

tinyML[®] Summit

Miniature dreams can come true...

March 28-30, 2022 | San Francisco Bay Area



www.tinyML.org

Automating Model Optimization for Efficient Edge AI from practical solutions to open- source toolkit

Hsin-Pai (Dave) Cheng, Senior Deep Learning Researcher
Qualcomm AI Research
Qualcomm Technologies, Inc.



Agenda

- Not all accelerators are equal
Neural Architecture Search and its applications
- Deployment friendly optimization
Automatic quantization
- Optimization tools
AI Model Efficiency Toolkit (AIMET)

The challenge of AI workloads

Constrained mobile environment



Holistic model efficiency research

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

Quantization

Learning to reduce
bit-precision while keeping
desired accuracy

Compilation

Learning to compile
AI models for efficient
hardware execution

Neural architecture search

Learning to design smaller
neural networks that are on par
or outperform hand-designed
architectures on real
hardware

Compression

Learning to prune
model while keeping
desired accuracy

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 phase for every new scenario



High cost

Brute force search is expensive
>40,000 epochs per platform



Do not scale

Repeated training phase for every new device
>40,000 epochs per platform



Unreliable hardware models

Requires differentiable cost-functions
Repeated training phase for every new device

Introducing new AI research

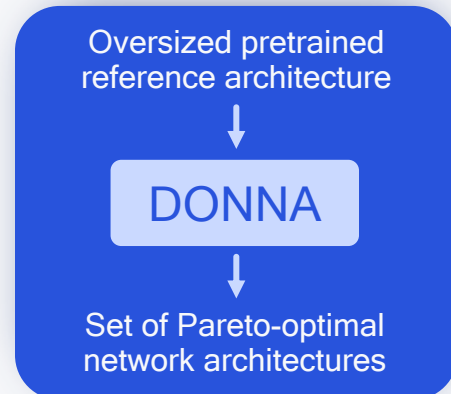
DONNA

Distilling Optimal Neural Network Architectures

Efficient NAS with hardware-aware optimization

A scalable method that finds pareto-optimal network architectures in terms of accuracy and latency for any hardware platform at low cost

Starts from an oversized pretrained reference architecture



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 of 1000-4000 epochs, 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

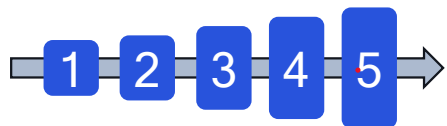
DONNA: Distilling Optimal Neural Network Architectures

Build an accuracy model **once** but **deploy to many scenarios**

A. Define a search space **once**

Define backbone:

- Fixed channels
- Head and Stem

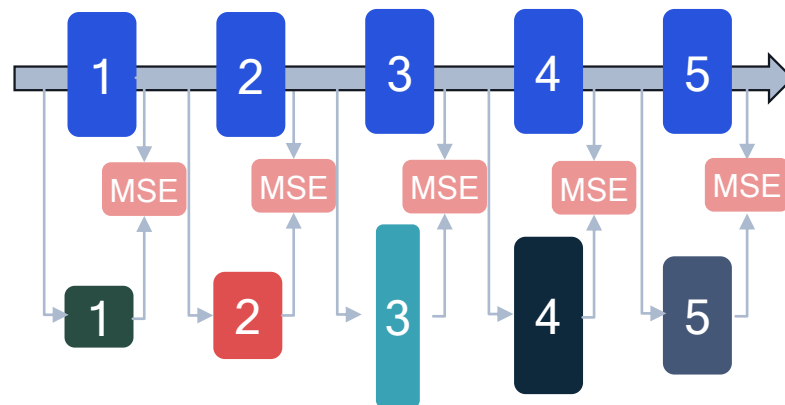


Varying parameters:

- Conv kernel size
- Expansion factor
- Num layers per block
- Layer types
- Num output channels

B. Build an accuracy model via knowledge distillation (KD) **once**

Using KD, train blocks from the search space using features from a reference model



Use the block losses to build an accuracy predictor for end-to-end architectures



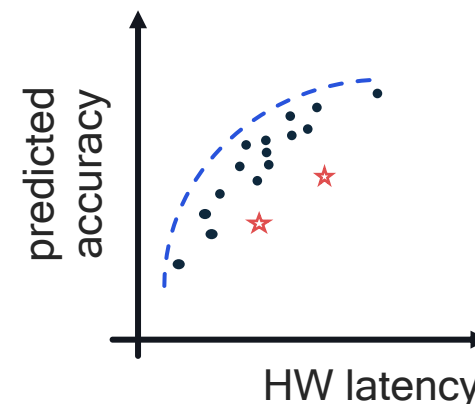
Accuracy Predictor for every architecture

C. Evolutionary search in **24h**

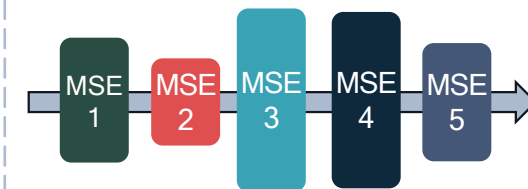


+ different compiler versions,
different image sizes

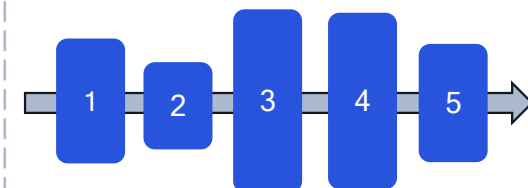
scenario-specific search



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**



Define reference architecture and search-space once

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

Select reference architecture

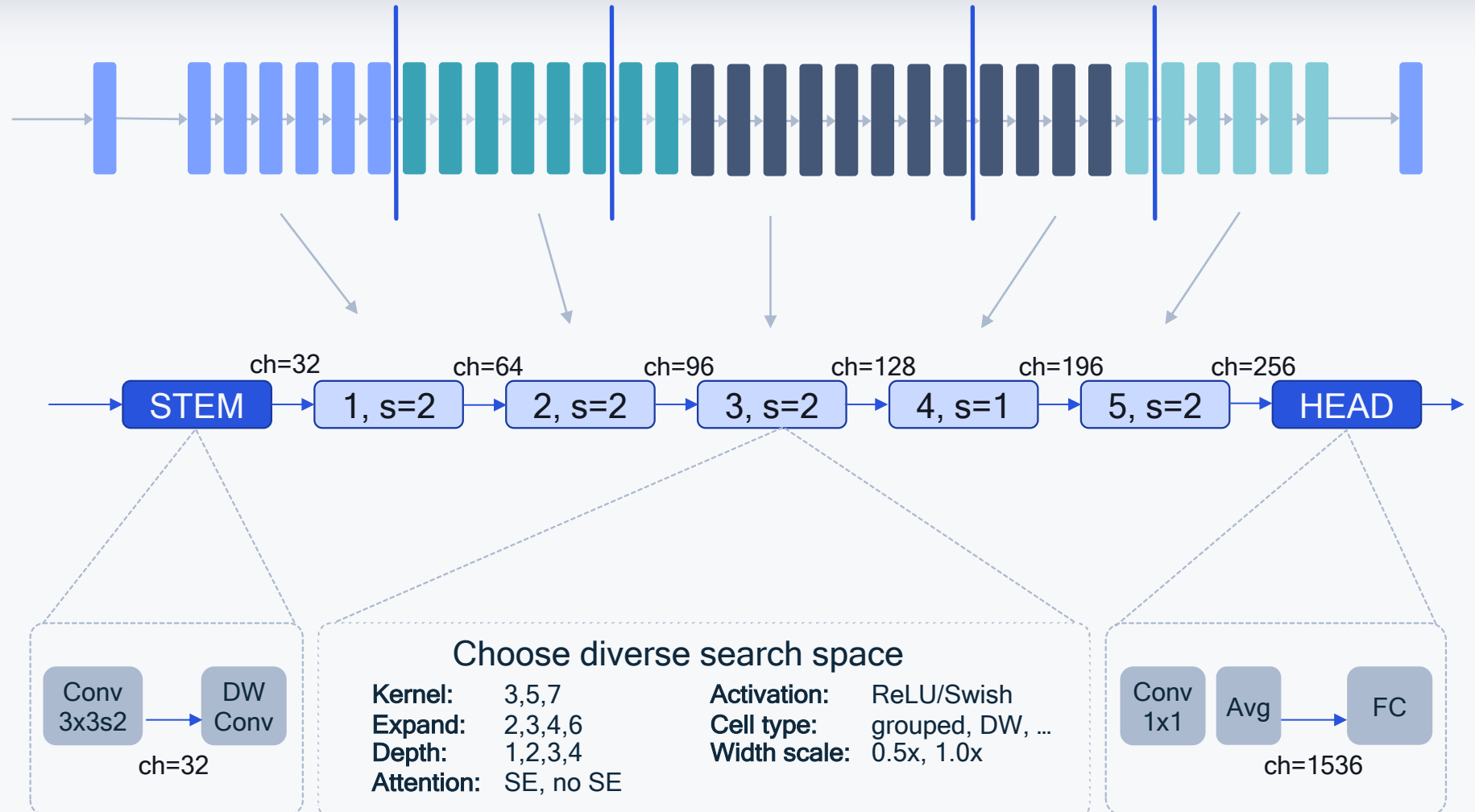
The largest model in the search-space

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 kernel-size, expansion-rate, depth, # channels, cell-type, activation, attention



Build accuracy predictor via BKD once

Low-cost hardware-agnostic training phase

Block library

Pretrain all blocks in search-space through blockwise knowledge distillation

Block pretrained weights



Block quality metrics



Fast block training
Trivial parallelized training
Broad search space

Architecture library

Quickly finetune a representative set of architectures

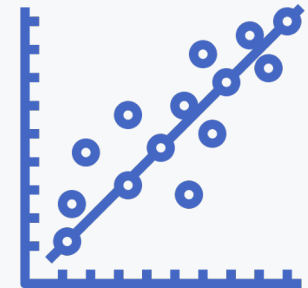


Finetuned architectures

Finetune sampled networks
Fast network training
Only 30-50 NN required

Accuracy predictor

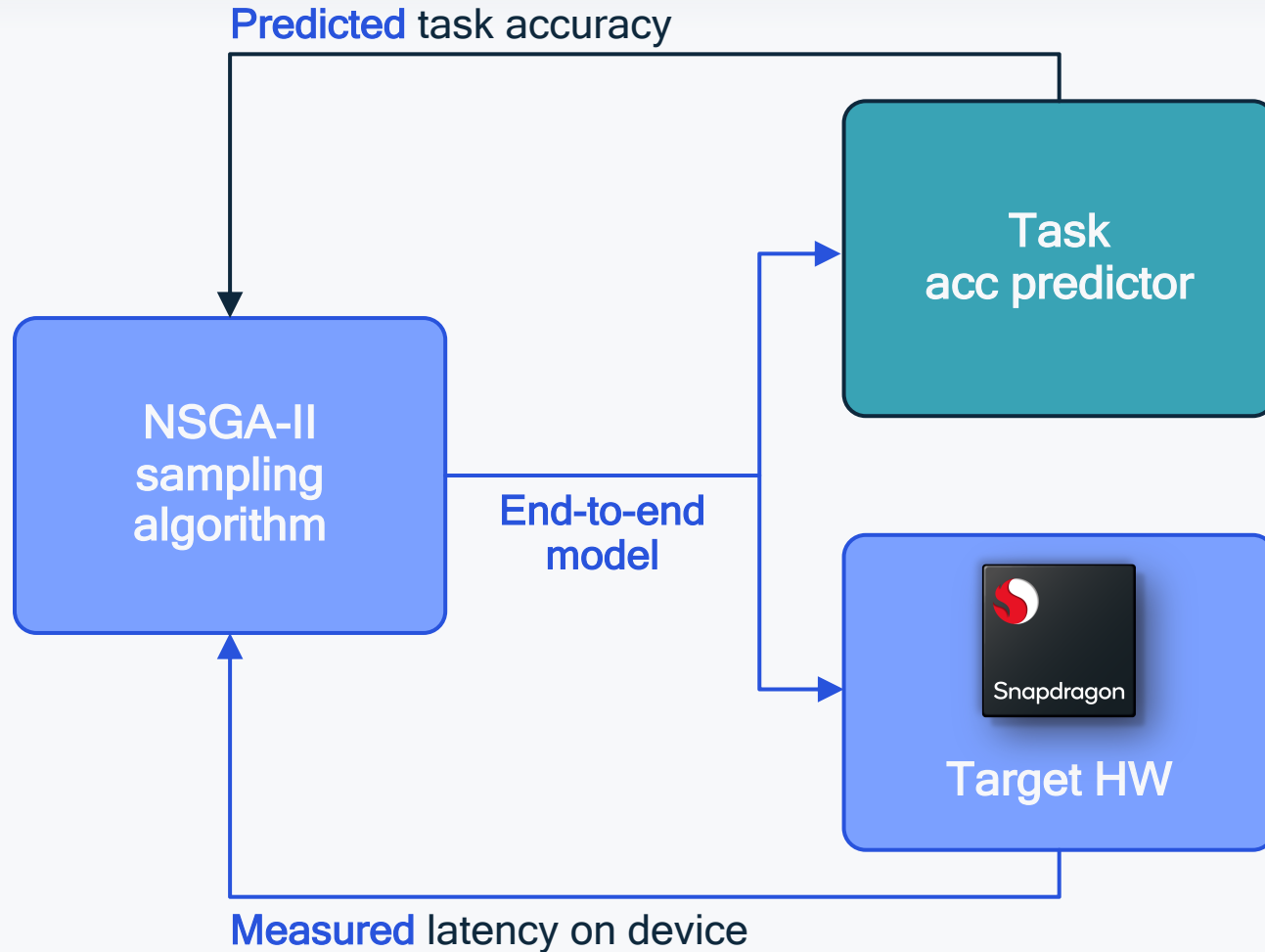
Fit linear regression model



Regularized Ridge Regression
Accurate predictions

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

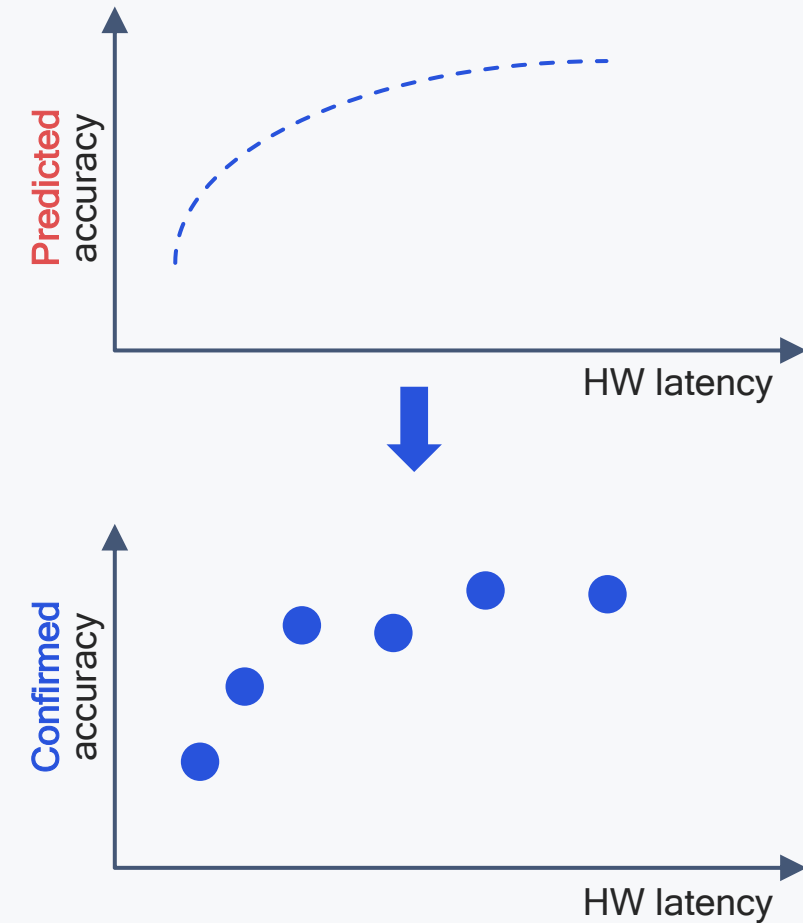
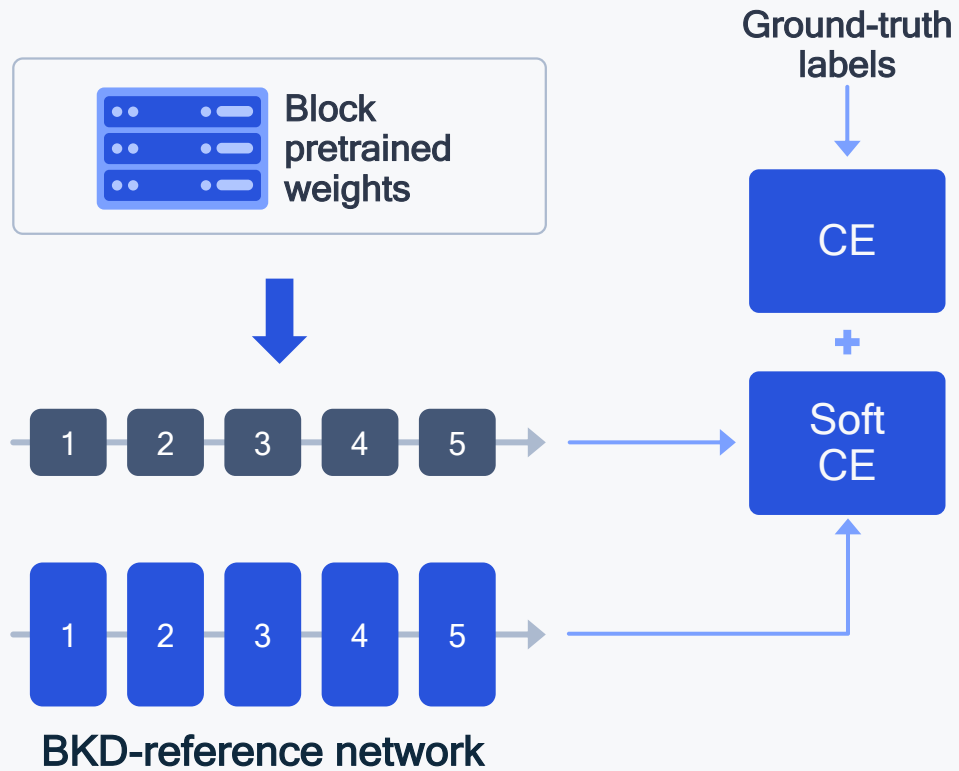
Accurate scenario-specific search

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

Quickly finetune predicted Pareto-optimal architectures

Finetune to reach full accuracy and complete hardware-aware optimization for on-device AI deployments

Soft distillation on teacher logits



DONNA efficiently finds optimal models over diverse scenarios

Cost of training
is a handful of
architectures*

Method	Granularity	Macro-diversity	Search-cost 1 scenario [epochs]	Cost / scenario 4 scenarios [epochs]	Cost / scenario ∞ scenarios [epochs]
OFA	Layer-level	Fixed	1200+10×[25 – 75]	550 – 1050	250 – 750
DNA	Layer-level	Fixed	770+10×450	4700	4500
MNasNet	Block-level	Variable	40000+10×450	44500	44500
DONNA	Block-level	Variable	4000+10×50	1500	500

Good OK Not good

DONNA provides MnasNet-level diversity at 100x lower cost

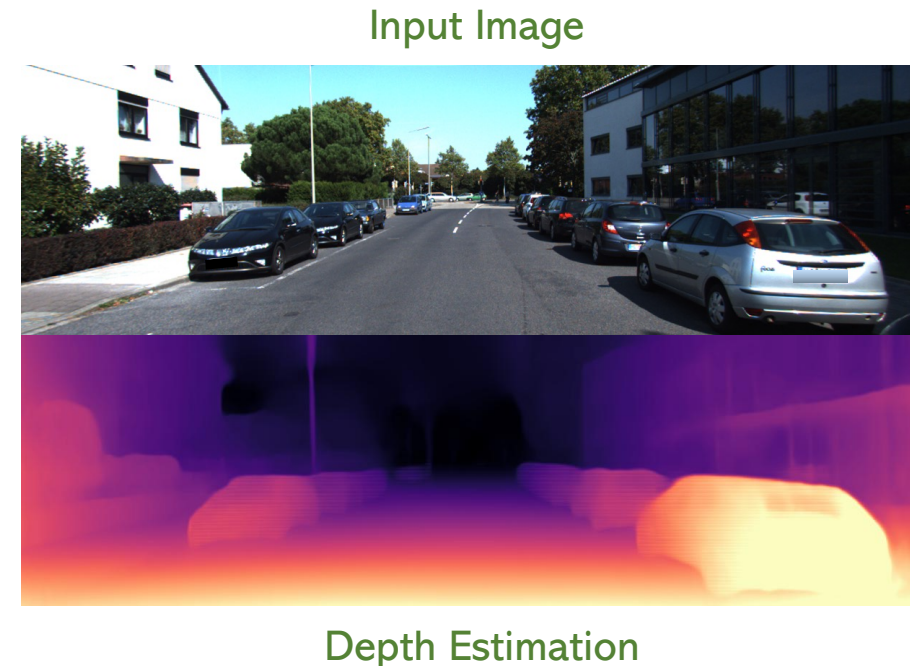
DONNA can be easily adapted to different tasks

Task	Reference Model	Reference model Accuracy	Donna Optimized Model Accuracy	Donna Gain (Latency Reduction)
Classification	EfficientDet-B0	77.7%	77.1%	25%
Object Detection	EfficientDet-D0	34.4	34.2	37%
Depth Estimation	Monodepth + X-distill	0.69	0.75	35%
Image Denoising	MPRNet	39.4dB	39.2dB	45%

20% - 40% Latency Reduction

Application on Depth Estimation

- **Accurate depth estimation is key for 3D understanding**
 - Autonomous driving, AR/VR, image/video processing, robotics
 - Recent years, learning-based methods have greatly advanced SOTA
 - However, high-quality, dense GT depth annotations are costly to collect
- **Self-supervised monocular depth**
 - Utilizes geometric relationship across video frames
 - Learns depth from unlabeled monocular videos
 - However, SOTA models are computationally heavy
- **Computation efficiency and low latency are critical**
 - Deployment on resource-constrained mobile platforms, e.g., headsets, smartphones
 - Real-time performance crucial for practical applications



Real-time Depth Estimation

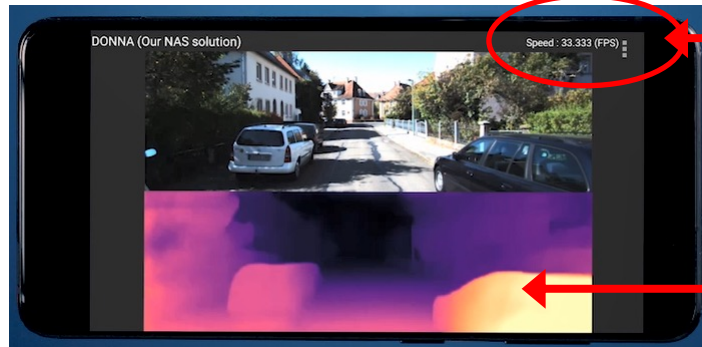
Depth estimation running on Snapdragon® powered smartphone

- **X-Distill (Self-supervision): Improves Accuracy**
 - Has significantly **smaller estimation errors** comparing to baseline
- **X-Distill + DONNA: Real-time Performance**
 - Leads to significantly **smaller model** and **real-time inference**
 - Has considerably **smaller estimation errors** comparing to baseline

Showing efficiency improvement with better accuracy



Baseline:
Monodepth2



Our solution has
~40% higher FPS

Ours:
X-Distill+DONNA

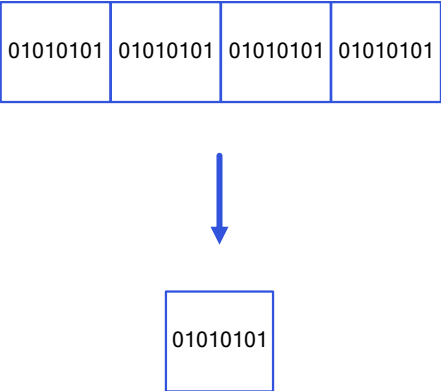
Quantization

Taking floating point trained models to target for efficient fixed-point inference

Quantizing AI models offers significant benefits

Memory usage

8-bit versus 32-bit weights and activations stored in memory



Power consumption

Significant reduction in energy for both computations and memory access

Add energy (pJ)	
INT8	FP32
0.03	0.9
30X energy reduction	
Mult energy (pJ)	
INT8	FP32
0.2	3.7
18.5X energy reduction	

Mem access energy (pJ)	
Cache (64-bit)	
8KB	10
32KB	20
1MB	100
DRAM	1300-2600
Up to 4X energy reduction	

Latency

With less memory access and simpler computations, latency can be reduced



Silicon area

Integer math or less bits require less silicon area compared to floating point math and more bits

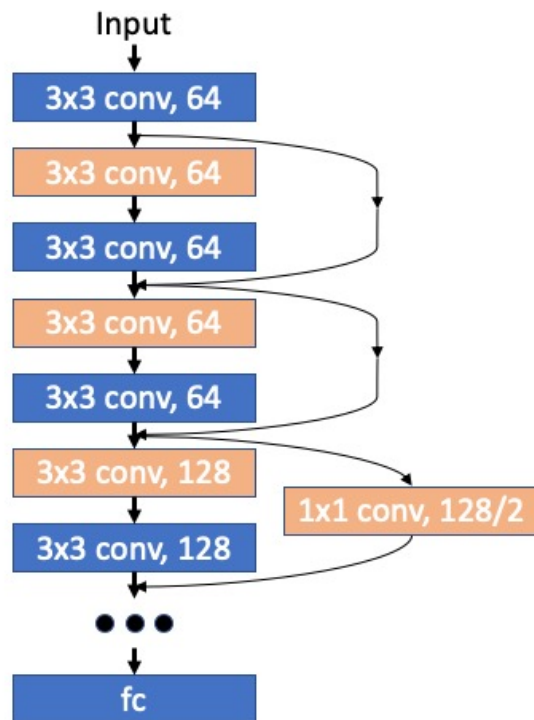
Add area (μm²)	
INT8	FP32
36	4184
116X area reduction	

Mult area (μm²)	
INT8	FP32
282	7700
27X area reduction	

What is neural network quantization?

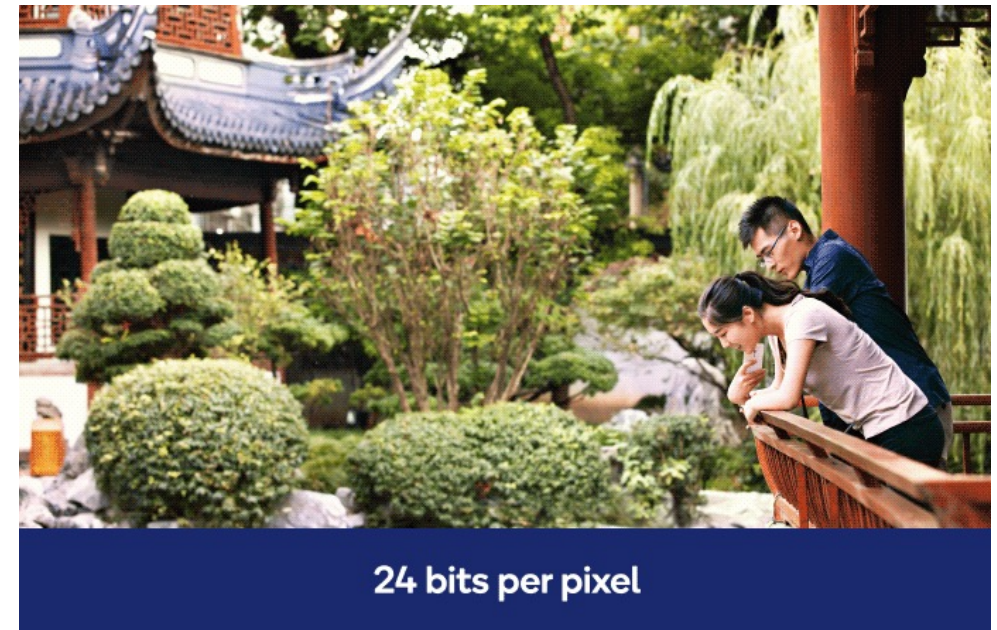
For any given trained neural network:

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



Quantization analogy

Similar to representing the pixels of an image with less bits



Challenge of Quantization

Quantization noise can reduce model accuracy

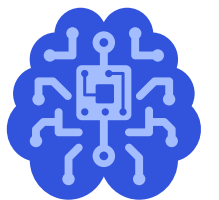


INT8 Baseline:
Inaccurate
segmentation



AIMET & AIMET Model Zoo

Open-source projects to scale model-efficient AI to the masses



AIMET

Providing advanced model efficiency features and benefits

Benefits



Lower power



Lower storage



Lower memory bandwidth



Higher performance



Maintains model accuracy



Simple ease of use

Quantization

State-of-the-art INT8 and INT4 performance

Post-training quantization methods, including Data-Free Quantization, *Adaptive Rounding (AdaRound)* & *AutoQuant*

Quantization-aware training (QAT)

Quantization simulation

Compression

Efficient tensor decomposition and removal of redundant channels in convolution layers

Spatial singular value decomposition (SVD)

Channel pruning

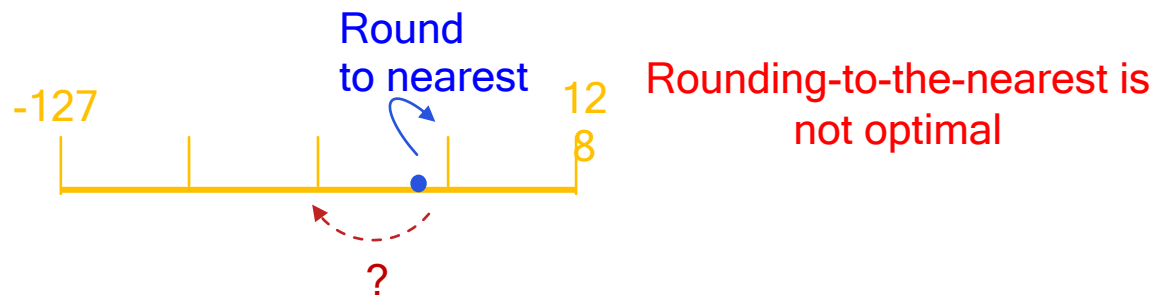
Visualization

Analysis tools for drawing insights for quantization and compression

Weight ranges

Per-layer compression sensitivity

AdaRound

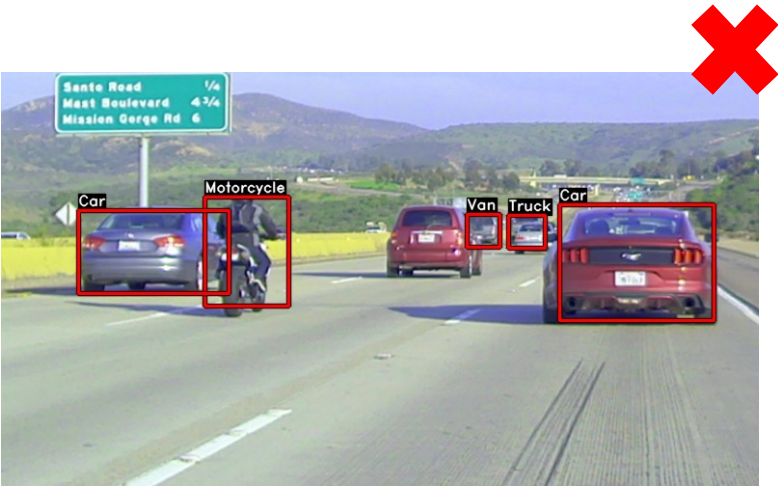


- AdaRound optimizes the network weights in minutes without model fine-tuning

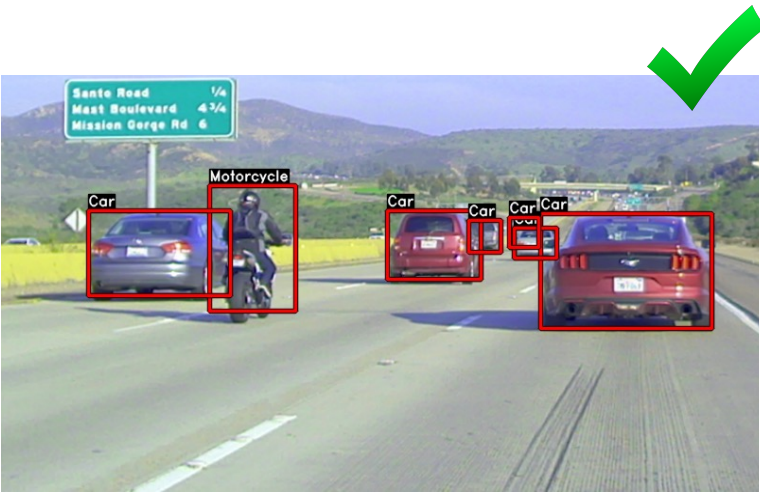
$$\arg \min_{\mathbf{V}} \left\| \mathbf{W}\mathbf{x} - \widetilde{\mathbf{W}}\mathbf{x} \right\|_F^2 + \lambda f_{reg}(\mathbf{V})$$

Object Detection

Configuration	mAP
Floating point	82.20
Nearest Rounding - 8-bit weights, 8-bit activations	49.85
AdaRound - 8-bit weights, 8-bit activations	81.21



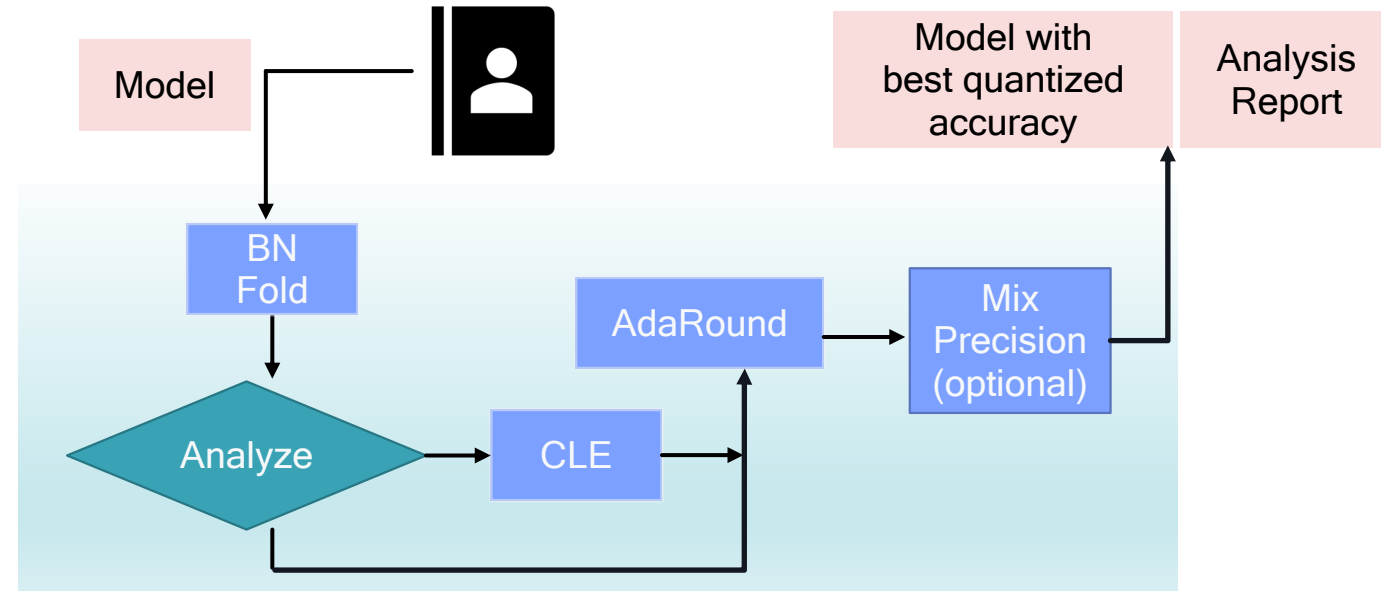
INT8, Baseline (Nearest Rounding)



INT8, AdaRound

AutoQuant

- *Making Post-Training Quantization (PTQ) Easy*
- Different models require different PTQ techniques
- Need to make provide push-button solution to users that navigates different options and provides best answer
- AutoQuant: Blackbox, push-button PTQ
 - ☐ Analyzes the model
 - ☐ Applies the best sequence of PTQ features
 - ☐ Returns the best possible accuracy model (withing PTQ constraints) together with analysis

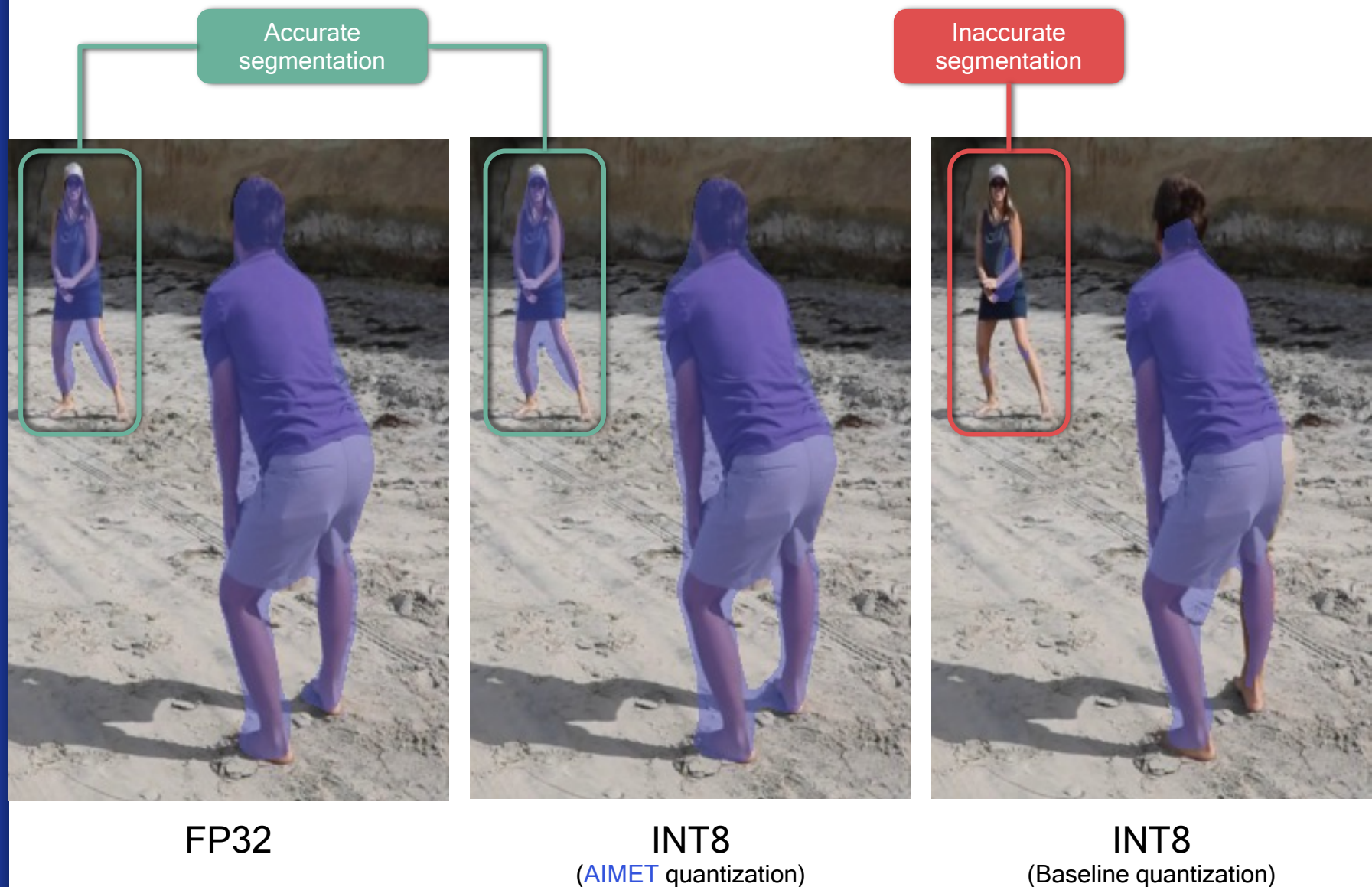


Quantization using AIMET preserves accuracy

Visual difference in model accuracy is telling between AIMET and baseline quantization methods

For DeepLabv3+ semantic segmentation, AIMET quantization maintains accuracy, while baseline quantization method is inaccurate

Baseline quantization: Post-training quantization using min-max based quantization grid
AIMET quantization: Model fine-tuned using Quantization Aware Training in AIMET



AIMET Model Zoo includes popular quantized AI models

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

 TensorFlow

<1%
Loss in
accuracy*

 PyTorch

75.21% 74.96%
FP32 INT8

Top-1 accuracy*

ResNet-50
(v1)

75% 74.21%
FP32 INT8

Top-1 accuracy*

MobileNet-
v2-1.4

74.93% 74.99%
FP32 INT8

Top-1 accuracy*

EfficientNet
Lite

0.2469 0.2456
FP32 INT8

mAP*

SSD
MobileNet-v2

0.35 0.349
FP32 INT8

mAP*

RetinaNet

0.383 0.379
FP32 INT8

mAP*

Pose
estimation

25.45 24.78
FP32 INT8

PSNR*

SRGAN

71.67% 71.14%
FP32 INT8

Top-1 accuracy*

MobileNetV2

75.42% 74.44%
FP32 INT8

Top-1 accuracy*

EfficientNet-
lite0

72.62% 72.22%
FP32 INT8

mIoU*

DeepLabV3+

68.7% 68.6%
FP32 INT8

mAP*

MobileNetV2-
SSD-Lite

0.364 0.359
FP32 INT8

mAP*

Pose
estimation

25.51 25.5
FP32 INT8

PSNR

SRGAN

9.92% 10.22%
FP32 INT8

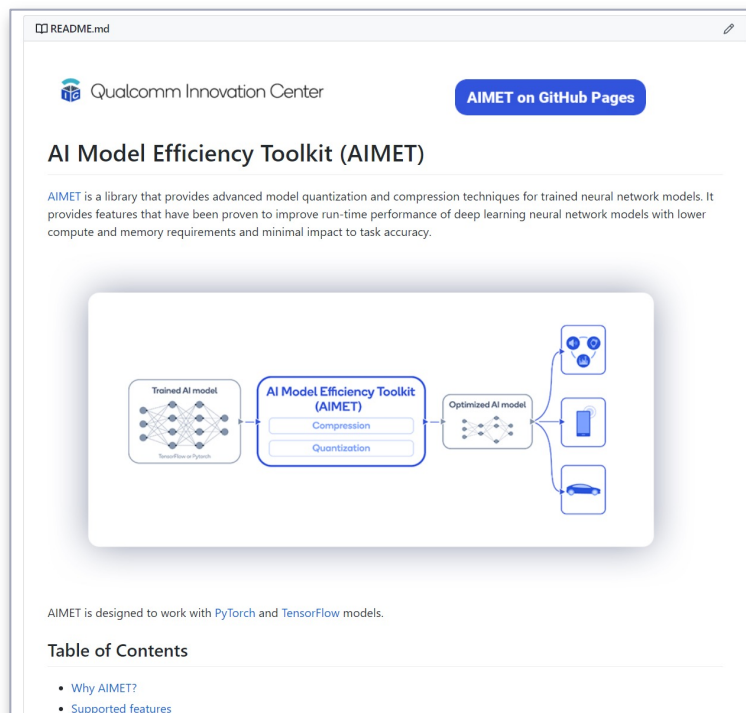
WER*

DeepSpeech2

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

AIMET

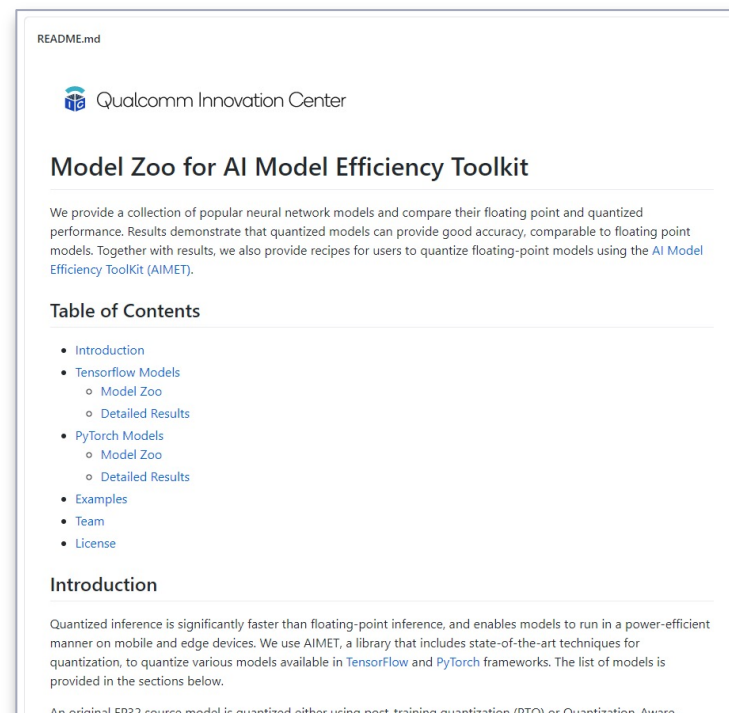
State-of-the-art quantization and compression techniques



github.com/quic/aimet

AIMET Model Zoo

Accurate pre-trained 8-bit quantized models



github.com/quic/aimet-model-zoo

Join our open-source projects



Automating deep neural network design and deployment is crucial for on-device machine learning

We are conducting leading research and development in AI model efficiency while maintaining accuracy

Our open-source projects, based on this leading research, are making it possible for the industry to adopt efficient AI models at scale



Thank you



Follow us on: [f](#) [t](#) [in](#) [@](#) [v](#)

For more information, visit us at:

qualcomm.com & qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

©2018-2022 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to "Qualcomm" may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.



AONdevices

arm

ASPINITY

brainchip
The Neuromorphic Computing Company

CEVA®

Deeplite

EDGE IMPULSE

emza
visual sense

FotaHub

GREENWAVES
TECHNOLOGIES

Grovetly Inc.

Himax

HOTC

imagimob

infineon

itemis

KLIKA·TECH
GLOBAL IOT SOLUTIONS

LatentAI

LATTICE
SEMICONDUCTOR

Micro.ai

OmniML

NXP

POI

Plumerai

PROPHESSEE

Qeexo

Qualcomm

Rackner

RealityAI®
Engineering Solutions for the Edge

REEXEN
technology

RENESAS

SAP

seeed
The IoT Hardware Enabler

SensiML

Sony Semiconductor
Solutions
Corporation

ST
life.augmented

SA STREAM ANALYZE

synaptics®

SynSense

SYNTIANT

Tensil.ai

TensorFlow

XMOS



Copyright Notice

This presentation in this publication was presented as a tinyML® Summit 2022. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org