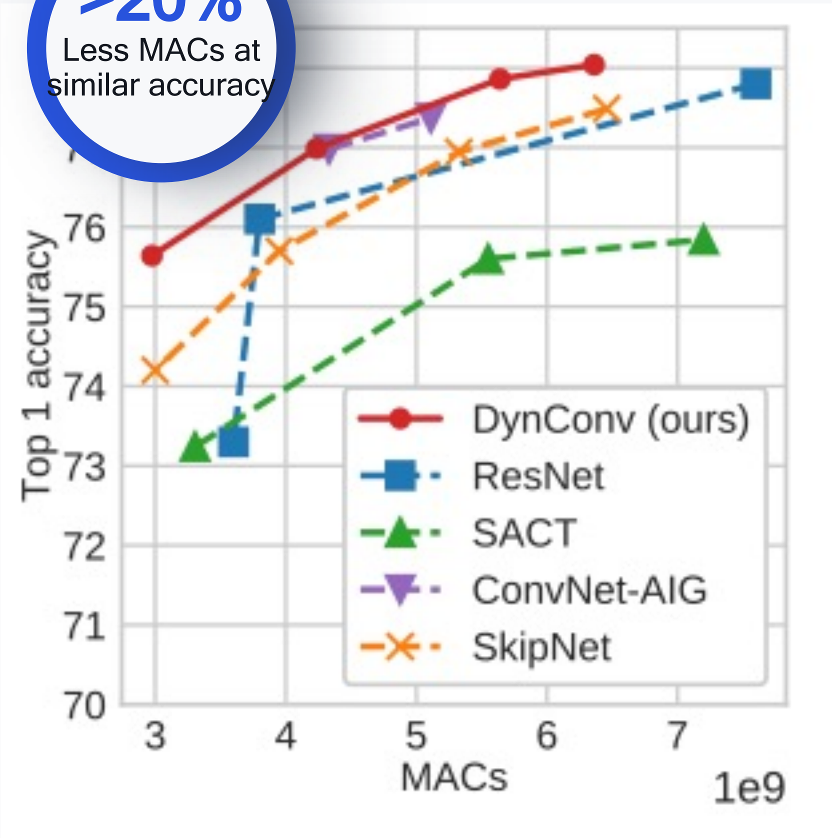
Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Dynamic Convolutions: exploiting spatial sparsity



Thomas Verelst, et al, "Dynamic Convolutions: Exploiting spatial sparsity for faster inference", CVPR20



>20%
Less MACs at similar accuracy

Conditional computing as a complementary technique to NAS

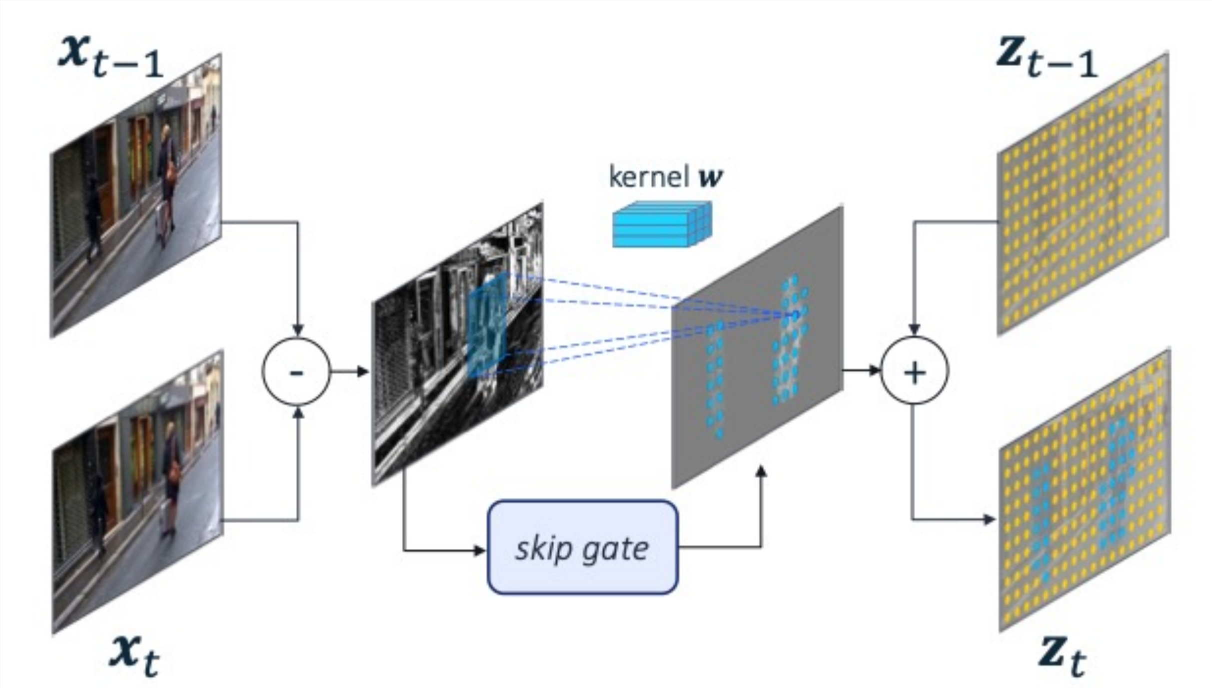
Input-dependent network architectures spend less time on easier samples

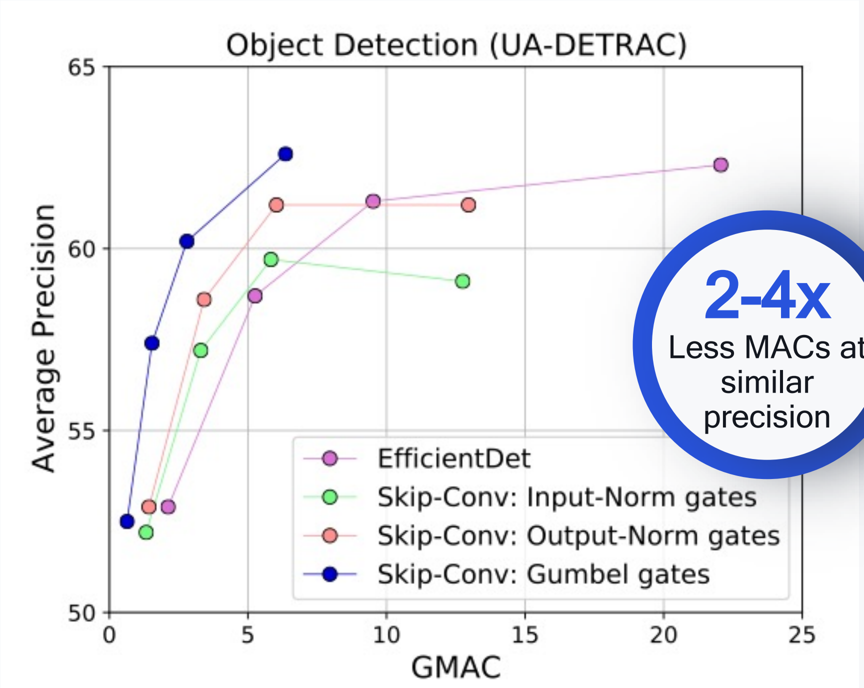
Video: Subsequent frames are correlated

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Temporal Skip-Convolutions in video segmentation/detection





2-4x Less MACs at similar precision

Amirhossein Habibian, et al, "Skip-Convolutions for Efficient Video Processing", CVPR21

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Conditional Early exiting in video/action recognition

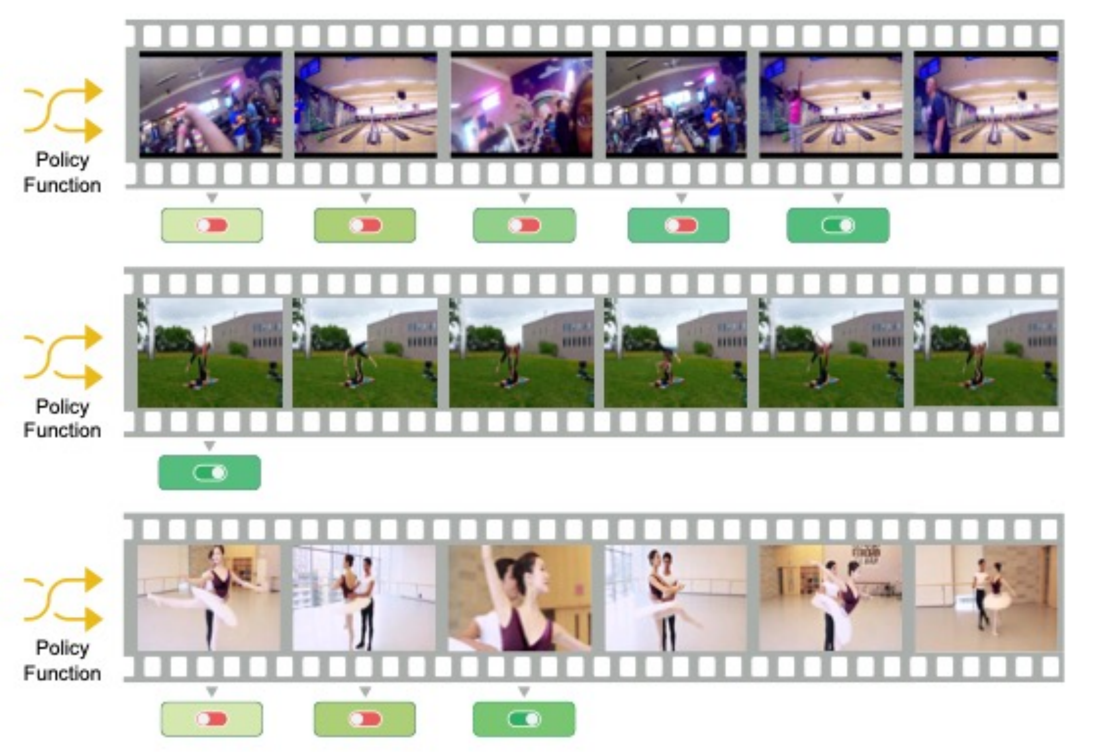


Figure 1: **Efficient video recognition by early exiting.**

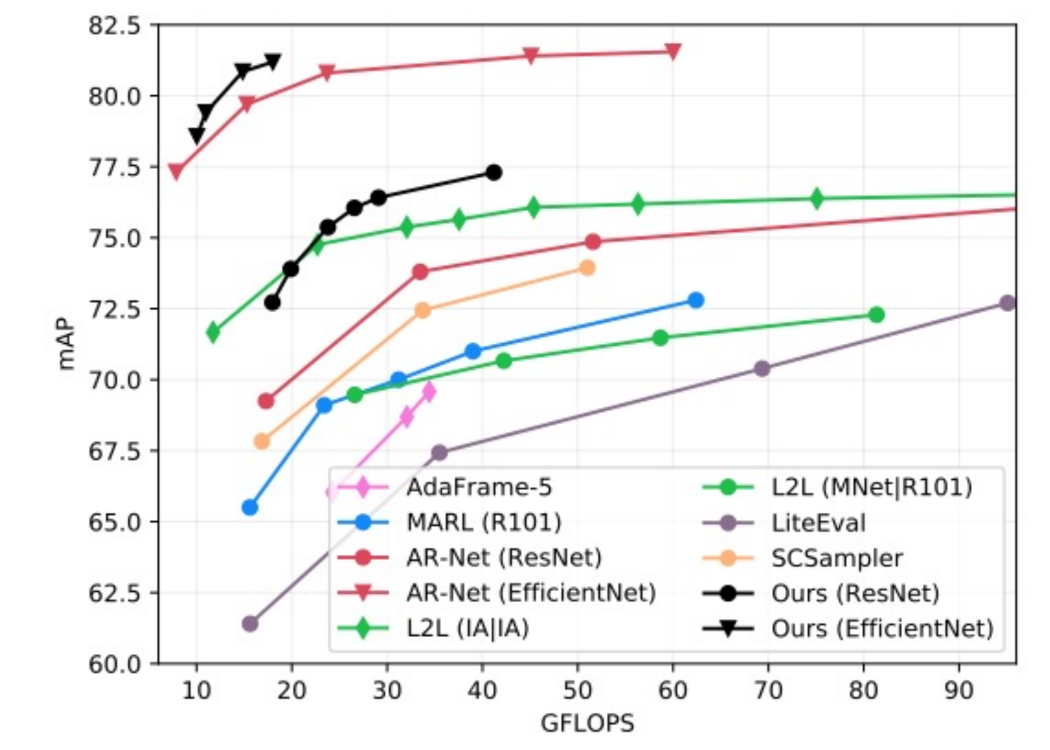
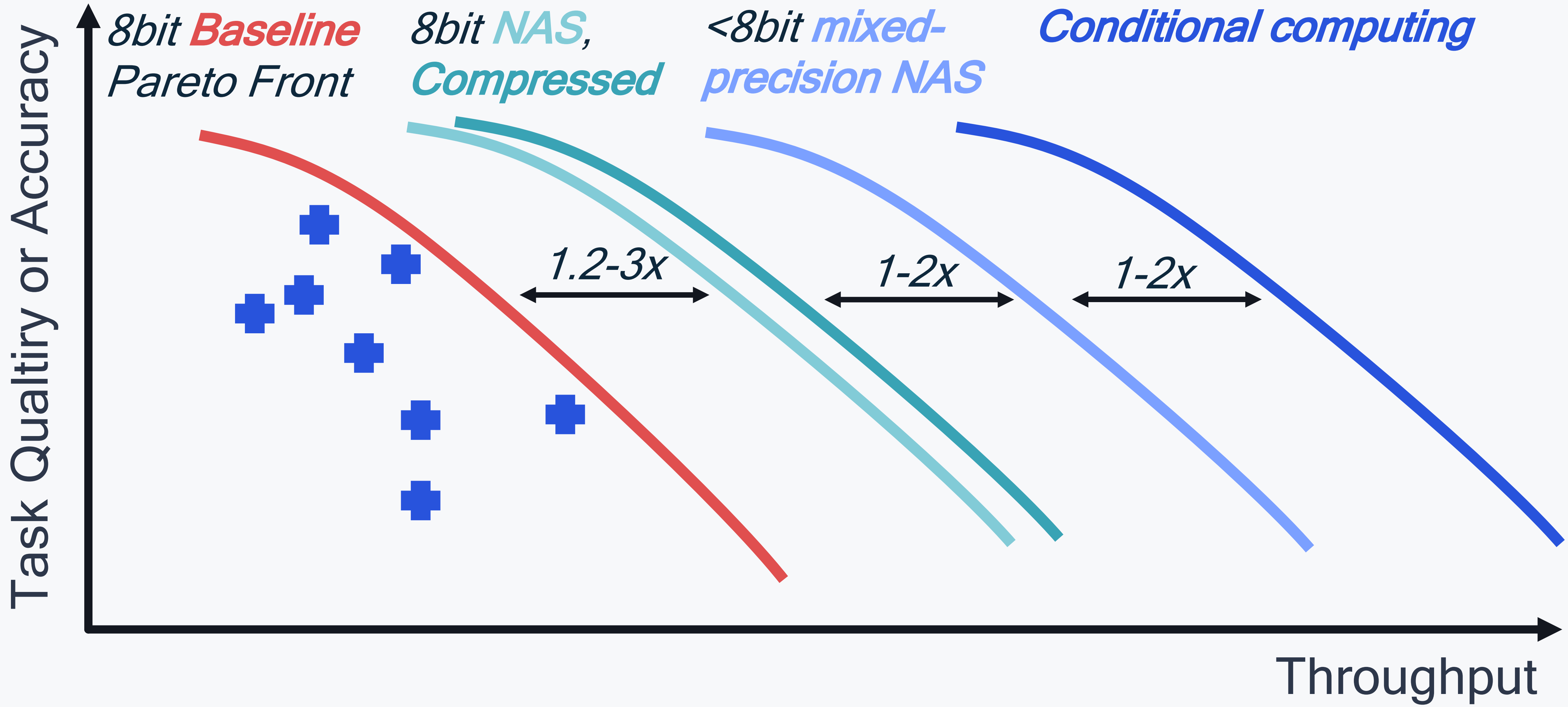


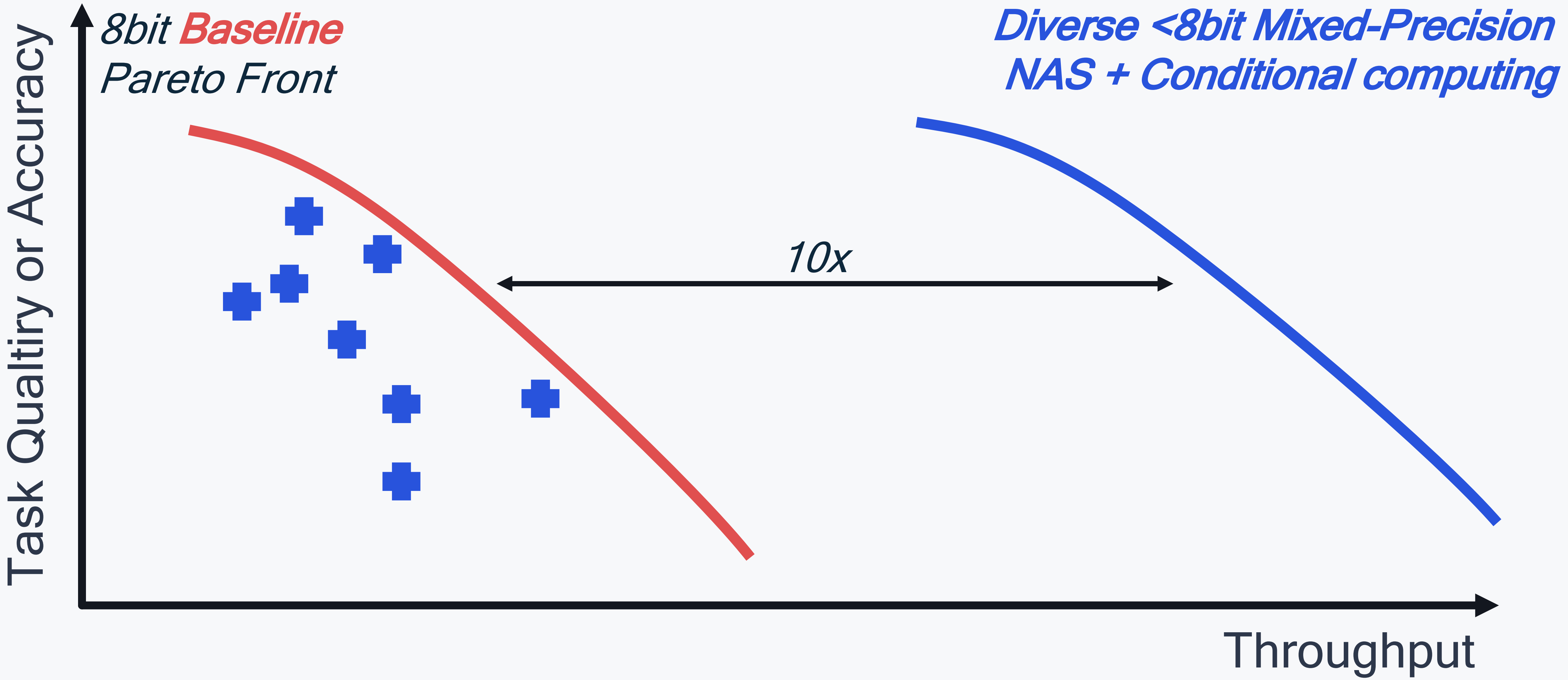
Figure 6: **Accuracy vs. efficiency curves on ActivityNet.**

Amir Ghodrati, et al, "FrameExit: Conditional Early Exiting for efficient Video Recognition", CVPR21

What's next in efficient AI models?



What's next in efficient AI models?



Overview

- Energy-Efficient machine learning and the computational budget gap
- The Model-Efficiency Pipeline reduces the cost of on-device inference

Qualcomm Innovation Center, Inc. open sources through AI Model Efficiency Toolkit (AIMET)

- What’s next in energy-efficient AI

Questions?

Connect with Us

www.qualcomm.com/ai

www.qualcomm.com/news/onq

@QCOMResearch

https://www.youtube.com/qualcomm?

http://www.slideshare.net/qualcommwirelessevolution

Thank you

Follow us on:

For more information, visit us at:

www.qualcomm.com & www.qualcomm.com/blog

Premier Sponsor

Intelligent Agent
Neuton

Automated TinyML

Zero-code SaaS solution

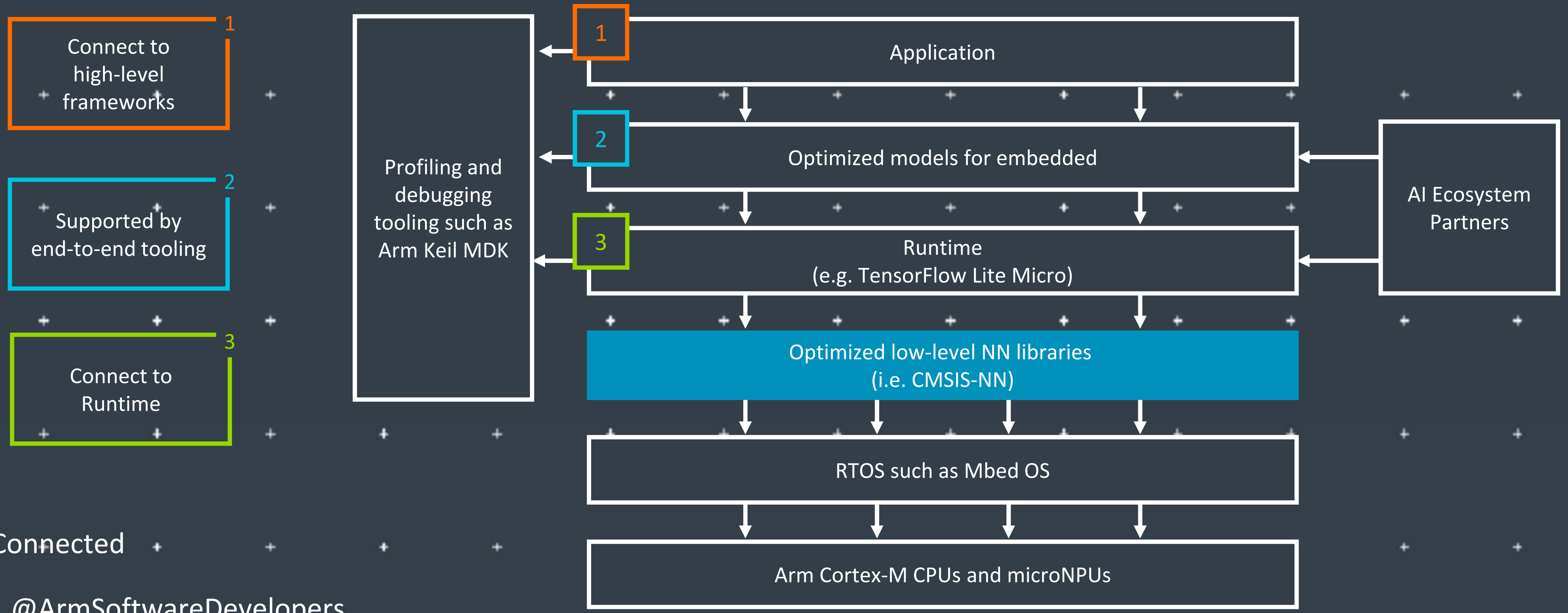
Create tiny models, ready for embedding, in just a few clicks!

Compare the benchmarks of our compact models to those of TensorFlow and other leading neural network frameworks.

Build Fast. Build Once. Never Compromise.

Executive Sponsors

Arm: The Software and Hardware Foundation for tinyML



Stay Connected

@ArmSoftwareDevelopers

@ArmSoftwareDev

Resources: developer.arm.com/solutions/machine-learning-on-arm

TinyML for all developers

C++ library

Arduino library

WebAssembly

Dataset

Acquire valuable training data securely

Enrich data and train ML algorithms

Edge Device

Real sensors in real time

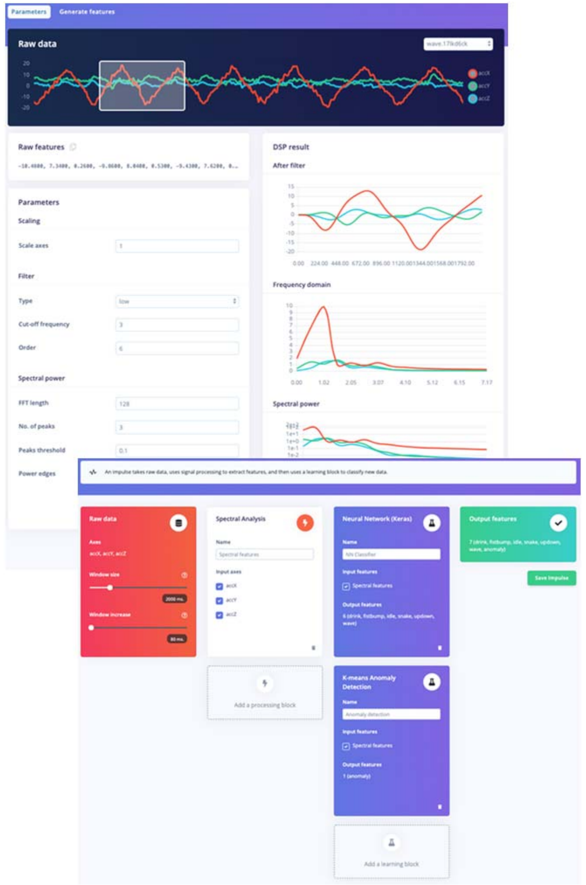
Open source SDK

Impulse

Embedded and edge compute deployment options

Test impulse with real-time device data flows

Test



www.edgeimpulse.com

Qualcomm
AI research

Advancing AI research to make efficient AI ubiquitous

Power efficiency

Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization

Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning

Robust learning through minimal data, unsupervised learning, on-device learning

A platform to scale AI across the industry

Perception

Object detection, speech recognition, contextual fusion

Reasoning

Scene understanding, language understanding, behavior prediction

Action

Reinforcement learning for decision making

Edge cloud

Cloud

IoT/IIoT

Automotive

Mobile

Syntiant Corp. is moving artificial intelligence and machine learning from the cloud to edge devices. Syntiant's chip solutions merge deep learning with semiconductor design to produce ultra-low-power, high performance, deep neural network processors. These network processors enable always-on applications in battery-powered devices, such as smartphones, smart speakers, earbuds, hearing aids, and laptops. Syntiant's Neural Decision Processors™ offer wake word, command word, and event detection in a chip for always-on voice and sensor applications.

Founded in 2017 and headquartered in Irvine, California, the company is backed by Amazon, Applied Materials, Atlantic Bridge Capital, Bosch, Intel Capital, Microsoft, Motorola, and others. Syntiant was recently named a CES® 2021 Best of Innovation Awards Honoree, shipped over 10M units worldwide, and unveiled the NDP120 part of the NDP10x family of inference engines for low-power applications.

www.syntiant.com

@Syntiantcorp

TINY ML EMEA
Virtual
JUNE 7-10,2021

Platinum Sponsors

Part of your life. Part of tomorrow.

www.infineon.com

Reality AI®

Add Advanced Sensing to your Product with Edge AI / TinyML

https://reality.ai info@reality.ai @SensorAI Reality AI

Pre-built Edge AI sensing modules, plus tools to build your own

Reality AI solutions

- Prebuilt sound recognition models for indoor and outdoor use cases
- Solution for industrial anomaly detection
- Pre-built automotive solution that lets cars "see with sound"

Reality AI Tools® software

- Build prototypes, then turn them into real products
- Explain ML models and relate the function to the physics
- Optimize the hardware, including sensor selection and placement

Gold Sponsors

LatentAI

Adaptive AI for the Intelligent Edge

Latentai.com

SensiML™

Build Smart IoT Sensor Devices From Data

SensiML pioneered TinyML software tools that auto generate AI code for the intelligent edge.

- End-to-end AI workflow
- Multi-user auto-labeling of time-series data
- Code transparency and customization at each step in the pipeline

We enable the creation of production-grade smart sensor devices.

sensiml.com

TINY ML EMEA
Virtual
JUNE 7-10,2021

Silver Sponsors

emza
visual sense

GREENWAVES
TECHNOLOGIES

HOTG

imagimob

Qeexo

seeed
studio
The IoT Hardware Enabler

ST
life.augmented

Copyright Notice