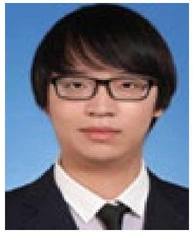
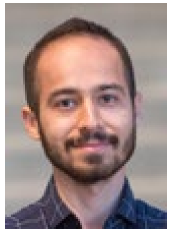
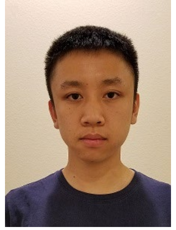
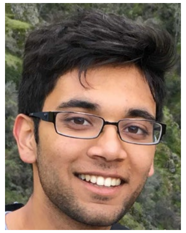
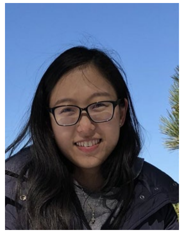
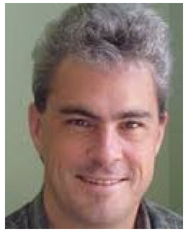
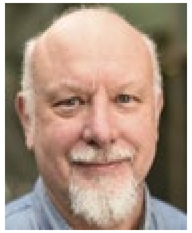
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Efficient Deep Learning through Quantization and Co-Design

Zhen Dong, Zhewei Yao, Amir Gholami, Zhangcheng Zheng, Eric Tan, Daiyaan Arfeen,
Sheng Shen, Qijing Huang, Michael Mahoney, Kurt Keutzer

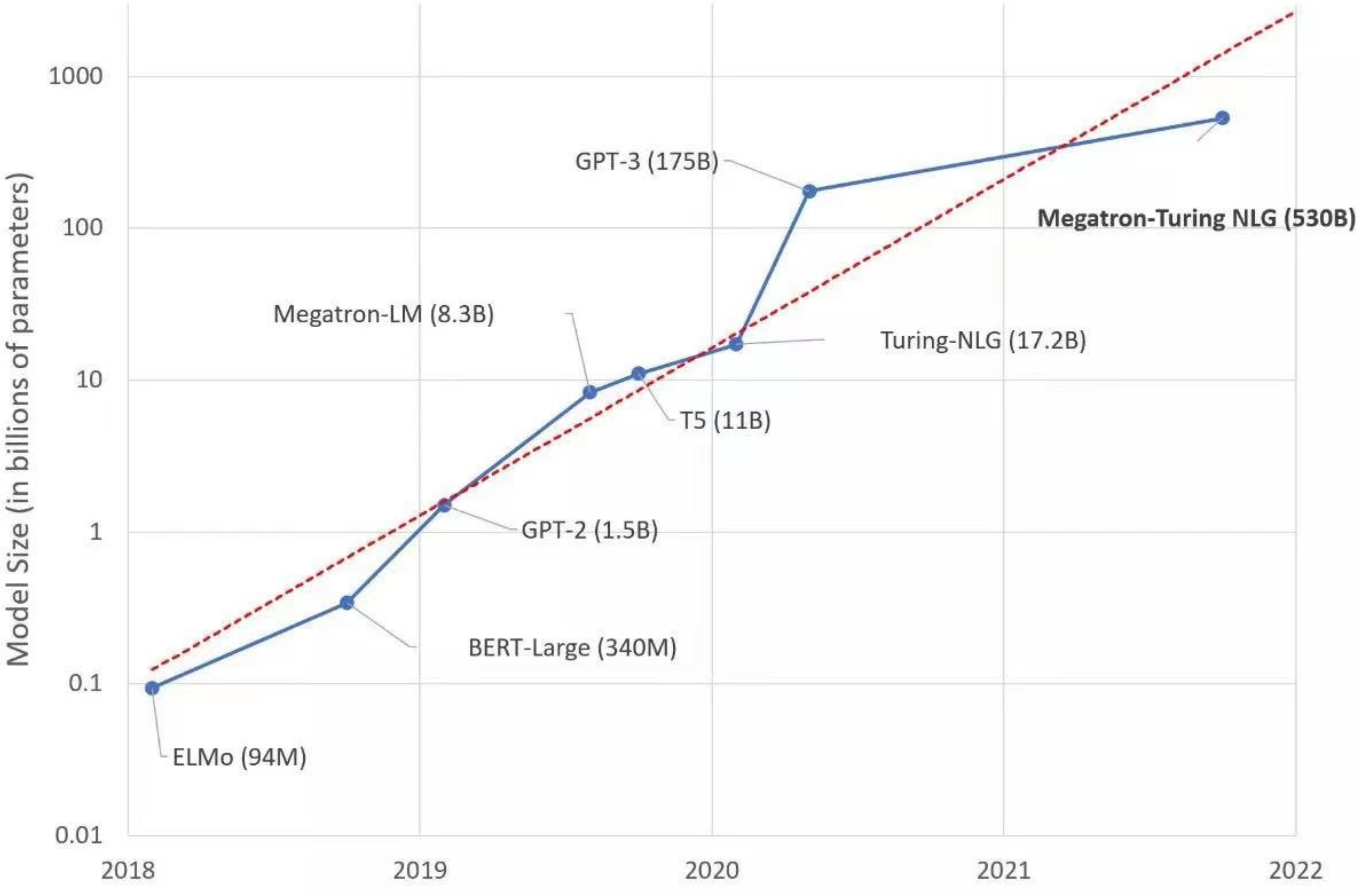




Berkeley
UNIVERSITY OF CALIFORNIA

Outline

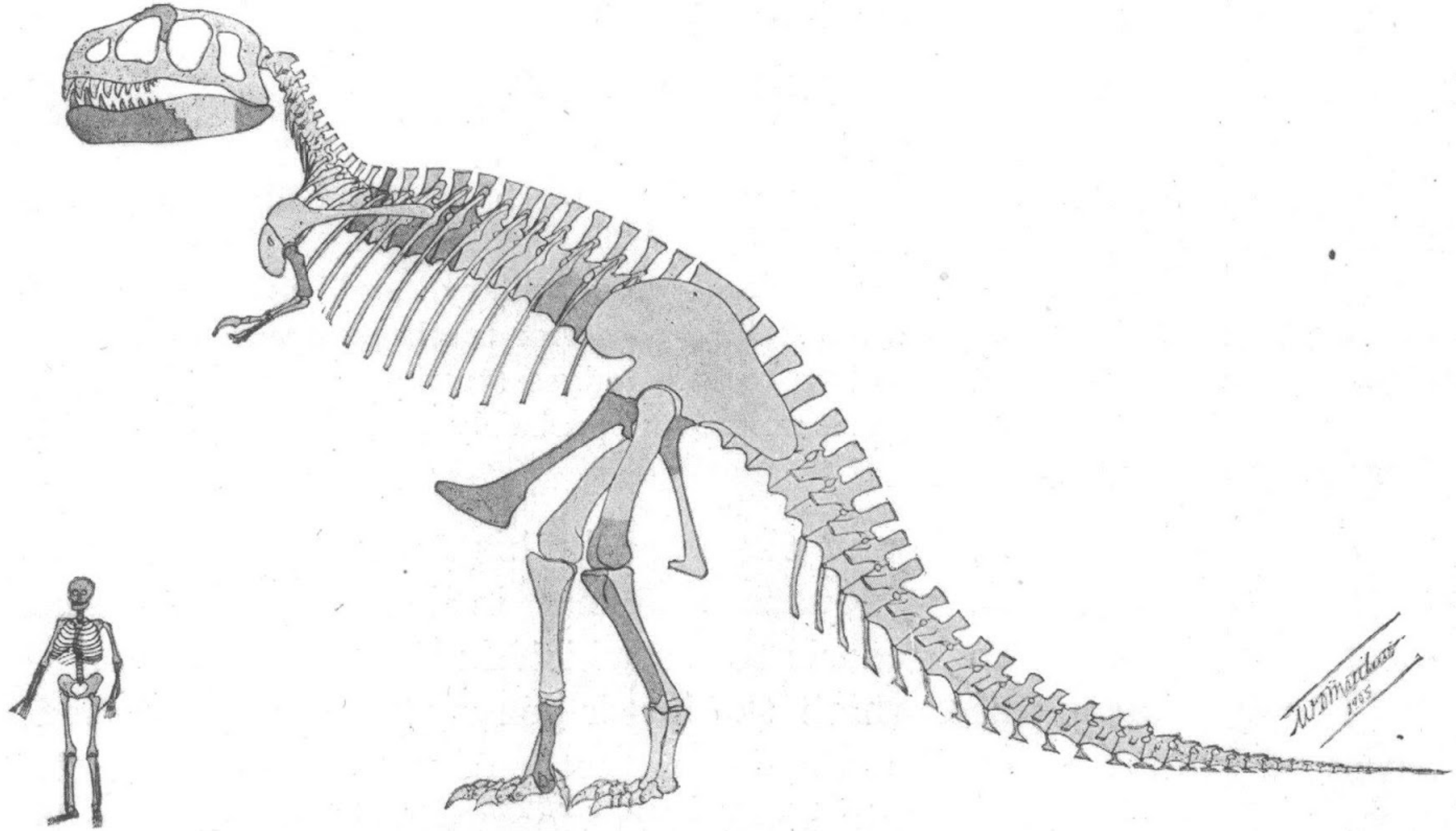
- Introduction
- Hessian-AWare Quantization (HAWQ)
- Hardware-aware Deployment
- Hardware-software Co-design
- Conclusion

Model Size and Computation are Increasing



[1] Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model.

Model Size and Computation are Increasing

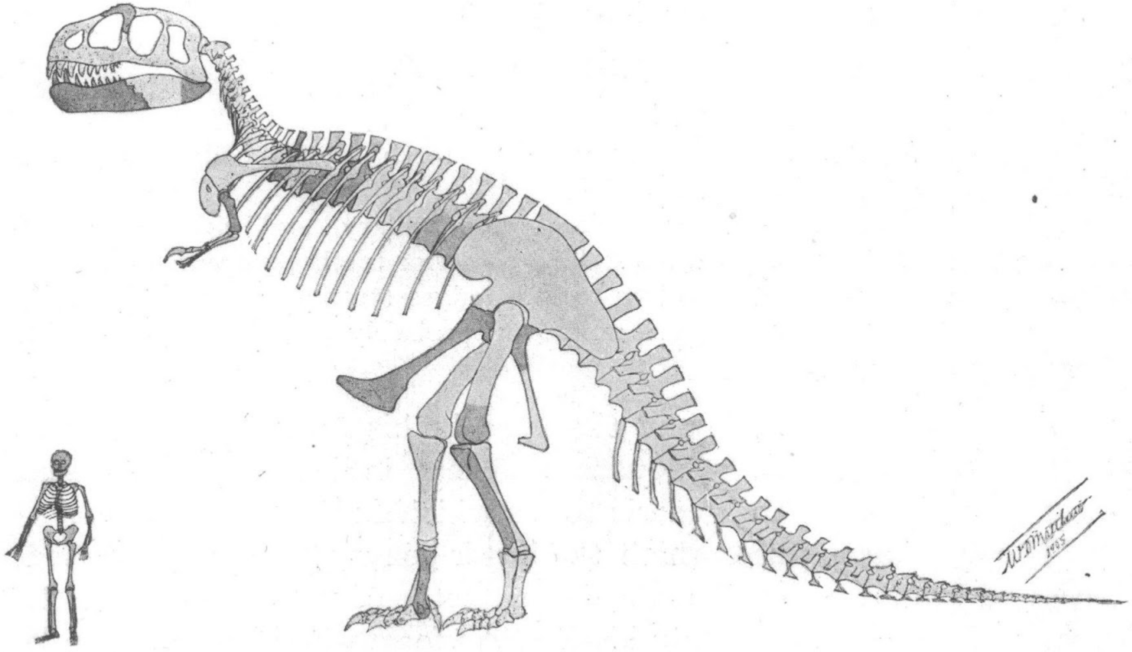


GPT-2
1.5B Parameters

GPT-3
175B Parameters

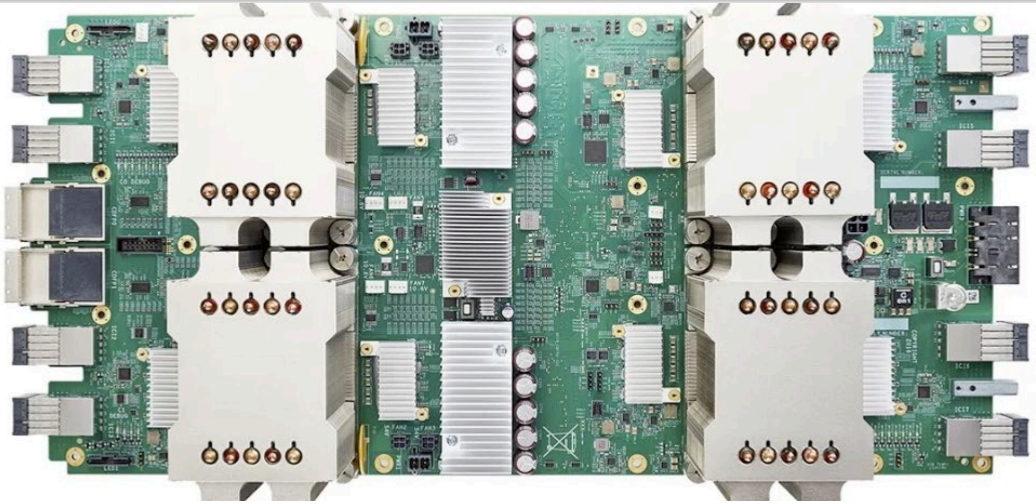
[1] Image from https://blog.exxactcorp.com/what-can-you-do-with-the-openai-gpt-3-language-model/.

Model Size and Computation are Increasing



GPT-2
1.5B Parameters

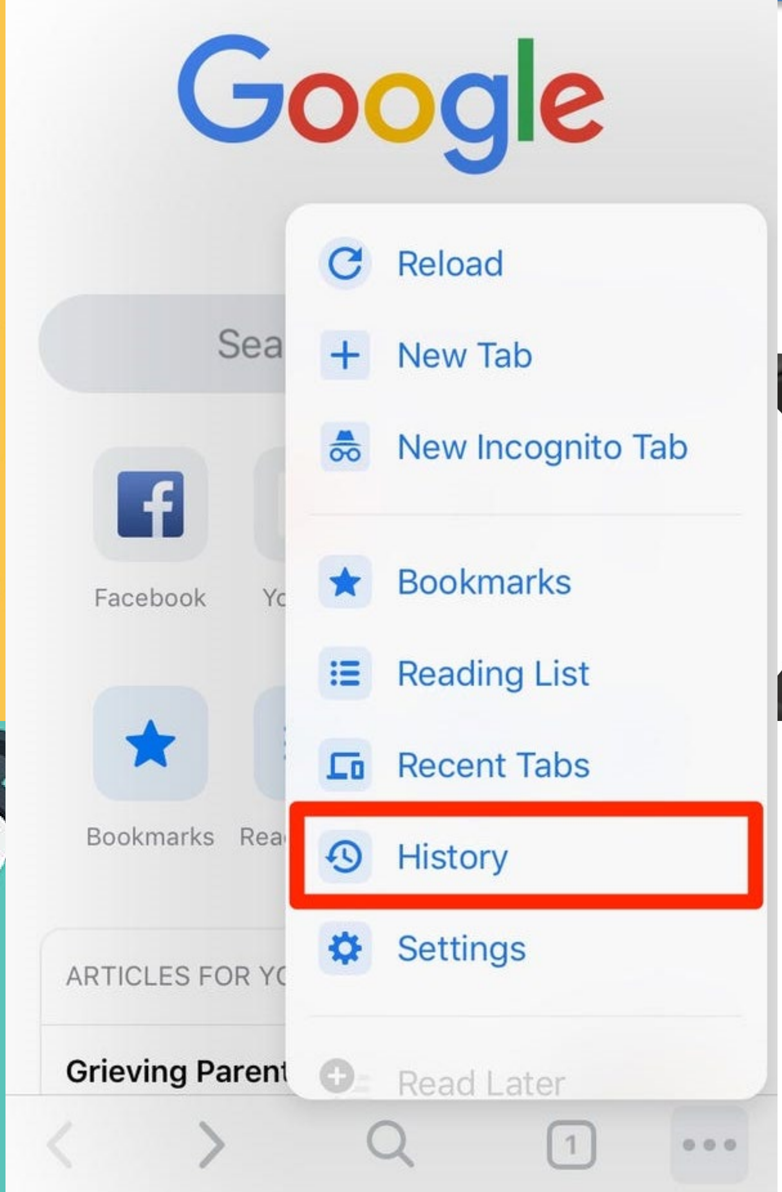
GPT-3
175B Parameters



[1] Image from https://cloud.google.com/tpu.

Why Edge Computing? Privacy

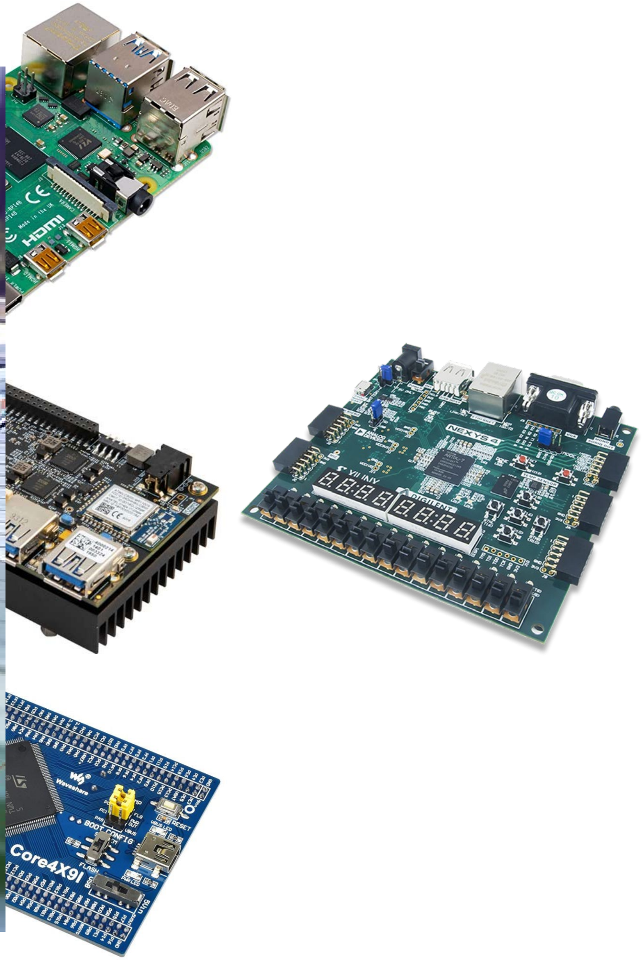
GPT-2
1.5B Parameters



[1] Image from the Internet.

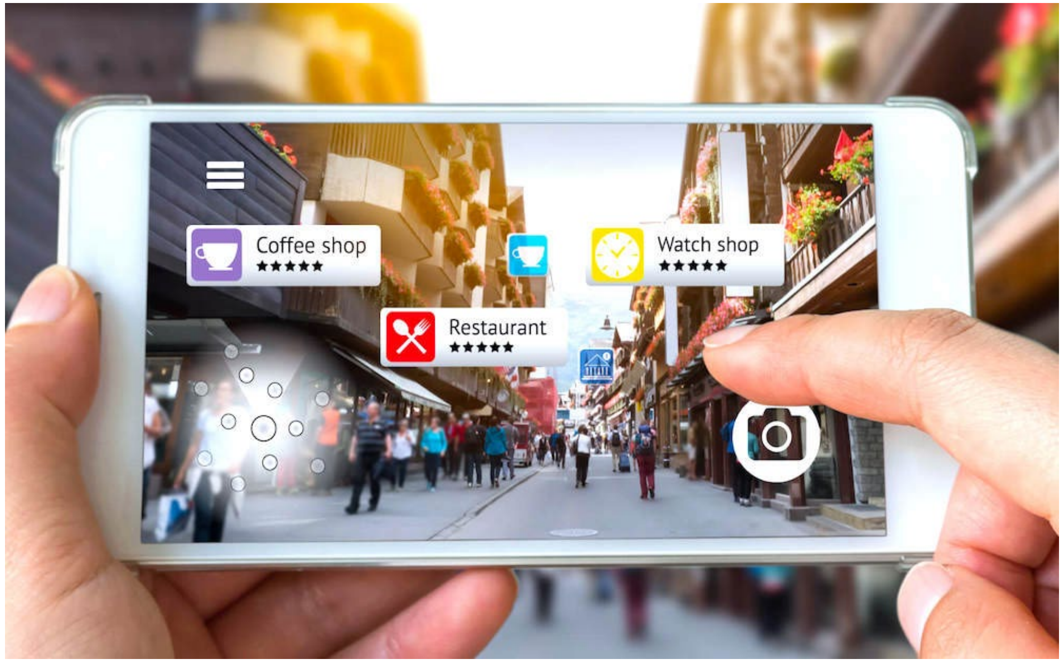
Why Edge Computing? Power

GPT-2
1.5B Parameters



[1] Image from the Internet.

Why Edge Computing? Latency



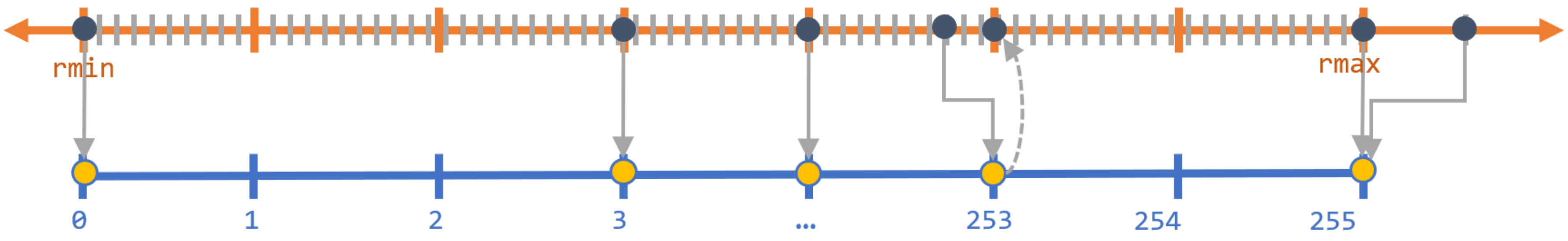
[1] Image from the Internet.

Quantization: Small Size and Fast Computation

- r (FP32): real value in a tensor
- r_{max}, r_{min} : max/min of values
- B: Quantization Bit-width
- S (FP32): Scaling Factor
- z (FP32): Zero Point Shift
- q (INT8): Quantized Values

$$S = \frac{r_{max} - r_{min}}{2^B - 1}$$

$$q = \frac{r - z}{S}$$

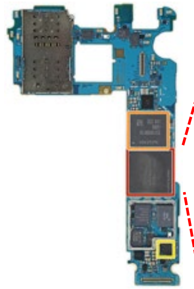


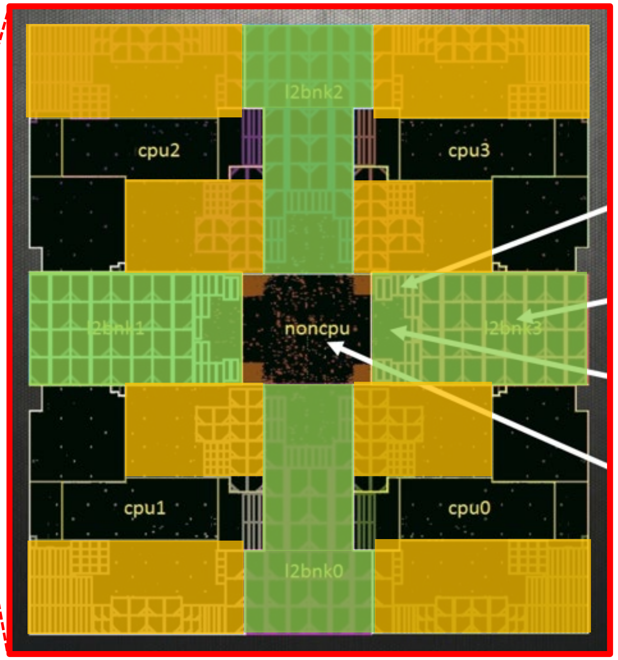
Uniform 8-bit Quantization

[1] Illustration from Sahni Manas.

Quantization: Low Power Consumption

Galaxy S7



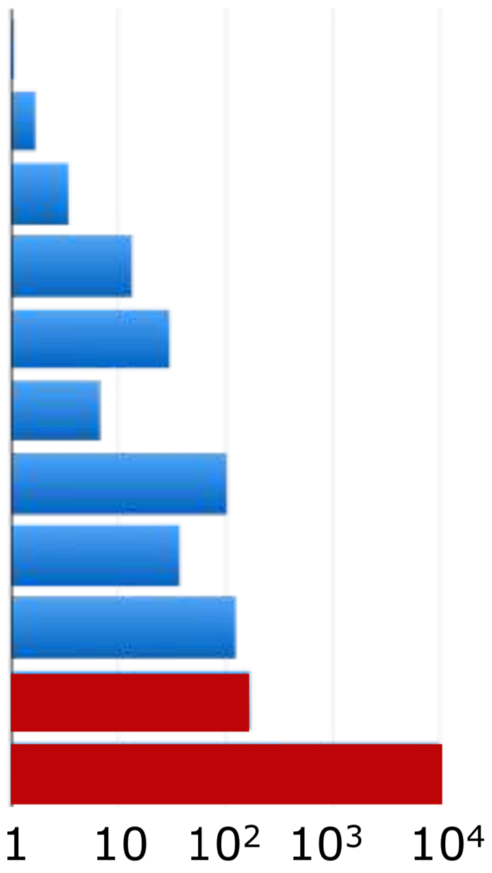


L1 Cache/TLB

L2 Cache

Operation:	Energy (pJ)
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

Relative Energy Cost

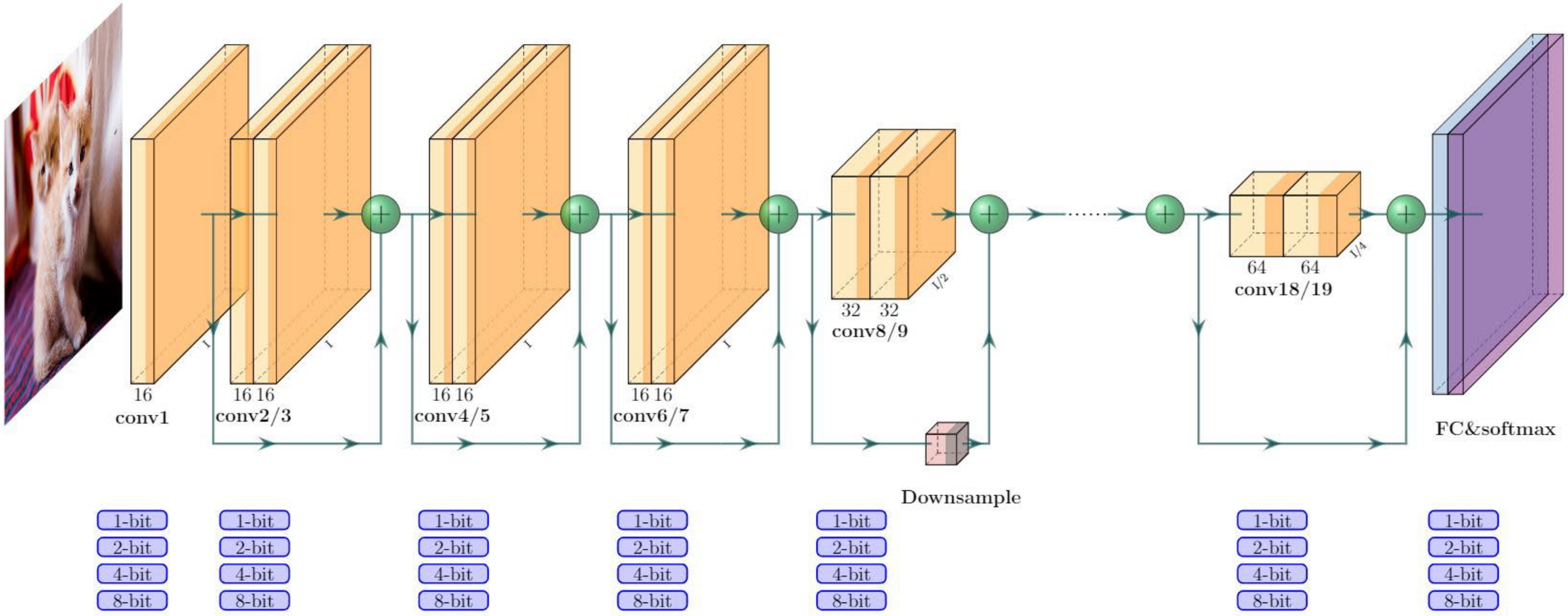


[**Horowitz**, *ISSCC* 2014]

Outline

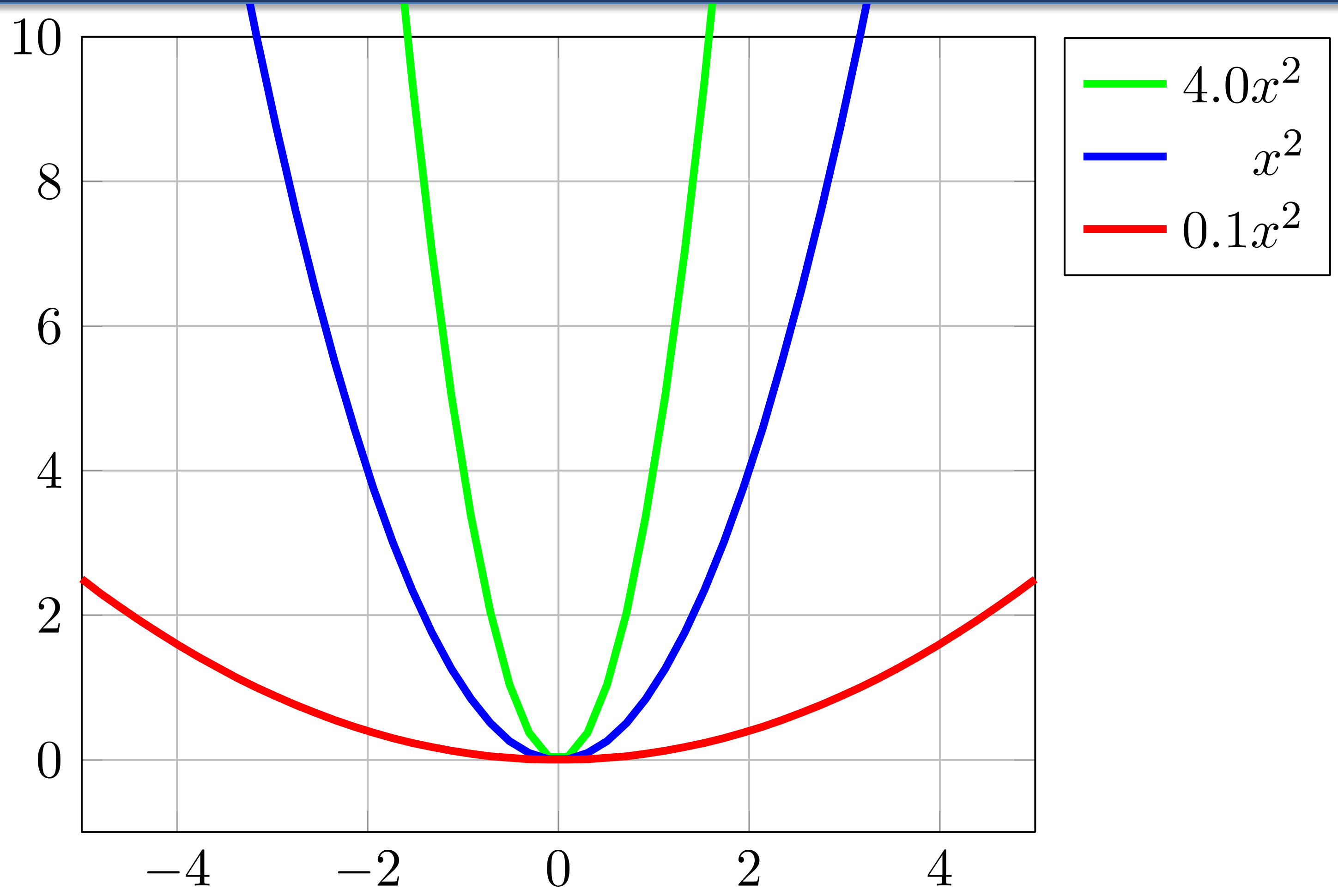
- Introduction
- Hessian-AWare Quantization (HAWQ)
- Hardware-aware Deployment
- Hardware-software Co-design
- Conclusion

Mixed-Precision Quantization: Exponential Search Space



Which mixed-precision setting works better?

Second Order Sensitivity Analysis

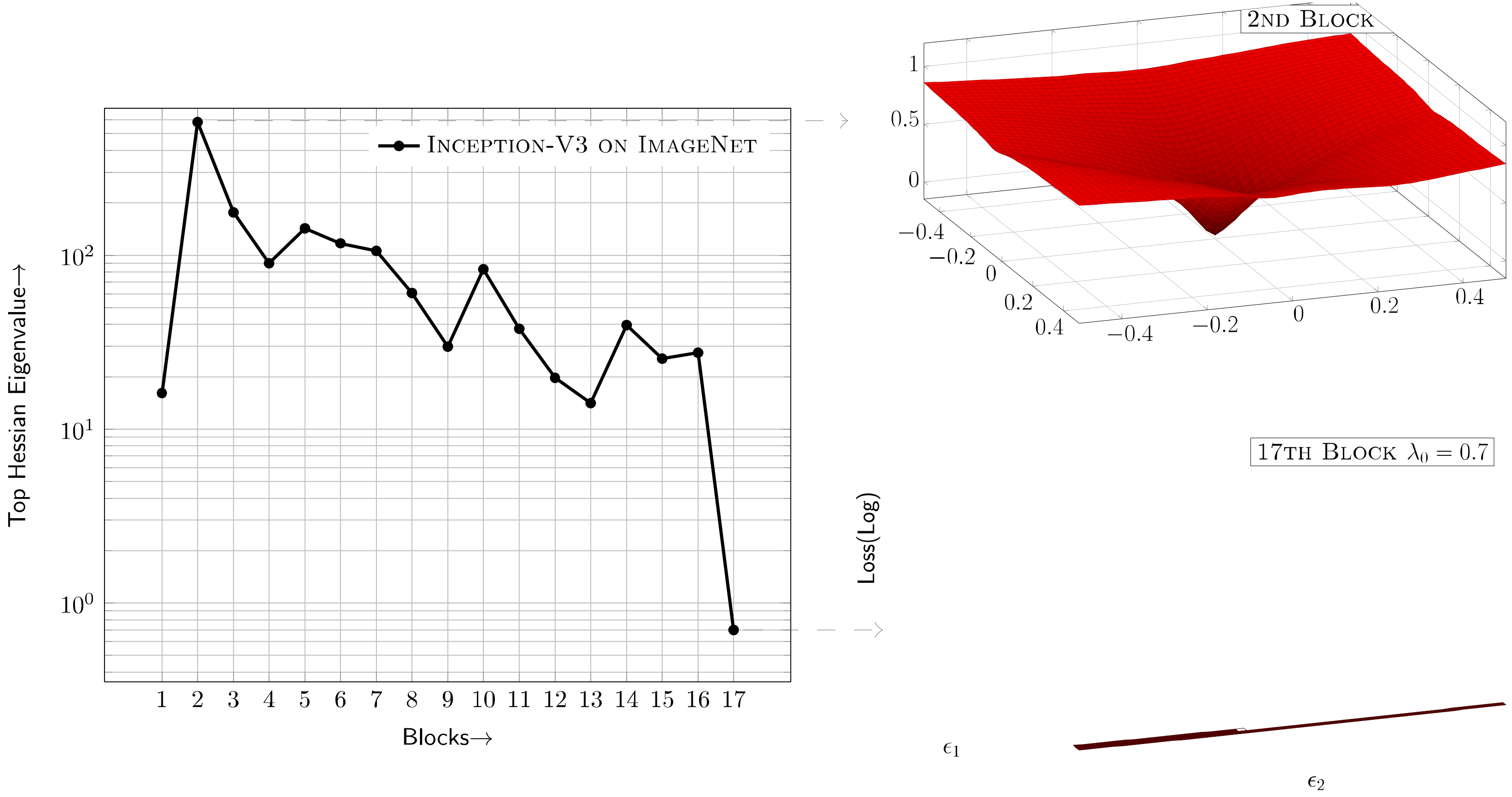


At the origin, the first derivative of y = $4x^2$, y =x^2 , y =0.1 x^2 is all the same: 0

The **second derivative** give more information: 8 , 2, and 0.2 respectively.

HAWQ: Hessian-AWare Quantization

Only quantize layers that have **small top eigenvalue** to **ultra-low precision**



[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV'19

HAWQ: ResNet50 on ImageNet

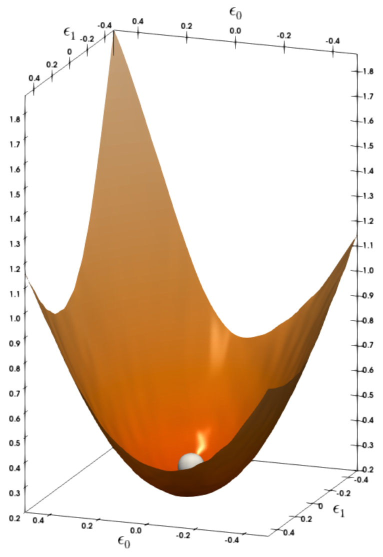
Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	77.39	1.00×	97.8
Dorefa [43]	2	2	67.10	16.00×	6.11
Dorefa [43]	3	3	69.90	10.67×	9.17
PACT [2]	2	2	72.20	16.00×	6.11
PACT [2]	3	3	75.30	10.67×	9.17
LQ-Nets [40]	3	3	74.20	10.67×	9.17
Deep Comp. [8]	3	MP	75.10	10.41×	9.36
HAQ [35]	MP	MP	75.30	10.57×	9.22
HAWQ	2 MP	4 MP	**75.48**	12.28×	**7.96**

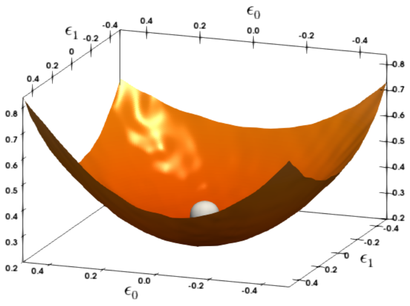
Go to page 8

[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV'19

Hessian-Aware Quantization for BERT-Base on MNLI

Method	w-bits	e-bits	Acc m	Acc mm	Size	Size w/o-e
Baseline	32	32	84.00	84.40	415.4	324.5
Q-BERT	8	8	83.91	83.83	103.9	81.2
DirectQ	4	8	76.69	77.00	63.4	40.6
Q-BERT	4	8	**83.89**	**84.17**	63.4	40.6
DirectQ	3	8	70.27	70.89	53.2	30.5
Q-BERT	3	8	**83.41**	**83.83**	53.2	30.5
Q-BERT$_{MP}$	2/4 $_{MP}$	8	**83.51**	**83.55**	53.2	30.5
DirectQ	2	8	53.29	53.32	43.1	20.4
Q-BERT	2	8	**76.56**	**77.02**	43.1	20.4
Q-BERT$_{MP}$	2/3 $_{MP}$	8	**81.75**	**82.29**	**46.1**	**23.4**





Automatic Mixed-Precision Quantization

We prove Hessian Trace is a better sensitivity metric than the Top-1 Eigenvalue.

Hessian Trace can be used to quantify second-order perturbation Ω.

Mixed-precision quantization becomes an Integer Linear Programming (ILP) problem:

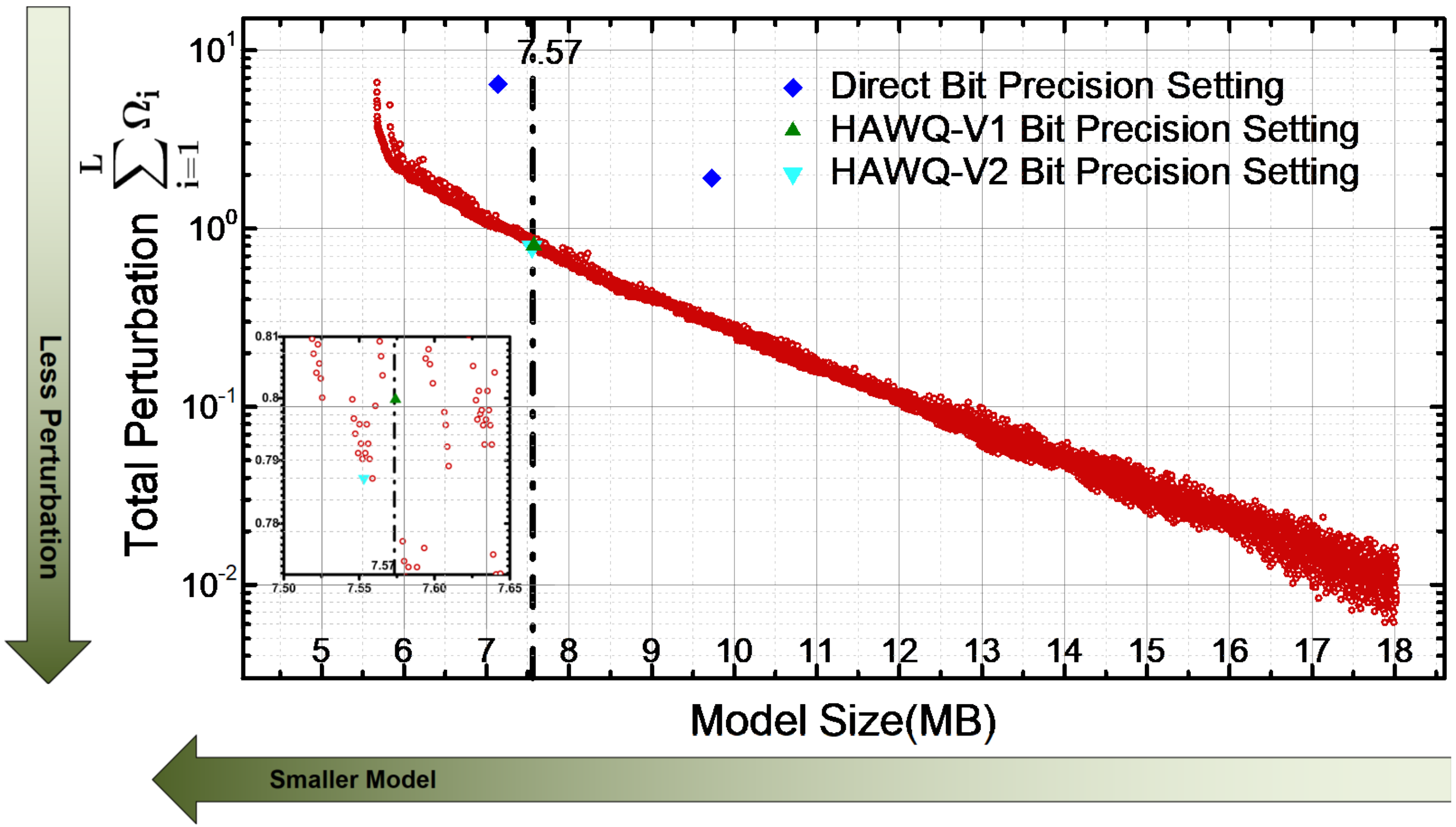
$$\Omega = \sum_{i=1}^{L} \Omega_i = \sum_{i=1}^{L} \overline{Tr}(H_i) \cdot \|Q(W_i) - W_i\|_2^2,$$

$$\textbf{Objective:} \quad \min_{\{b_i\}_{i=1}^{L}} \sum_{i=1}^{L} \Omega_i^{(b_i)},$$

$$\textbf{Subject to:} \quad \sum_{i=1}^{L} M_i^{(b_i)} \leq \textbf{Model Size Limit},$$

[1] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

Automatic Mixed-Precision Quantization



[1] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

HAWQ-V2: ResNet50 on ImageNet

Precisions for all layers are 100% automatically selected.

Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	77.39	1.00×	97.8
Dorefa [28]	2	2	67.10	16.00×	6.11
Dorefa [28]	3	3	69.90	10.67×	9.17
PACT [6]	2	2	72.20	16.00×	6.11
PACT [6]	3	3	75.30	10.67×	9.17
LQ-Nets [26]	3	3	74.20	10.67×	9.17
Deep Comp. [10]	3	MP	75.10	10.41×	9.36
HAQ [23]	MP	MP	75.30	10.57×	9.22
HAWQ [7]	2 MP	4 MP	75.48	12.28×	7.96
HAWQ-V2	2 MP	4 MP	**75.92**	12.24×	7.99

[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV 2019.
[2] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

HAWQ-V2: SqueezeNext on ImageNet

Precisions for all layers are 100% automatically selected.

Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	69.38	1.00×	10.1
Direct [7]	3 MP	8	65.39	9.04×	1.12
HAWQ [7]	3 MP	8	68.02	9.26×	1.09
HAWQ-V2	3 MP	8	**68.68**	9.40×	1.07

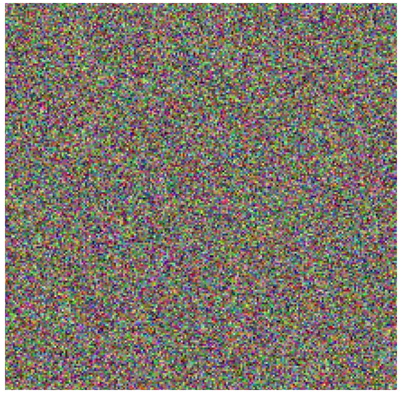
[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV 2019.
[2] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

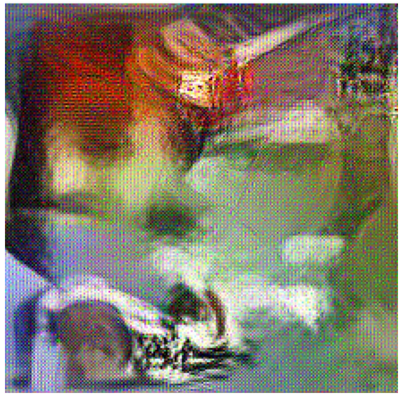
HAWQ-V2: RetinaNet on MS COCO

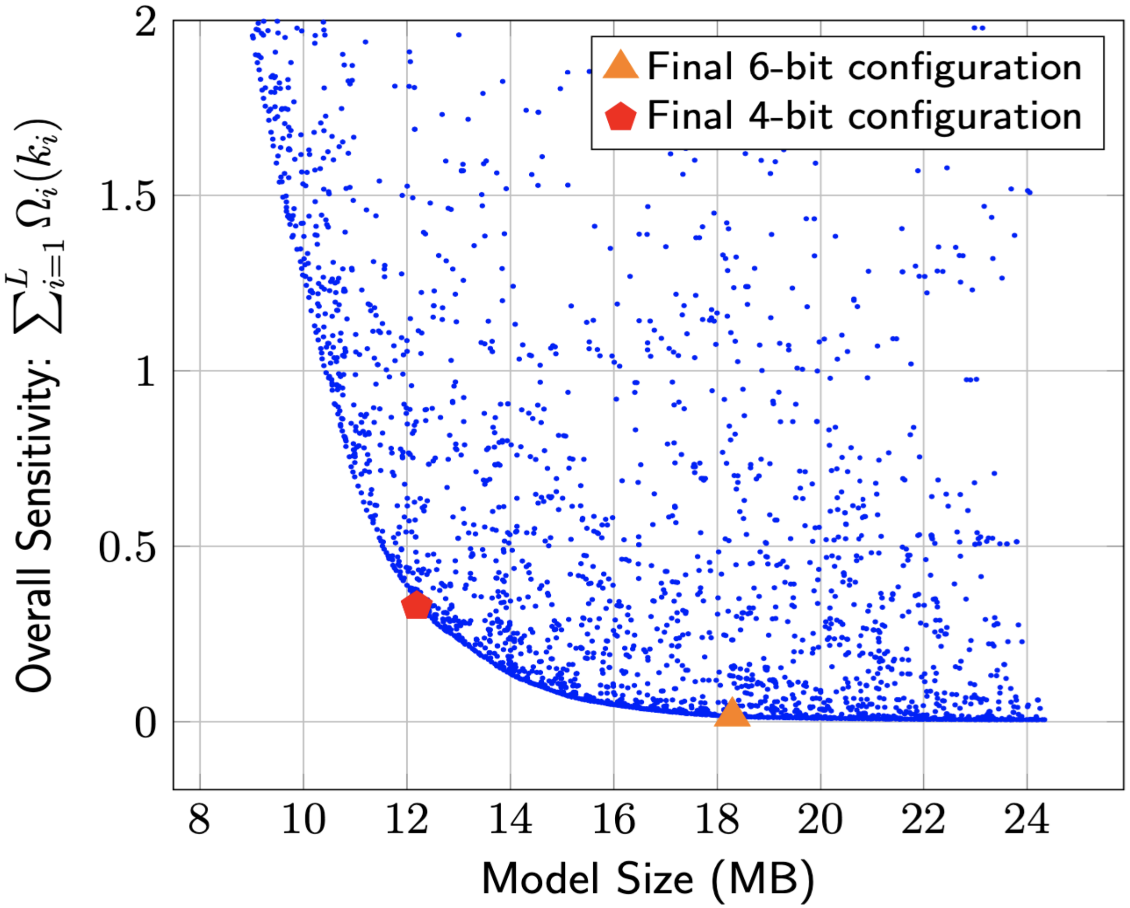
Method	w-bits	a-bits	mAP	W-Comp	A-Comp	Size(MB)
Baseline	32	32	35.6	1.00×	1.00×	145
Direct	4	4	31.5	8.00×	8.00×	18.13
FQN [15]	4	4	32.5	8.00×	8.00×	18.13
HAWQ-V2	3 MP	4	**34.1**	8.10×	8.00×	17.90
HAWQ-V2	3 MP	4 MP	**34.4**	8.10×	7.62×	17.90
HAWQ-V2	3 MP	6	**34.8**	8.10×	5.33×	17.90

[1] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

ZeroQ: Zero Shot Quantization







- Zero Shot quantization without data and fine-tuning.
- Quantize ResNet-50 end-to-end within 30s.
- Generalize well to various models on ImageNet and Microsoft COCO.

[1] Yaohui Cai*, Zhewei Yao*, Zhen Dong*, Amir Gholami, Michael W. Mahoney and Kurt Keutzer. ZeroQ: A Novel Zero Shot Quantization Framework, CVPR 2020.

ZeroQ: Zero Shot Quantization

ResNet50 on ImageNet

Method	No D	No FT	W-bit	A-bit	Size (MB)	Top-1
Baseline	–	–	32	32	97.49	77.72
OMSE [18]	✓	✓	4	32	12.28	70.06
OMSE [18]	✗	✓	4	32	12.28	74.98
PACT [4]	✗	✗	4	4	12.19	76.50
ZeroQ	✓	✓	MP	8	**12.17**	**75.80**
OCS [43]	✗	✓	6	6	18.46	74.80
ZeroQ	✓	✓	MP	6	**18.27**	**77.43**
ZeroQ	✓	✓	8	8	24.37	**77.67**

RetinaNet-ResNet50 on MS COCO

Method	W-bit	A-bit	Size (MB)	mAP
Baseline	32	32	145.10	36.4
ZeroQ	8	8	36.25	**36.4**
FQN	4	4	18.13	32.5
ZeroQ	MP	8	18.13	**33.7**

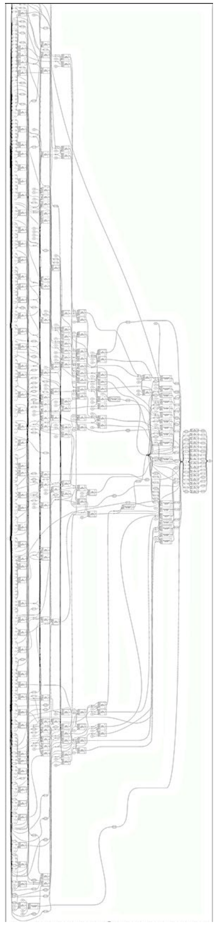
[1] Yaohui Cai*, Zhewei Yao*, Zhen Dong*, Amir Gholami, Michael W. Mahoney and Kurt Keutzer. ZeroQ: A Novel Zero Shot Quantization Framework, CVPR 2020.

Outline

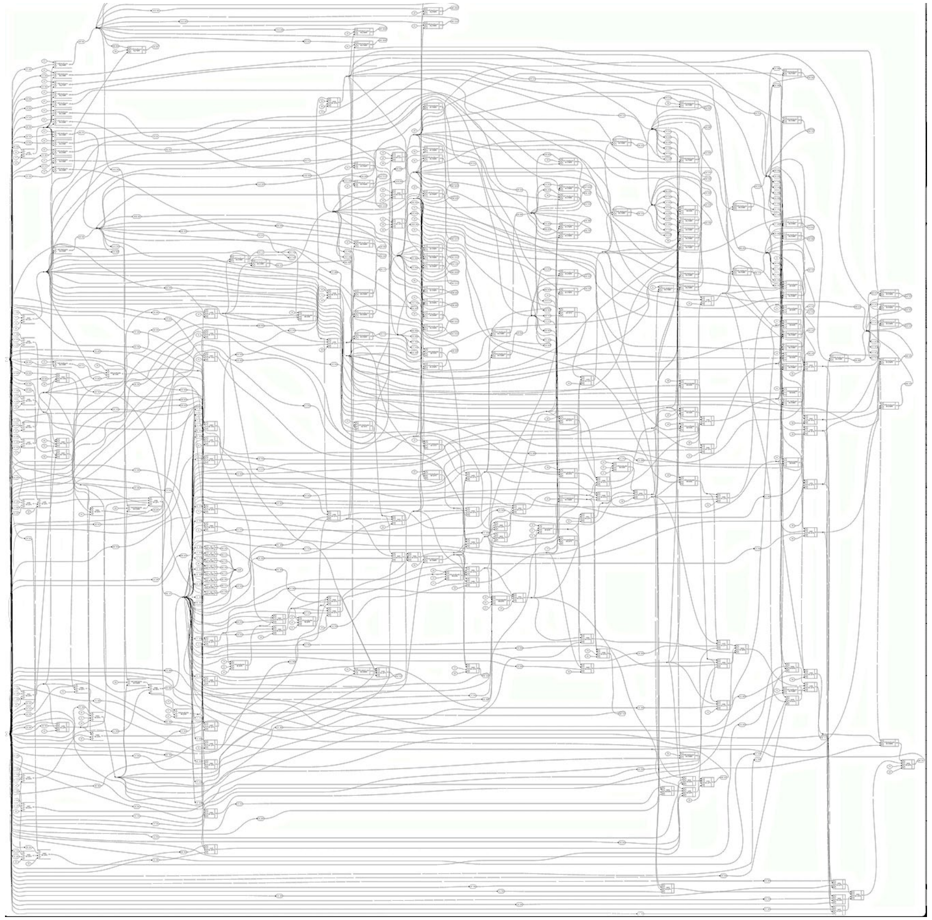
- Introduction
- Hessian-AWare Quantization (HAWQ)
- Hardware-aware Deployment
- Hardware-software Co-design
- Conclusion

Division Cost

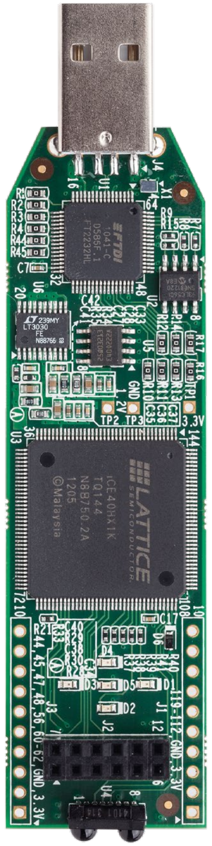
Multiplication vs division on an ICE40 FPGA



8x8 INT MUL



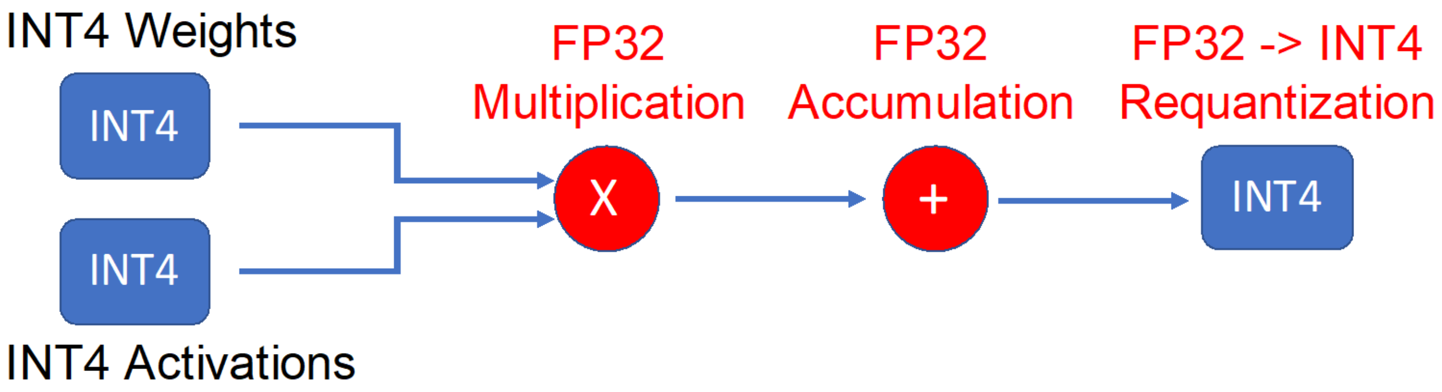
8x8 INT DIV

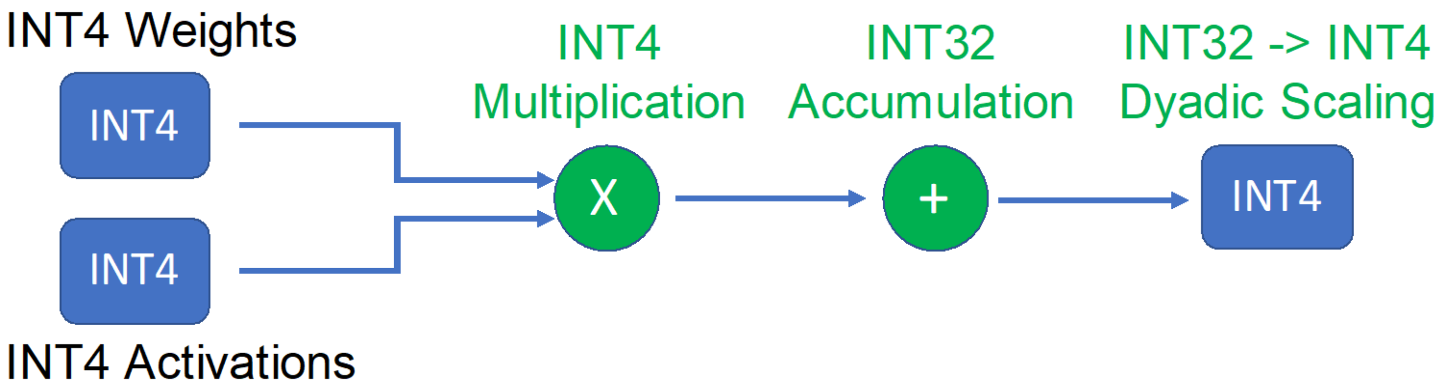


ICE40
Ultra low power FPGA

Reference: Marcus Muller

Dyadic Quantization with Integer Arithmetic





TVM

- A compiler stack for CPU, GPU and accelerators

- Autotuning framework

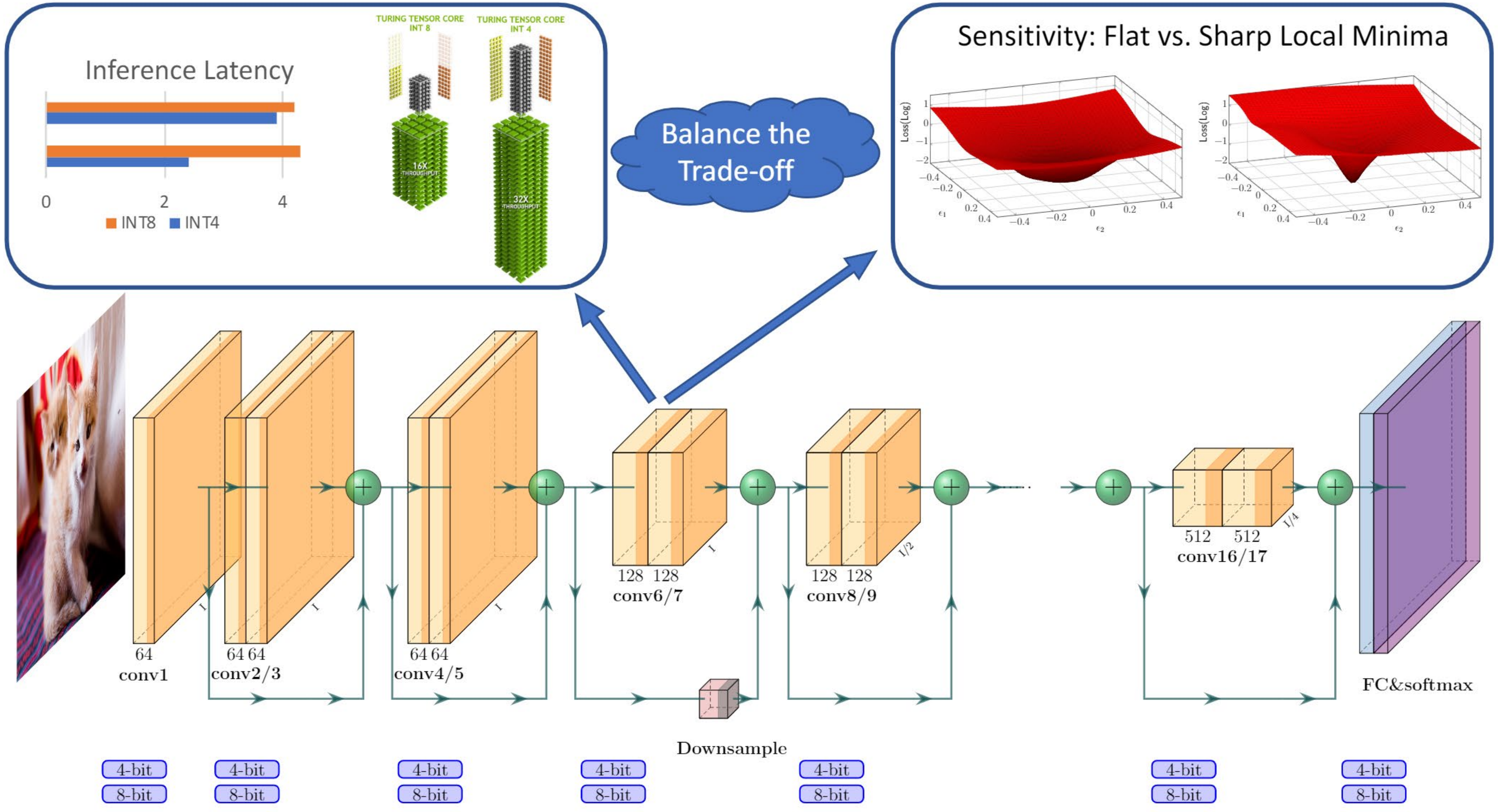
Need to add:

1. Mixed-precision support

2. Low-bit operations support

[1] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. TVM: An Automated End-to-End Optimizing Compiler for Deep Learning. In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), pages 578–594, 2018.

HAWQ-V3: Hardware-Aware Deployment



[1] Z. Yao*, Z. Dong*, Z. Zheng*, A. Gholami*, J. Yu, E. Tan, L. Wang, Q. Huang, Y. Wang, M. Mahoney, K. Keutzer, HAWQ-V3: Dyadic Neural Network Quantization, ICML 2021.

HAWQ-V3: Hardware-Aware Deployment

We find the best bit precision configuration that:

- Minimally perturbs the model
- Meets application specific requirements:
  - Model size constraint
  - Total bit operations for inference
  - Inference Latency

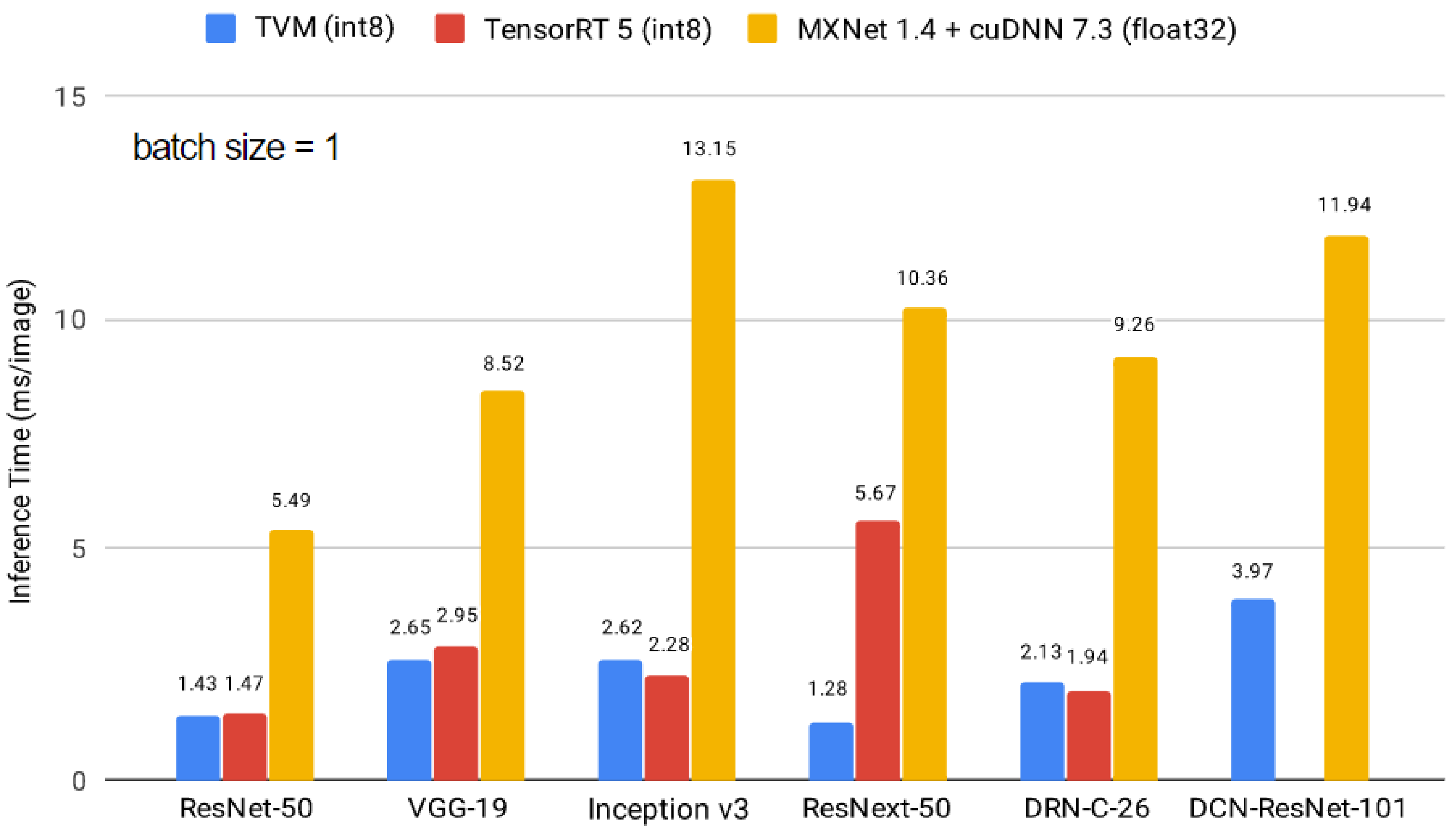
$$\text{Objective:} \quad \min_{\{b_i\}_{i=1}^{L}} \sum_{i=1}^{L} \Omega_i^{(b_i)},$$

$$\text{Subject to:} \quad \sum_{i=1}^{L} M_i^{(b_i)} \leq \text{ Model Size Limit},$$

$$\sum_{i=1}^{L} G_i^{(b_i)} \leq \text{ Bops Limit},$$

$$\sum_{i=1}^{L} Q_i^{(b_i)} \leq \text{ Latency Limit}.$$

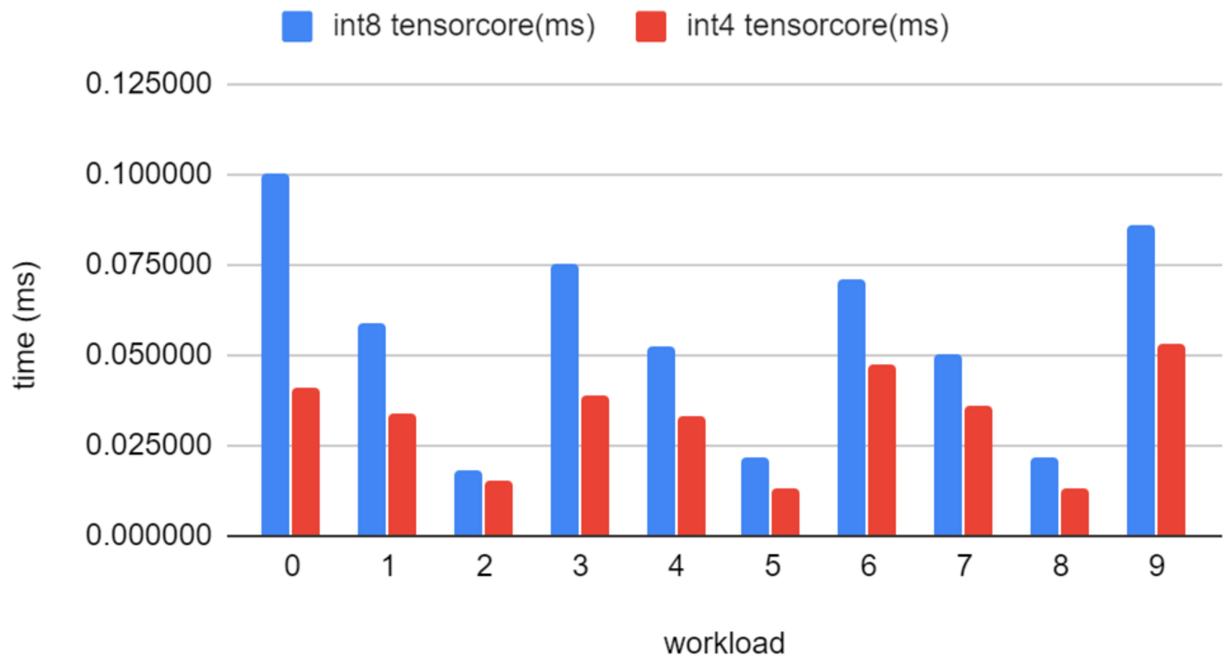
Uniform 8-bit Performance



[1] Image from https://tvm.apache.org/2019/04/29/opt-cuda-quantized

Int4 Performance of ResNet18 on ImageNet

Convolution Benchmark for ResNet18 Workloads



**A workload is a convolutional function with certain shape*

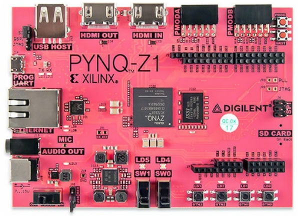
ResNet18 End-to-end Speedup

Resnet 18	Int8 (ms)	Int4 (ms)	Speed-up
Batch=1	0.85	0.62	1.37x
Batch=8	4.55	3.02	1.51x
Batch=16	8.84	5.91	1.50x

Outline

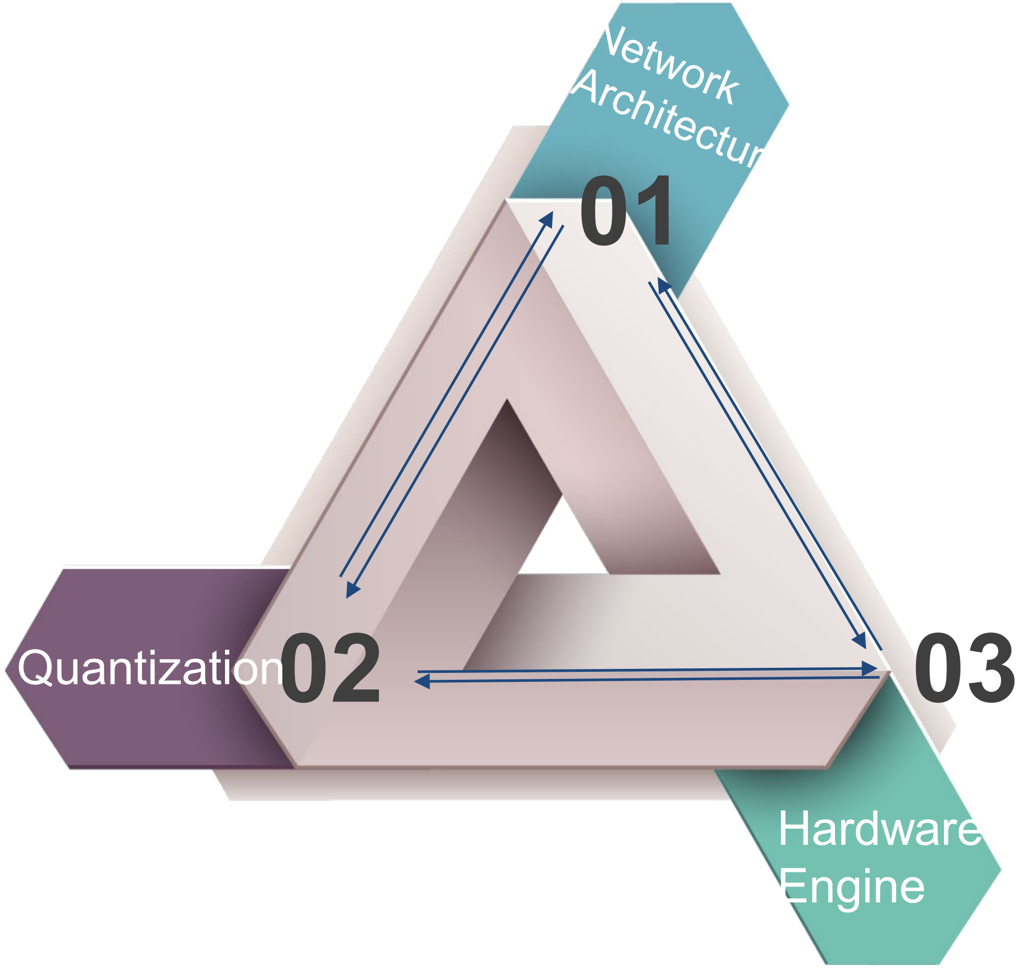
- Introduction
- Hessian-AWare Quantization (HAWQ)
- Hardware-aware Deployment
- Hardware-software Co-design
- Conclusion

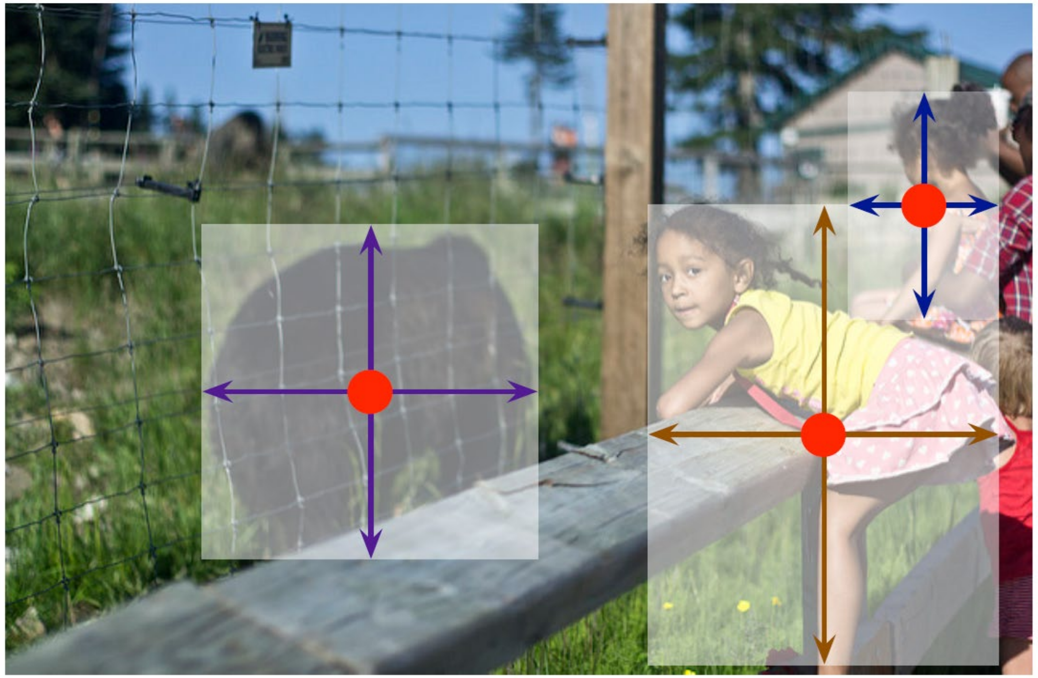
Hardware-Software Co-Design

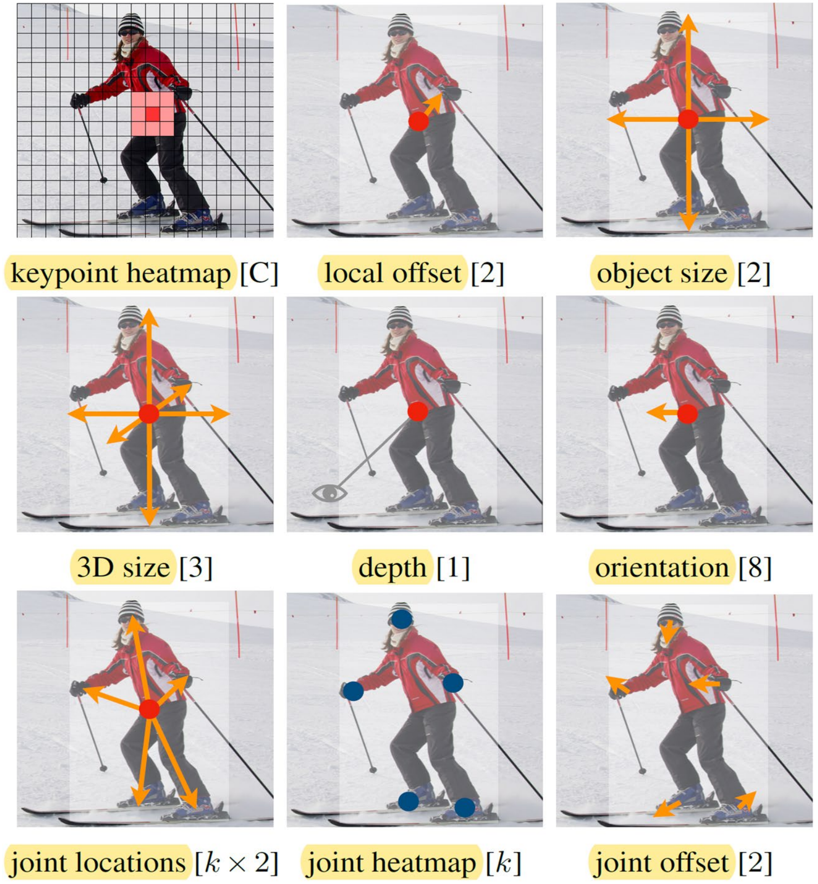


Board Specifications	
Linux Image	PYNQ v2.4 (Ubuntu v18.04)
SoC Chip	Xilinx ZYNQ XC7Z020-1CLG400C CPU: ARM Cortex-A9 650MHz FPGA: Artix-7
DRAM	DDR3 512MB
FPGA Specifications	
BRAM	280 blocks
DSP	220 slices
FF	106,400 instances
LUT	53,200 instances

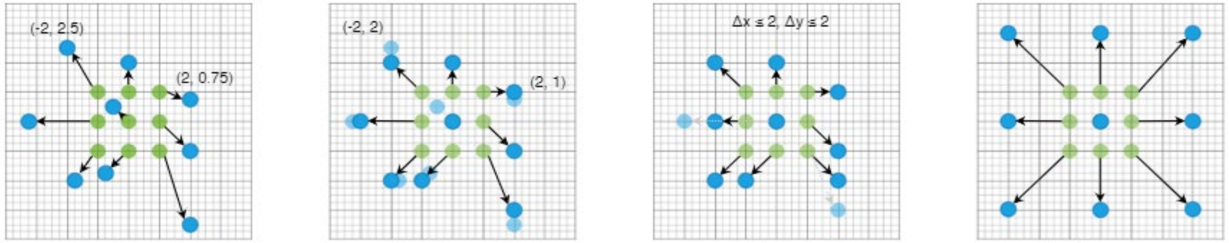
	Criteria	Specification	Percentage usage of FPGA
FPGA resource	Number of slides	46076	67
	Number of RAM Blocks	31	10
	Number of DSPs	64	66
Speed	Max. frequency	50 MHz	Maximum
Power	Dynamic power	686 mW	
	Leakage power	1128 mW	



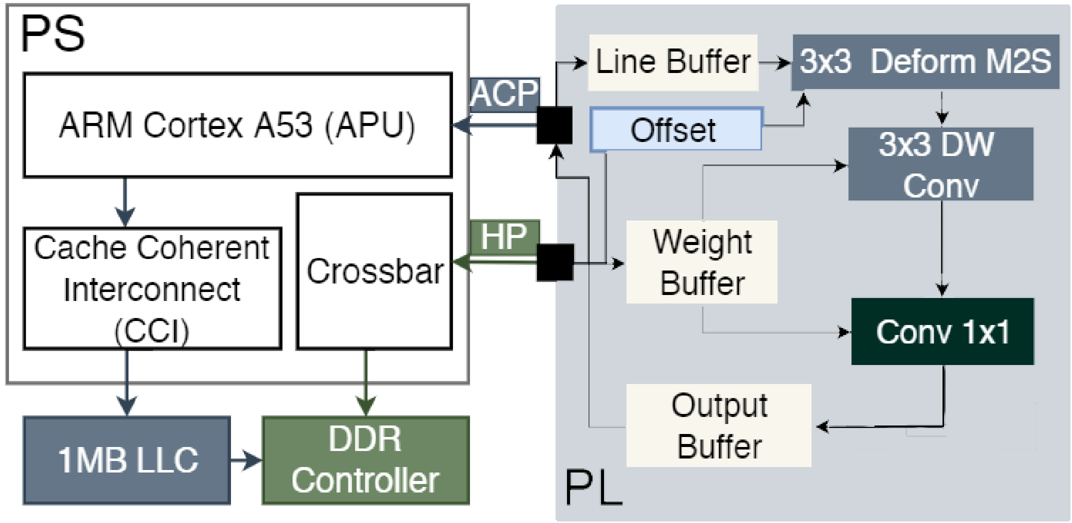



keypoint heatmap [C]
local offset [2]
object size [2]
3D size [3]
depth [1]
orientation [8]
joint locations [$k \times 2$]
joint heatmap [k]
joint offset [2]

CoDeNet: Algorithm-Hardware Co-design

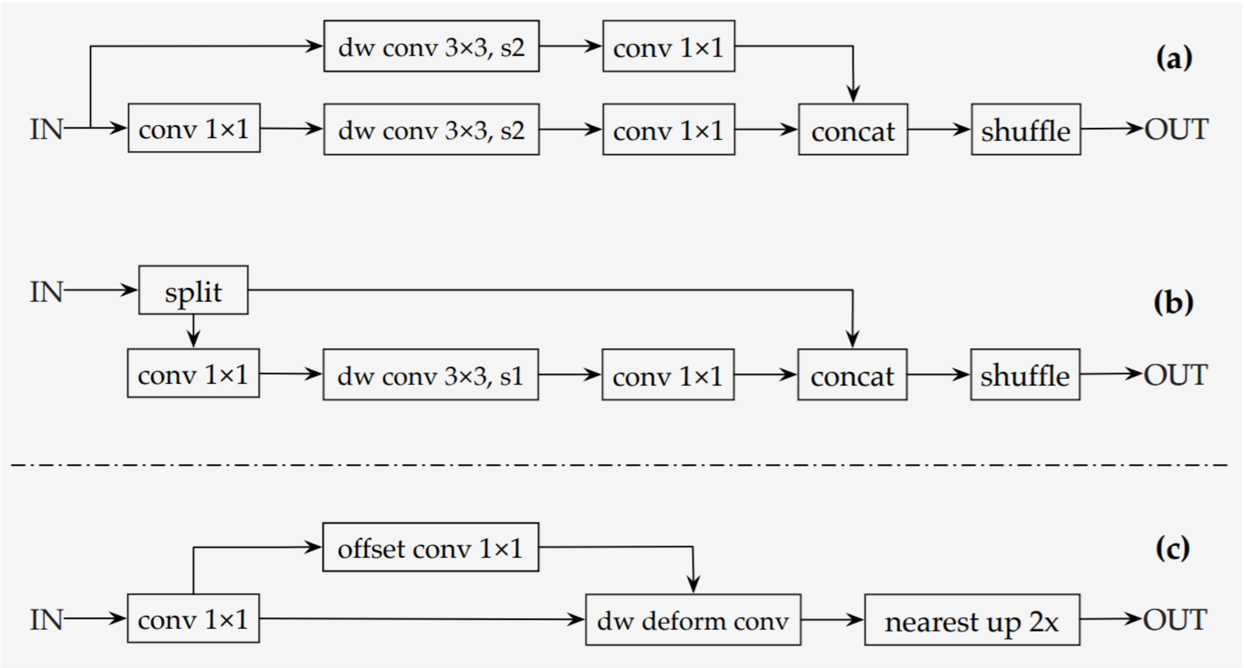


a. Deformable b. Round c. Bound d. Square

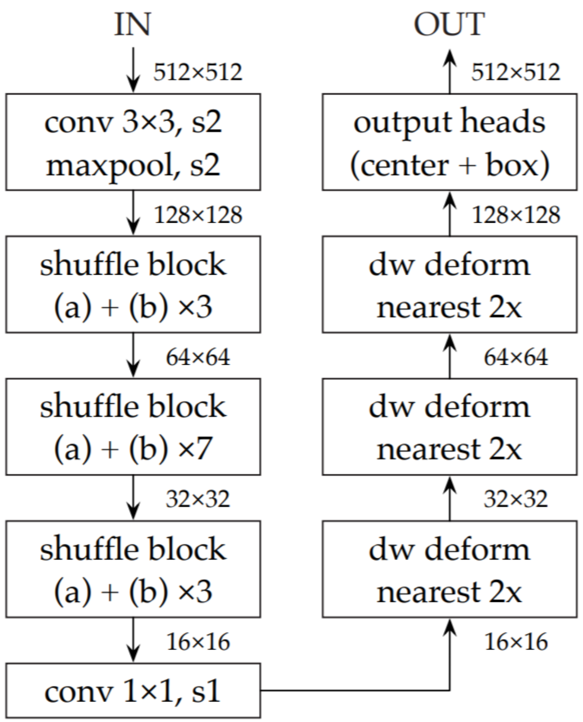


[1] Z. Dong, D. Wang, Q. Huang, Y. Gao, Y. Cai, B. Wu, K. Keutzer, J. Wawrzynek, CoDeNet: Algorithm-hardware Co-design for Deformable Convolution, FPGA 2021.

CoDeNet: Algorithm-Hardware Co-design



(i) building blocks



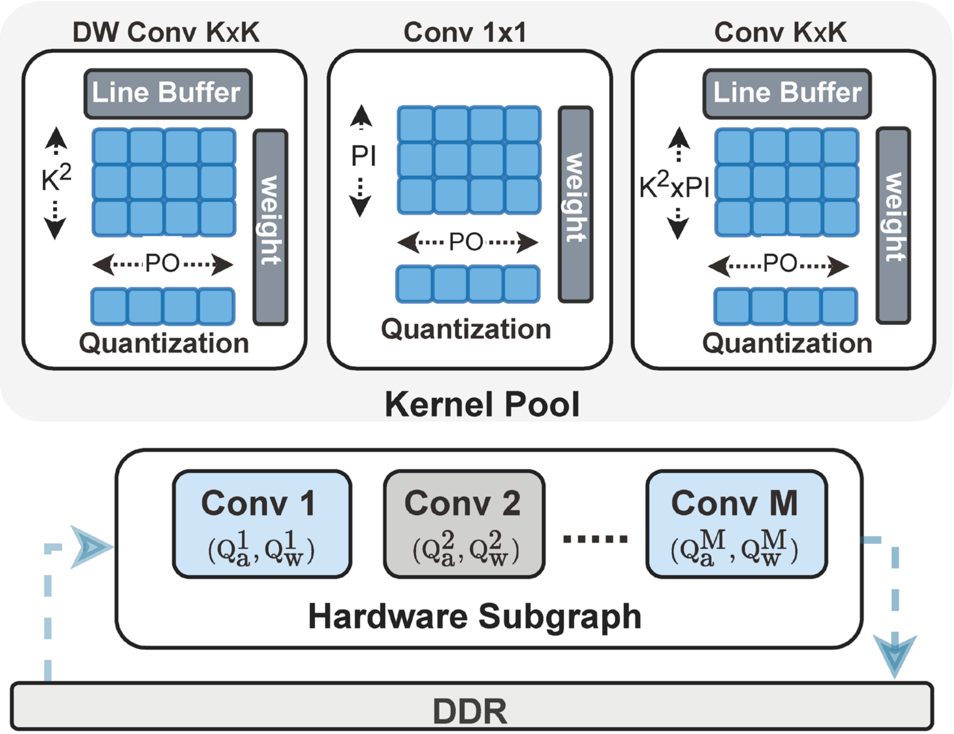
(ii) model architecture

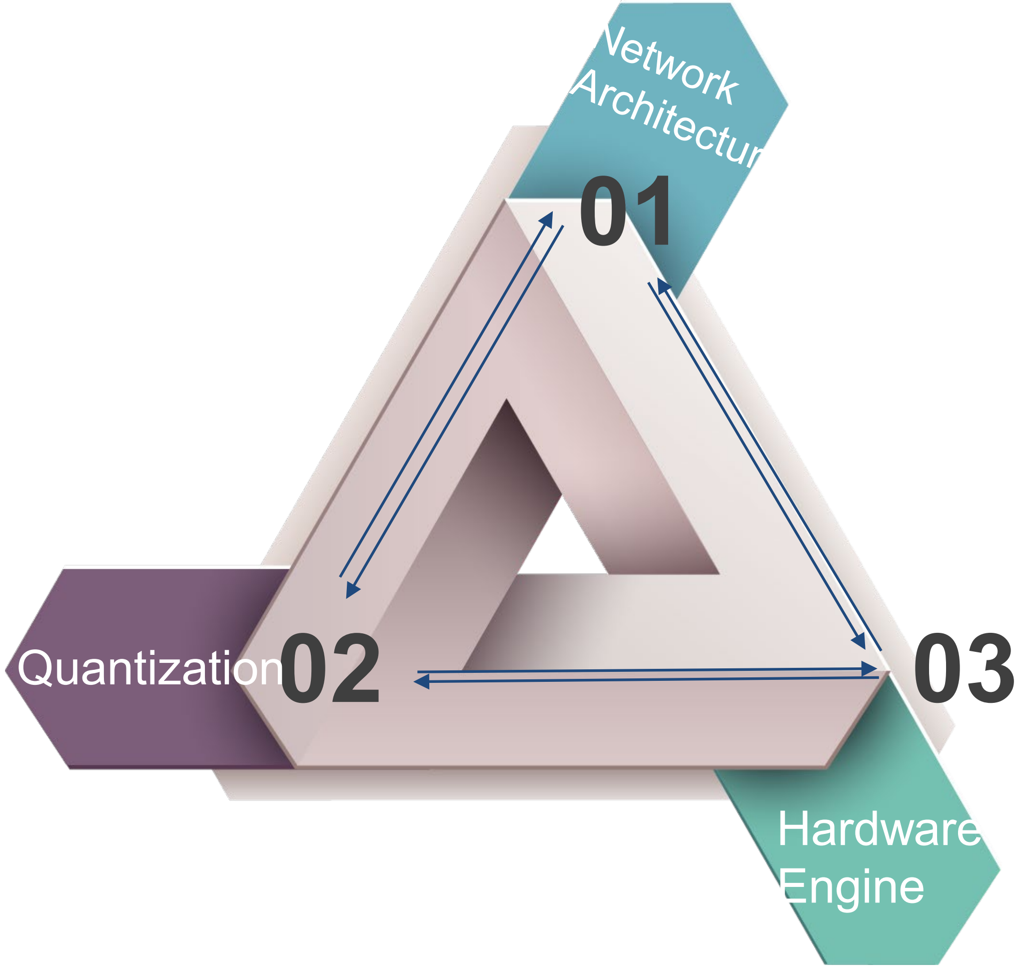
CoDeNet: Algorithm-Hardware Co-design

	Platform	Input Resolution	Framerate (fps)	Test Dataset	Precision	Accuracy
DNN1 [12]	Pynq-Z1	-	17.4	DJI-UAV	a8	IoU(68.8)
DNN3 [12]	Pynq-Z1	-	29.7		a16	IoU(59.3)
Skynet [41]	Ultra96	160 × 360	25.5		w11a9	IoU(71.6)
AP2D [20]	Ultra96	224 × 224	30.5	AD2P	w(1-24)a3	IoU(55)
Finn-R [2] [28]	Ultra96	-	16	VOC07	w1a3	AP50(50.1)
Tiny-Yolo-v2 [11]	Zynq-706 XC7Z045	224 × 224	43.1		w16a16	AP50(48.5)
Ours (config a)	Ultra96	256 × 256	32.2	VOC07	w4a8	AP50(51.1)
Ours (config b)		256 × 256	26.9			AP50(55.1)
Ours (config c)		512 × 512	9.3			AP50(61.7)
Ours (config d)		512 × 512	5.2			AP50(67.1)
Ours (config e)		512 × 512	4.6			AP50(69.7)

[1] Z. Dong, D. Wang, Q. Huang, Y. Gao, Y. Cai, B. Wu, K. Keutzer, J. Wawrzynek, CoDeNet: Algorithm-hardware Co-design for Deformable Convolution, FPGA 2021.

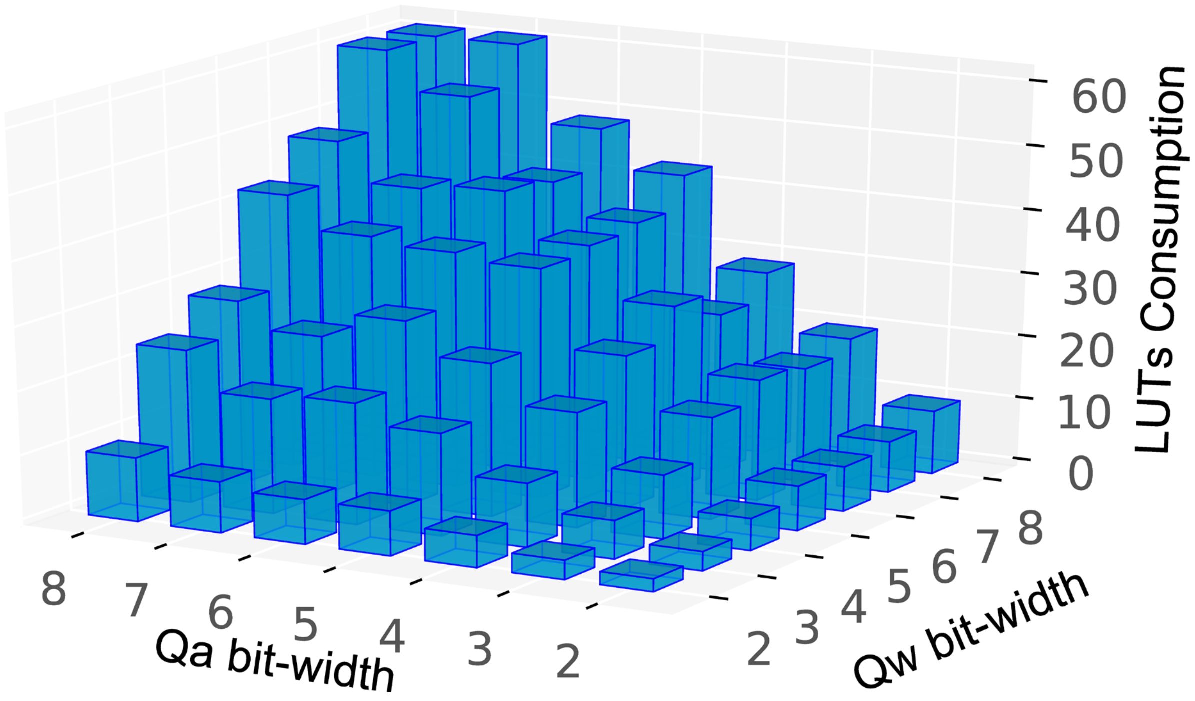
HAO: Hardware-Aware Architecture Optimization

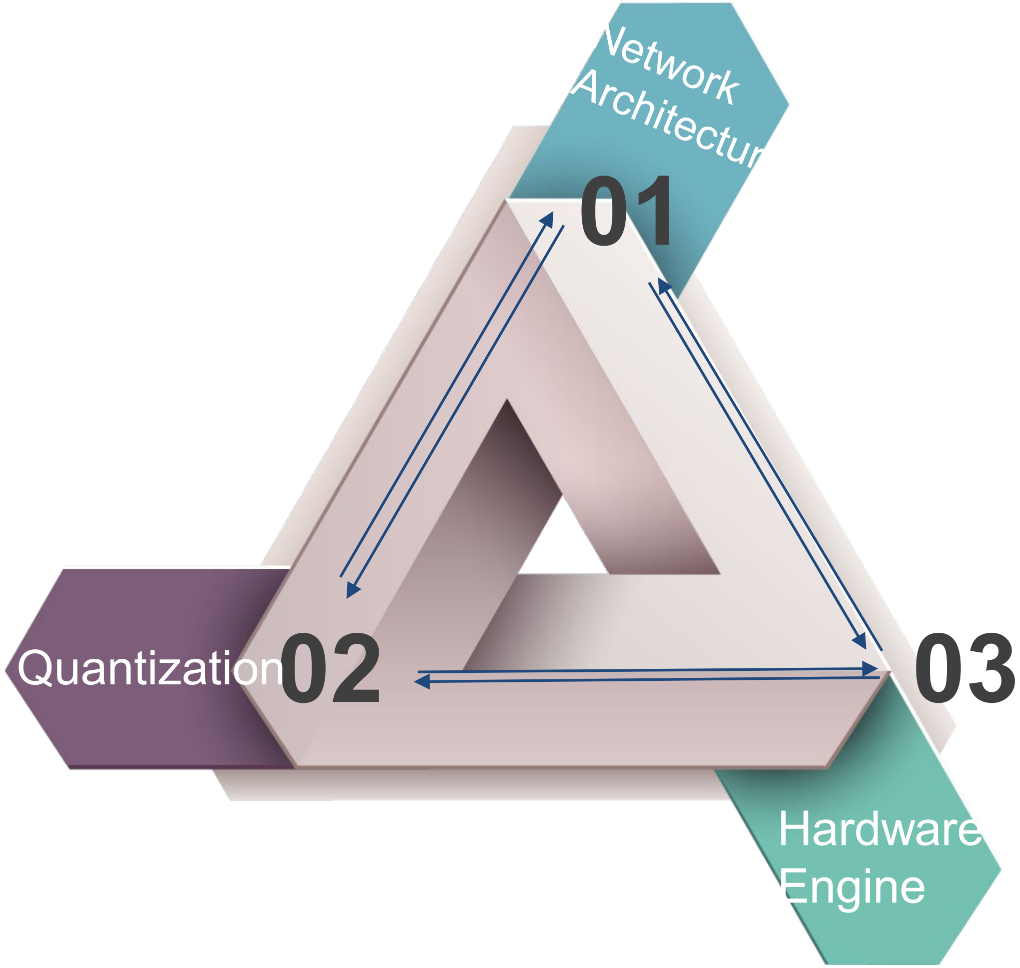




[1] Z. Dong, Y. Gao, Q. Huang, J. Wawrzynek, H. So, K. Keutzer, Hardware-Aware Neural Architecture Optimization for Efficient Inference, FCCM 2021.

HAO: Hardware-Aware Architecture Optimization





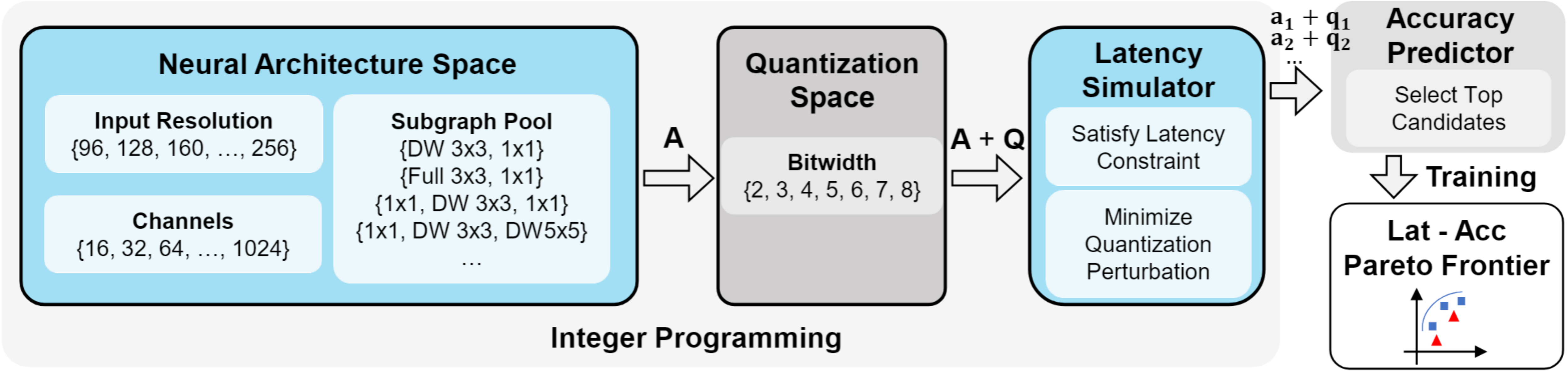
[1] Z. Dong, Y. Gao, Q. Huang, J. Wawrzynek, H. So, K. Keutzer, Hardware-Aware Neural Architecture Optimization for Efficient Inference, FCCM 2021.

Latency Simulator

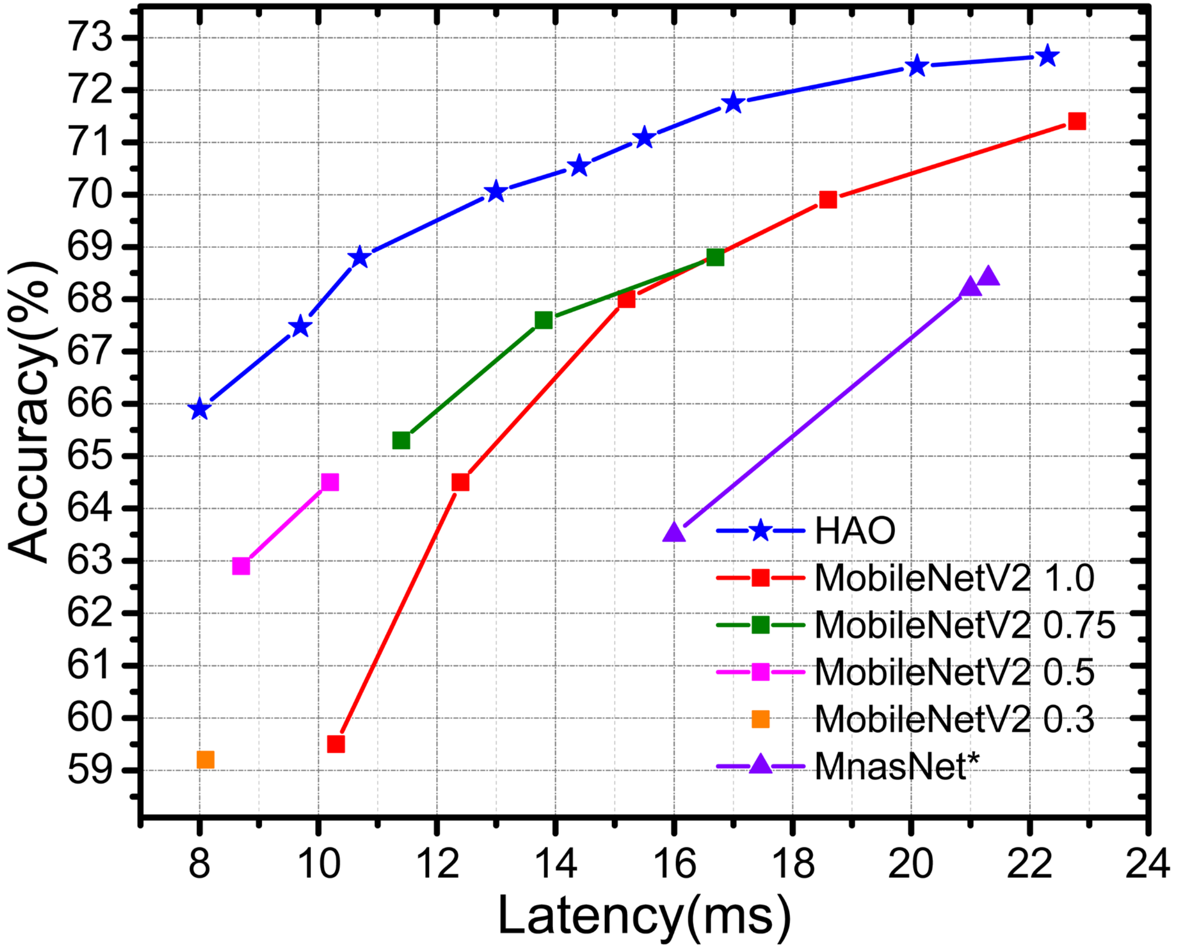
Given subgraph, quantization config, and network architecture, the latency simulator finds the lowest latency by selecting optimal hardware design parameters that minimize latency while satisfying resource constraints.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{L} Lat(g_i) \\
\text{s.t.} \quad & \sum_{k \in S} N_{\text{dsp}}^{k} \leq T_{\text{dsp}} \\
& \sum_{k \in S} N_{\text{luts}}^{k} \leq T_{\text{luts}} \times \beta \\
& \sum_{k \in S} N_{\text{bram}}^{k} \leq T_{\text{bram}}
\end{aligned}
$$

HAO Pipeline



HAO: Hardware-Aware Architecture Optimization



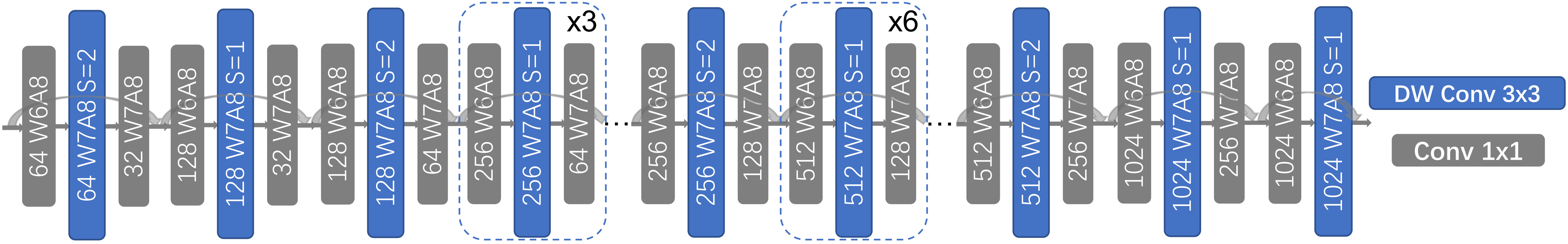
[1] Z. Dong, Y. Gao, Q. Huang, J. Wawrzynek, H. So, K. Keutzer, Hardware-Aware Neural Architecture Optimization for Efficient Inference, FCCM2021.

HAO Results

	Platform	Input Resolution	Framerate(fps)	Quantization Bitwidth	Top-1 Accuracy(%)
EDD-Net-2 [25]	Zynq ZU9EG	224×224	125.6	W16A16	74.6
HotNas-Mnasnet [20]	Zynq ZU9EG	224×224	200.4	NA	73.24
HotNas-ProxylessNAS [20]	Zynq ZU9EG	224×224	205.7	NA	73.39
EDD-Net-3 [25]	Zynq XC7Z045	224×224	40.2	W16A16	74.4
VGG16 [52]	Zynq XC7Z045	224×224	27.7	W16A16	69.3
VGG-SVD [31]	Zynq XC7Z045	224×224	4.5	W16A16	64.64
VGG16 [35]	Stratix-V	224×224	3.8	W8A16	66.58
VGG16 [13]	Zynq 7Z020	224×224	5.7	W8A8	67.72
Dorefa [23]	Zynq 7Z020	224×224	106.0	W2A2	46.10
Synetgy [47]	Zynq ZU3EG	224×224	66.3	W4A4	68.30
FINN-R [4]	Zynq ZU3EG	224×224	200.0	W1A2	50.30
MobileNetV2 [33]	Zynq ZU3EG	224×224	43.5	W8A8	71.40
MnasNet-A1 [37]	Zynq ZU3EG	224×224	22.3	W8A8	74.60
MnasNet-A1 [37]	Zynq ZU3EG	192×192	27.8	W8A8	73.33
MnasNet-A1-0.75 [37]	Zynq ZU3EG	224×224	31.0	W8A8	72.70
MnasNet-A1 [37]	Zynq ZU3EG	160×160	35.8	W8A8	71.35
FBNet-B [43]	Zynq ZU3EG	224×224	24.6	W8A8	73.20
FBNet-iPhoneX [43]	Zynq ZU3EG	224×224	21.3	W8A8	72.62
HAO	Zynq ZU3EG	256×256	44.9	W-mixed A8	72.68
HAO	Zynq ZU3EG	256×256	50.0	W-mixed A8	72.45
HAO	Zynq ZU3EG	224×224	58.9	W6A8	71.76
HAO	Zynq ZU3EG	224×224	77.0	W-mixed A8	70.06
HAO	Zynq ZU3EG	192×192	93.5	W-mixed A8	68.80

[1] Z. Dong, Y. Gao, Q. Huang, J. Wawrzynek, H. So, K. Keutzer, Hardware-Aware Neural Architecture Optimization for Efficient Inference, FCCM2021.

HAO Results



Some intuitive findings:

- Subgraph with only one type of Depthwise Conv (3x3) performs the best for our implementation on ZU3EG FPGA, HAO automatically finds that the support for 5x5 or 7x7 DW Conv here are inefficient.
- The proportion of channels between DW Conv and Conv are not necessarily a constant value through the network, though it is always kept constant in manual design and cell-based NAS.
- For our specific implementation, 6-7bit mixed-precision quantization can achieve the sweet point of trade-offs and therefore is much faster than 8-bit quantization of weights.

Outline

- Introduction
- Hessian-AWare Quantization (HAWQ)
- Hardware-aware Deployment
- Hardware-software Co-design
- Conclusion

Open-Source Projects

- HAWQ: https://github.com/Zhen-Dong/HAWQ
  Easy deployment and Fast inference,
  Support ResNets, Inceptions, MobileNets, EfficientNets, etc.
  High accuracy mixed-precision models (19MB ResNet50, 77% Acc on ImageNet).
- ZeroQ: https://github.com/amirgholami/ZeroQ
- CoDeNet: https://github.com/Zhen-Dong/CoDeNet
- BitPack: https://github.com/Zhen-Dong/BitPack
- HAP: https://github.com/yaozhewei/HAP
- Awesome Quantization: https://github.com/Zhen-Dong/Awesome-Quantization-Papers

Thank you for listening!