EQ-ViT: Algorithm-Hardware Co-Design for End-to-End Acceleration of Real-Time Vision Transformer Inference on Versal ACAP Architecture

International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) 2024

P. Dong*, J. Zhuang* , Z. Yang , S. Ji , Y. Li, D. Xu, H. Huang, J. Hu, A.K. Jones, Y. Shi,
Y. Wang, P. Zhou

*Co-first authors

Massachusetts Institute of Technology; Brown University; Northeastern University;
North Carolina State University; Syracuse University;
University of Maryland, College Park; University of Pittsburgh; University of Notre Dame

peggy281@mit.edu
jinming_zhuang@brown.edu
yanz.wang@northeastern.edu
peipei_zhou@brown.edu

EMBEDDED SYSTEMS WEEK

Latency critical applications

Latency critical applications

- Autonomous Driving

Latency critical applications

- Autonomous Driving

Latency critical applications

- Autonomous Driving

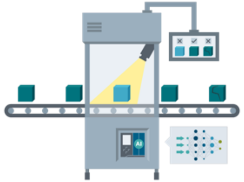
- Radio Access Network

Latency critical applications

- Autonomous Driving

- Radio Access Network

Online Defect Detection



AR/VR

Robot Systems Control

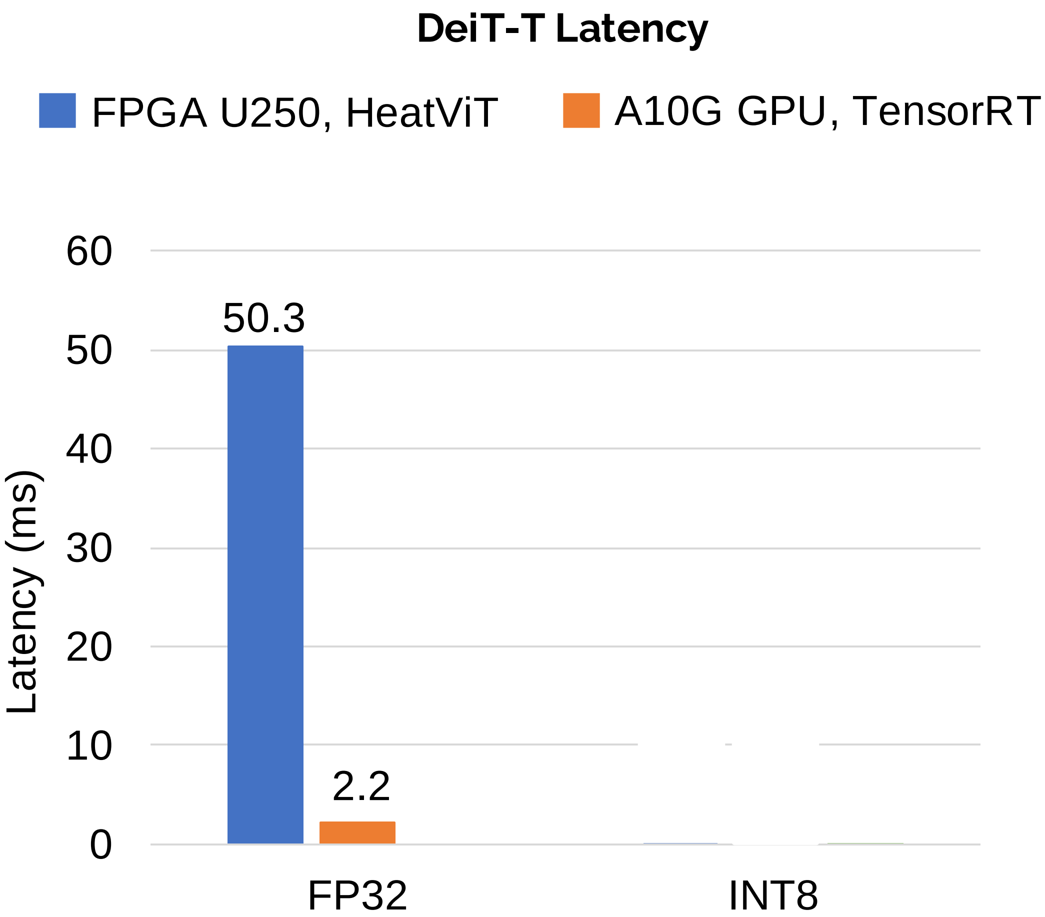
FPGA vs. GPU?

FPGA vs. GPU?

Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s

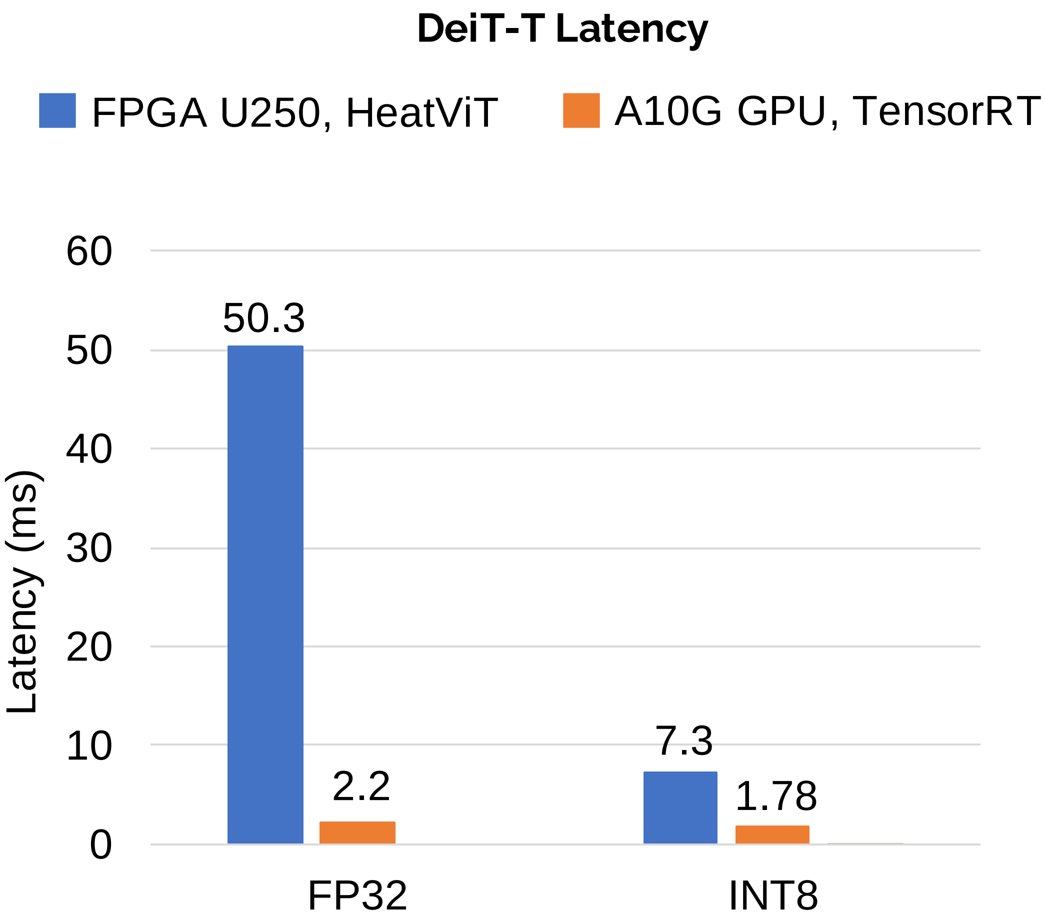
FPGA vs. GPU?



Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s

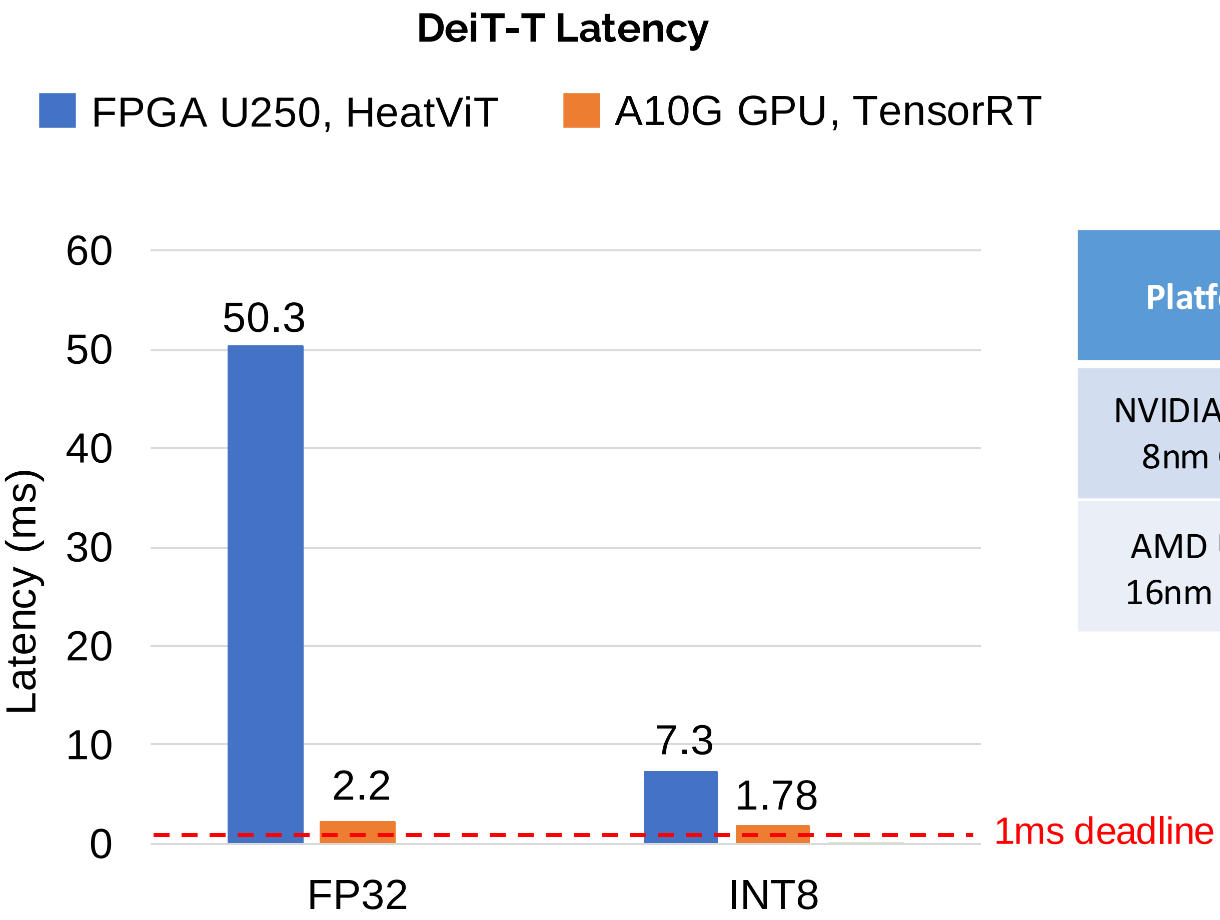
FPGA vs. GPU?



Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s

FPGA vs. GPU?

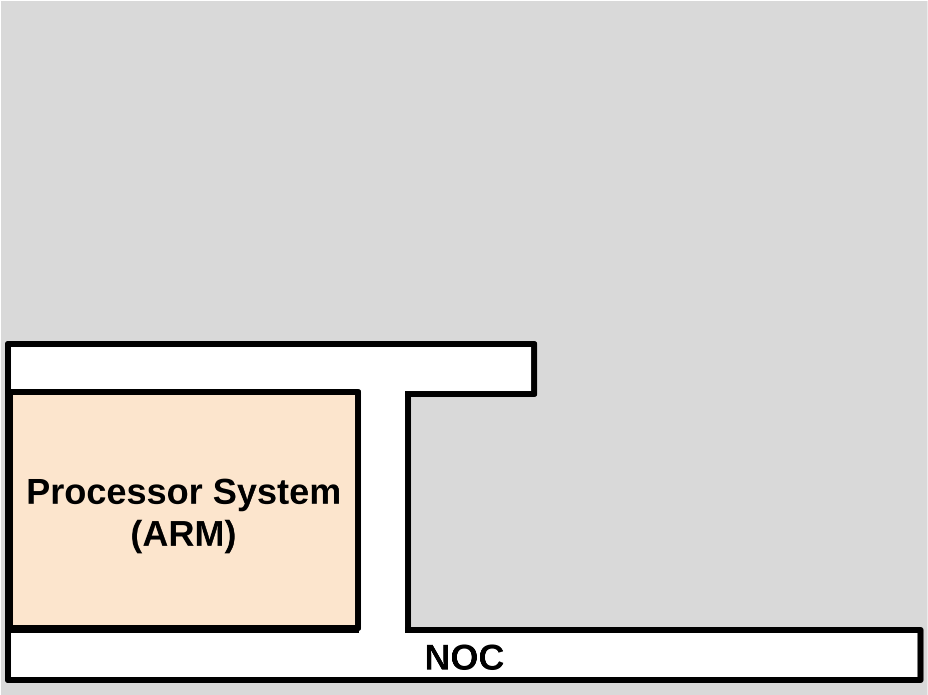


Hardware Specification

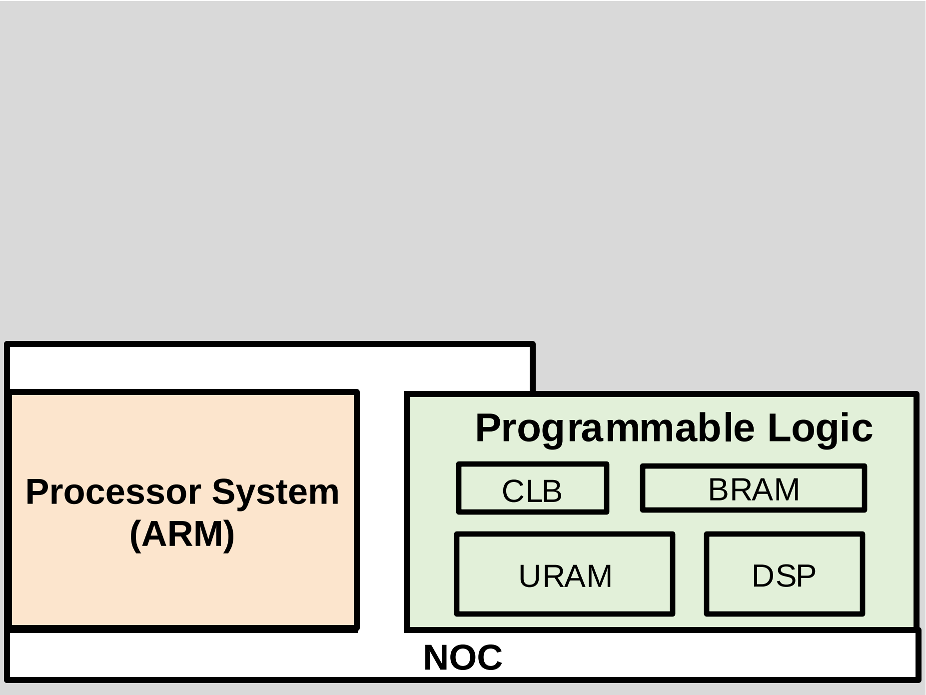
Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s

Versal ACAP Architecture

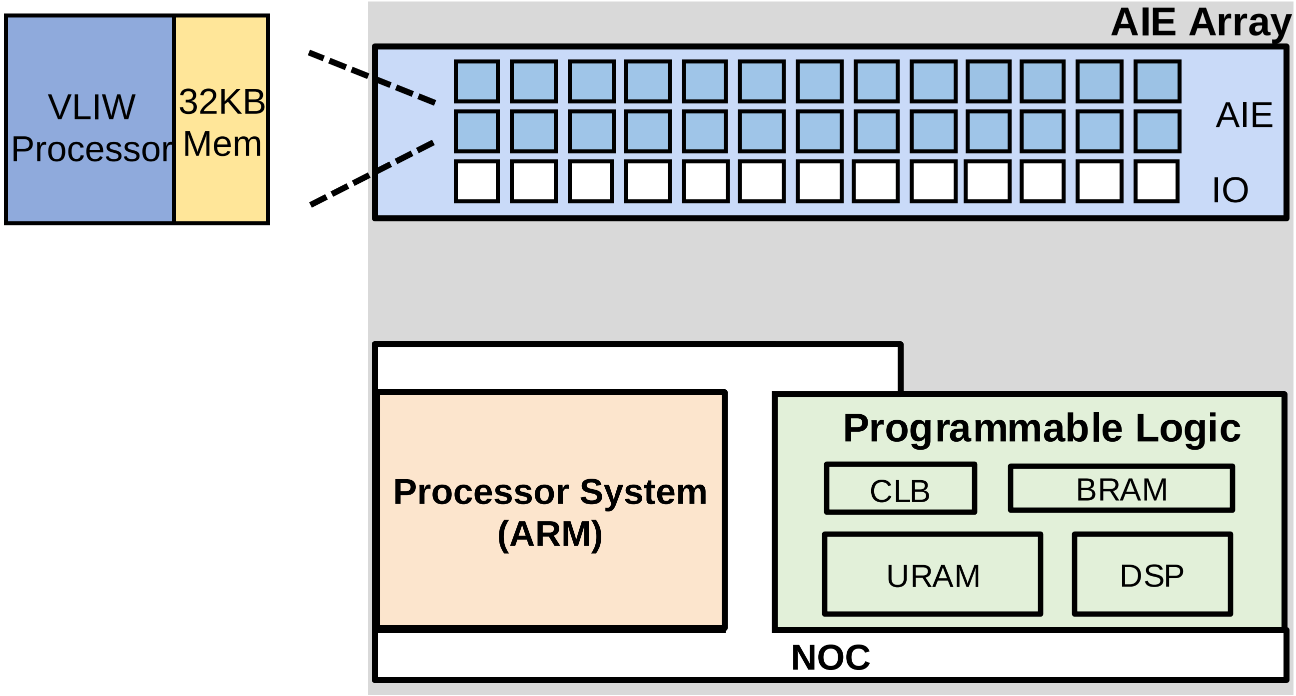
Versal ACAP Architecture



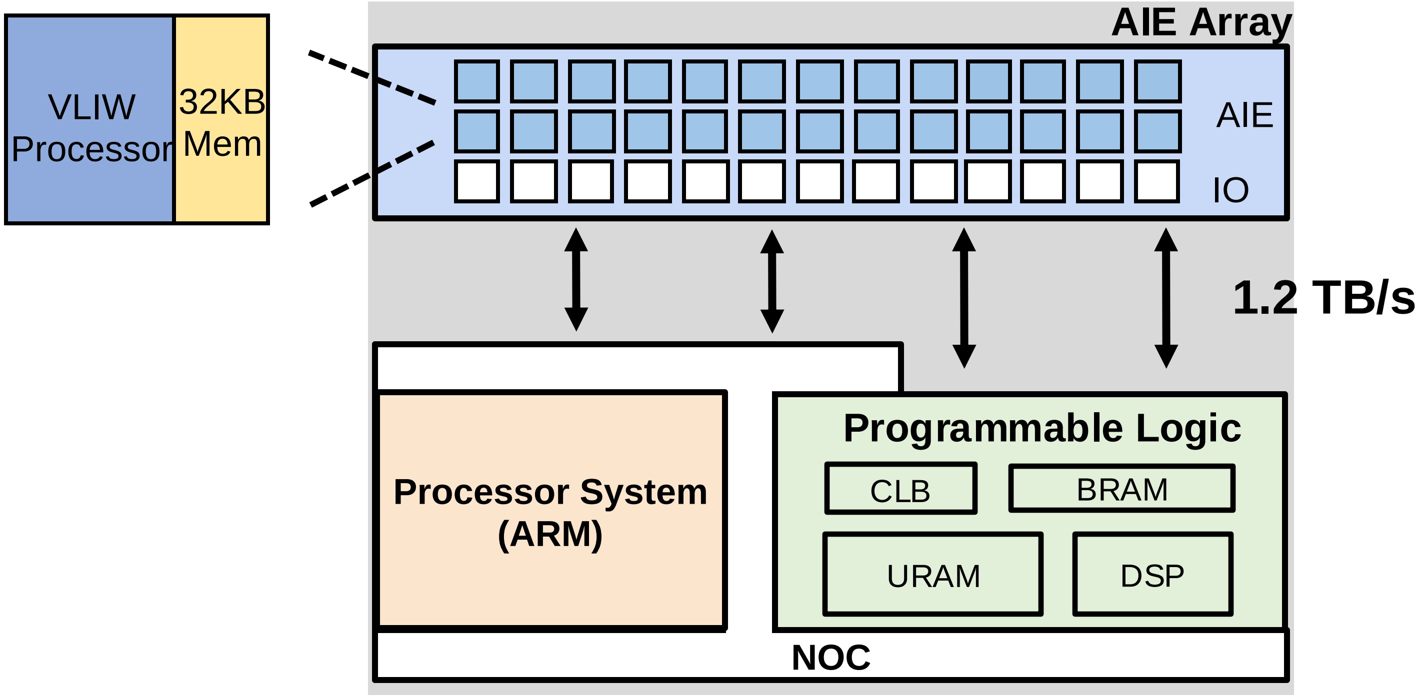
Versal ACAP Architecture



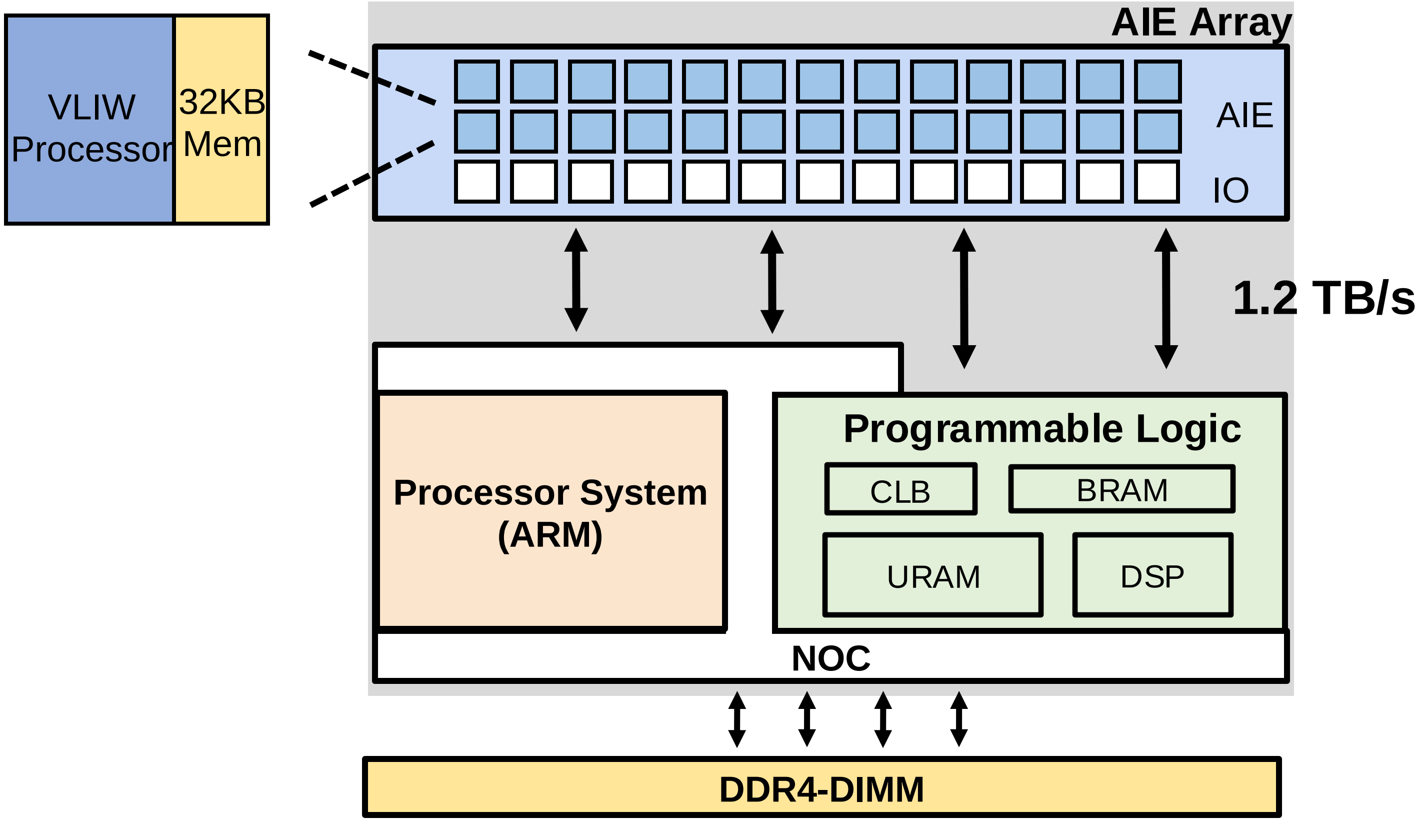
Versal ACAP Architecture



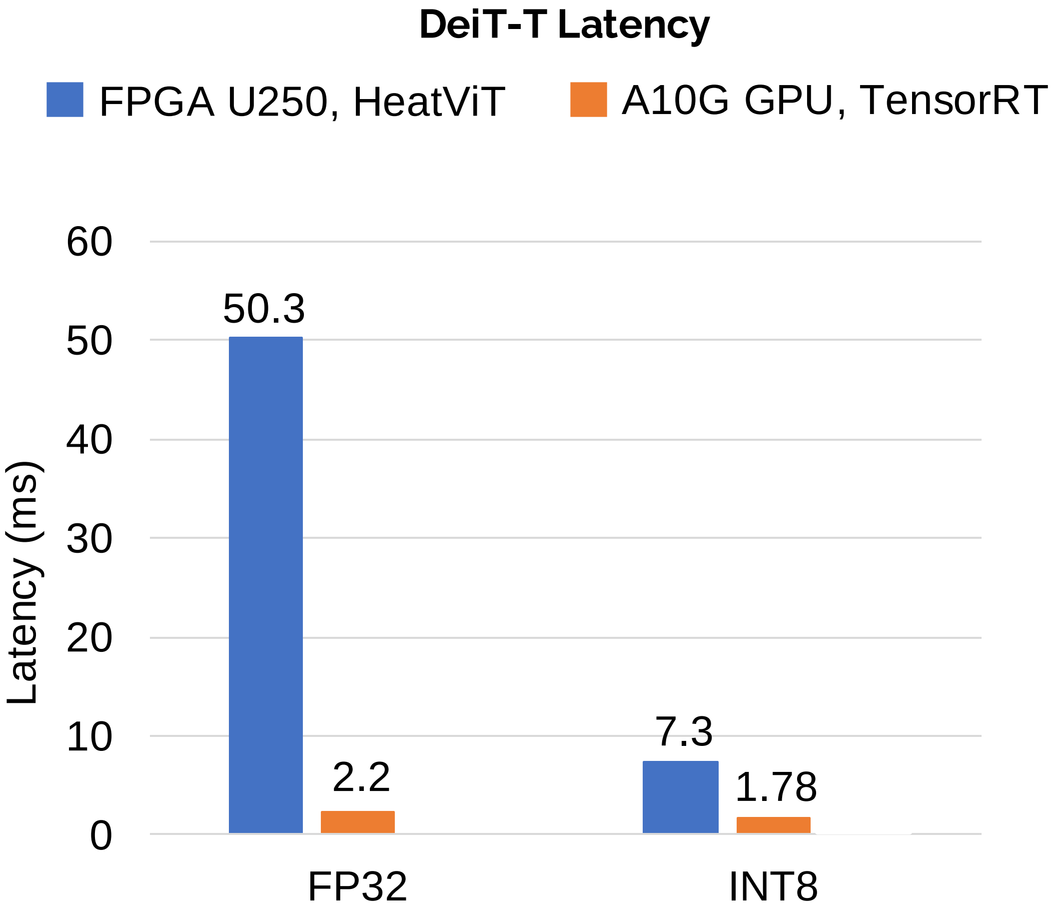
Versal ACAP Architecture



Versal ACAP Architecture



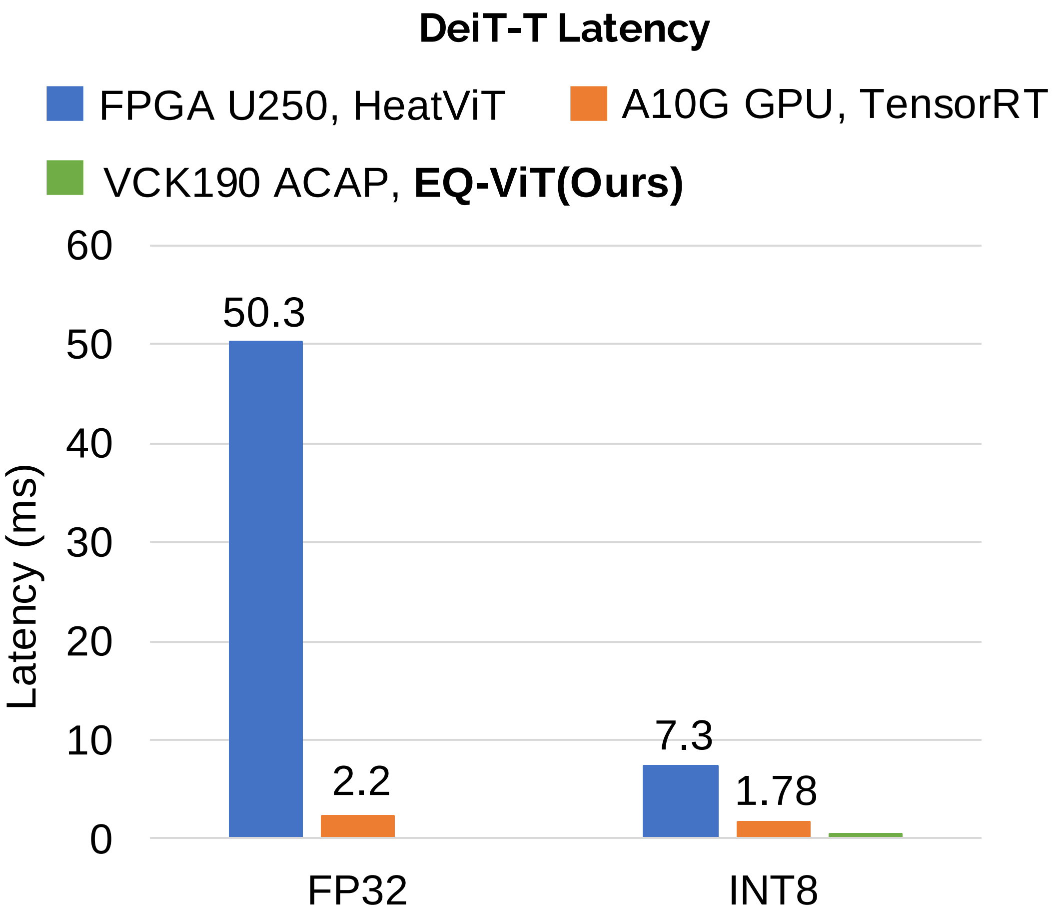
FPGA + Vector Processor?



Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s

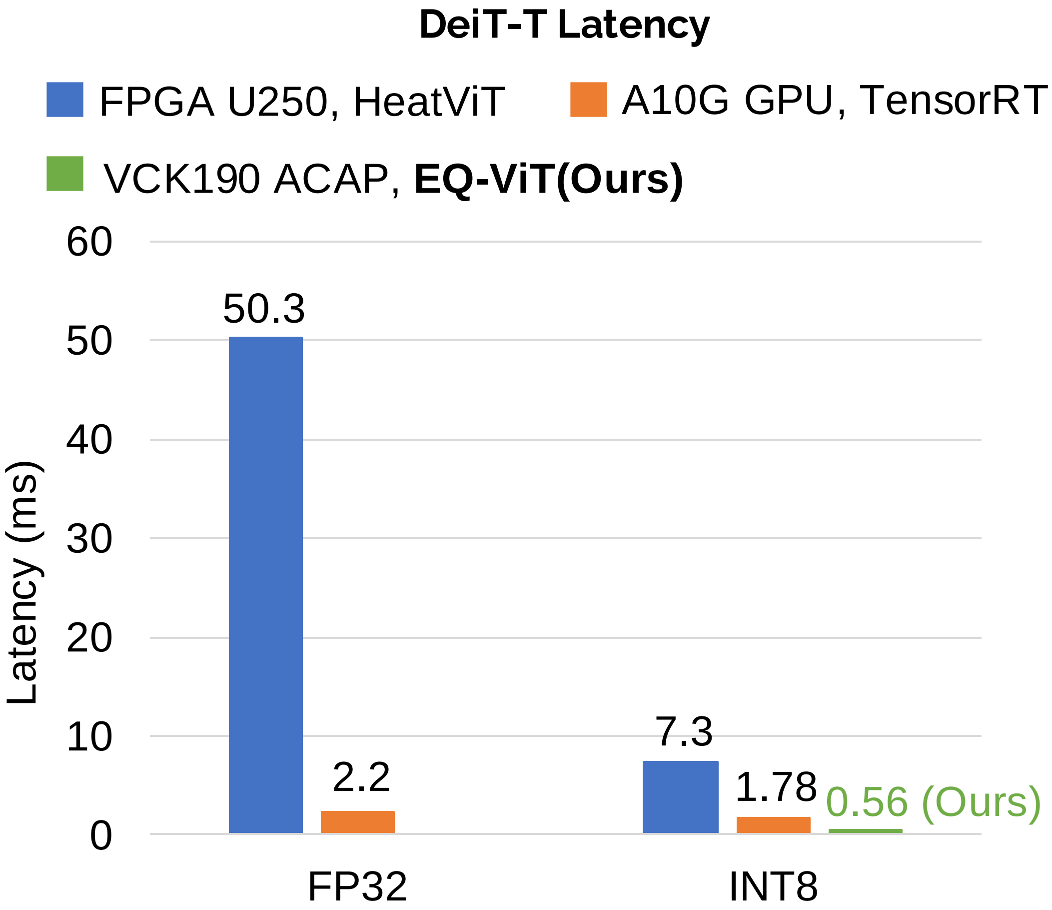
FPGA + Vector Processor?



Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s
AMD VCK190 7nm ACAP	6.4 T	102 T	25 GB/s

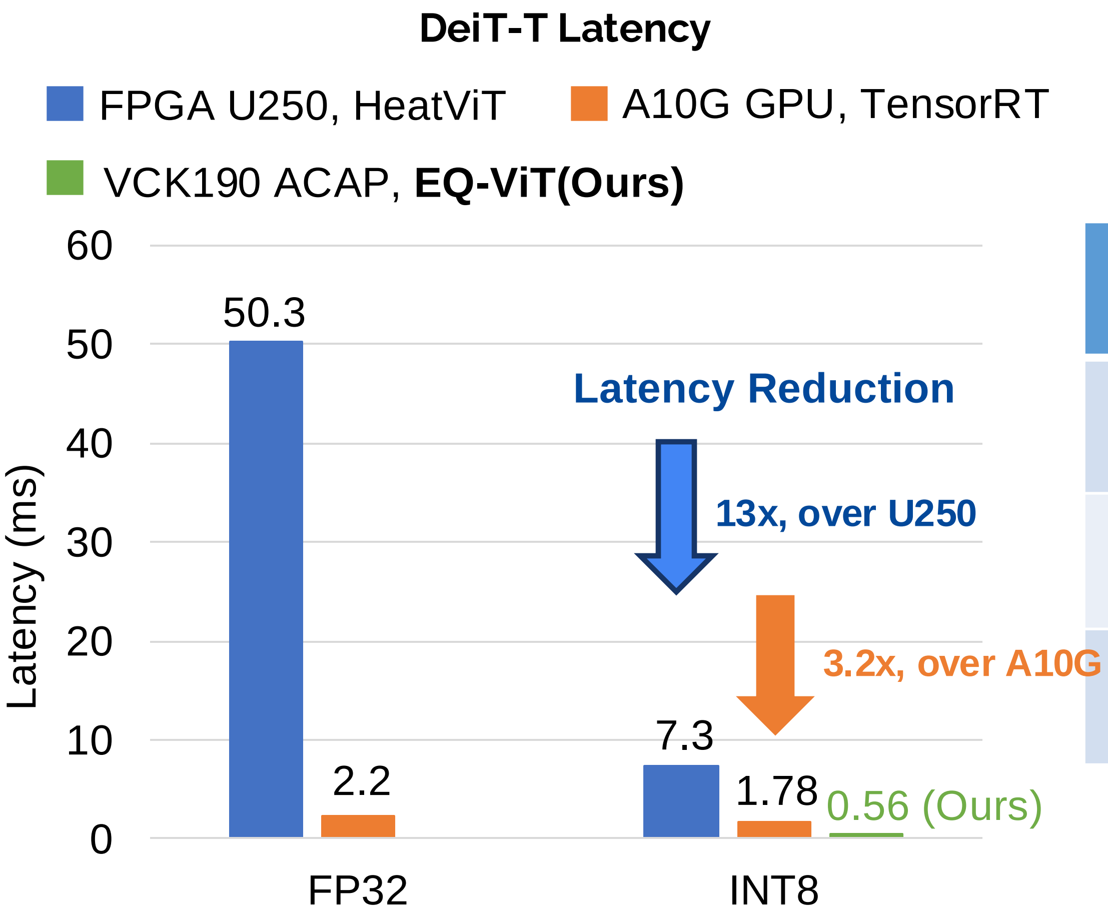
FPGA + Vector Processor?



Hardware Specification

Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s
AMD VCK190 7nm ACAP	6.4 T	102 T	25 GB/s

FPGA + Vector Processor?



Hardware Specification

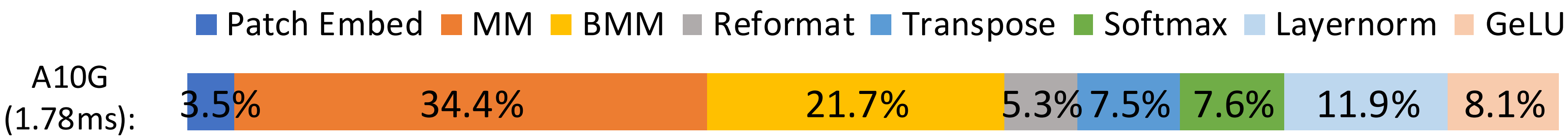
Platform	FP32	INT8	Off-Chip BW
NVIDIA A10G 8nm GPU	35 T	140 T	600 GB/s
AMD U250 16nm FPGA	1.2 T	6.95 T	77 GB/s
AMD VCK190 7nm ACAP	6.4 T	102 T	25 GB/s

Time Breakdown

- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T

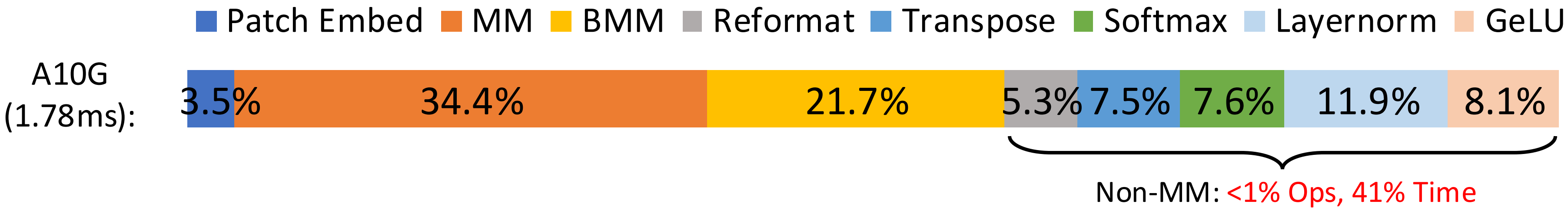
Time Breakdown

- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



Time Breakdown

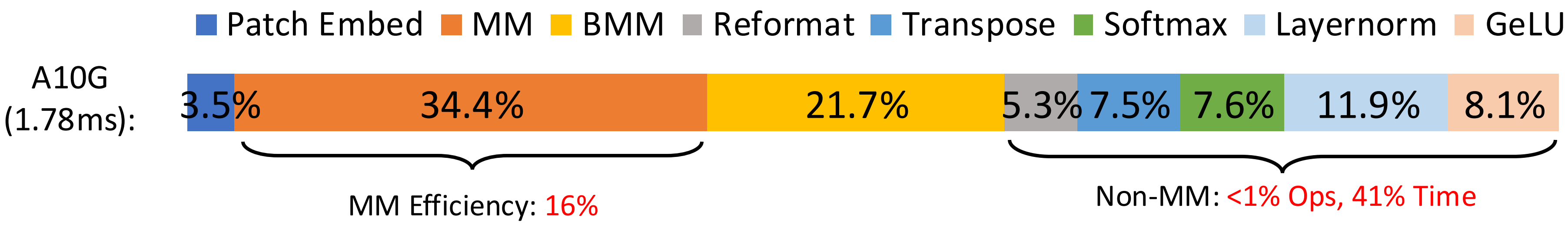
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time

Time Breakdown

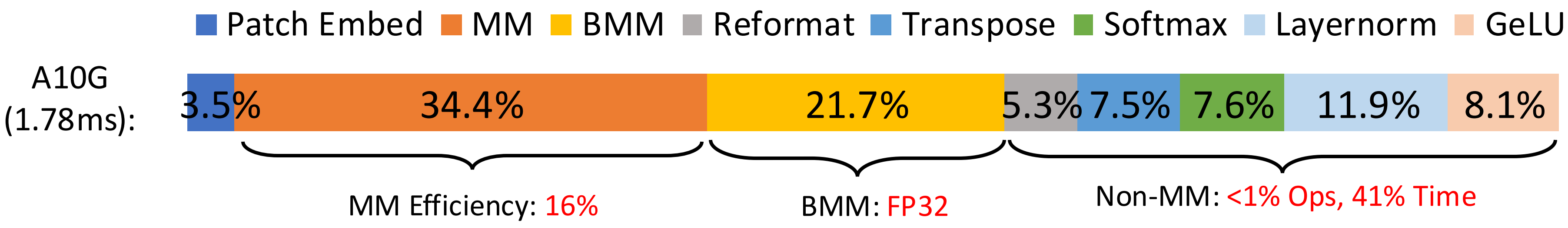
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time
- ❷ Tensor core utilization is not high enough (MM): ~23 TOPS, 16% of INT8 throughput (140TOPS)

Time Breakdown

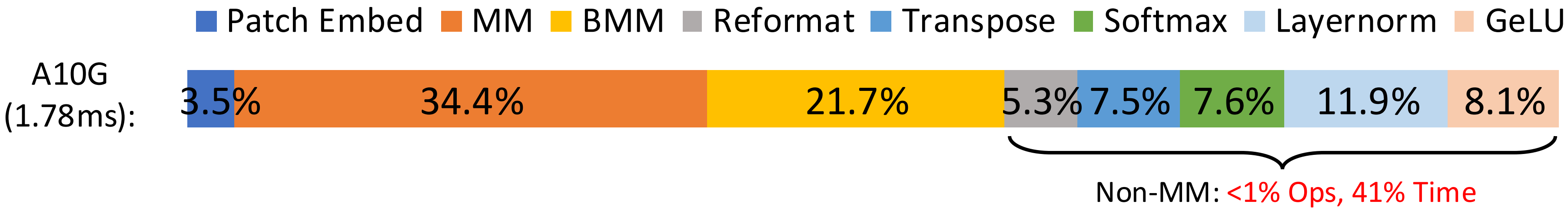
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time
- ❷ Tensor core utilization is not high enough (MM): ~23 TOPS, 16% of INT8 throughput (140TOPS)
- ❸ TensorRT adopts an implicit quantization policy: BMM FP32 data type

Time Breakdown

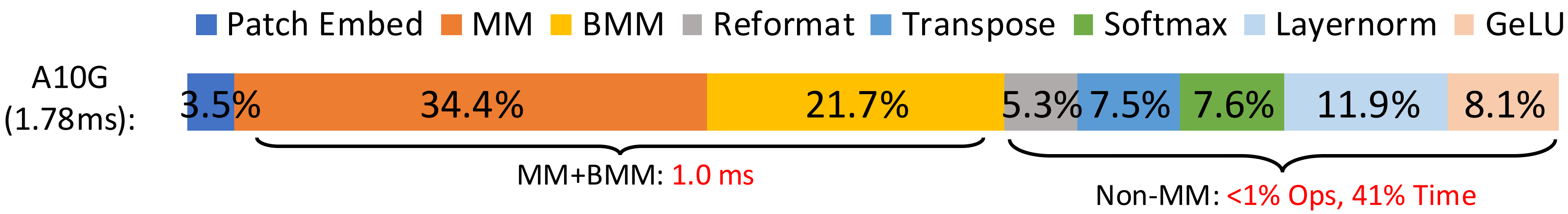
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time
- ❷ Tensor core utilization is not high enough (MM): ~23 TOPS, 16% of INT8 throughput (140TOPS)
- ❸ TensorRT adopts an implicit quantization policy: BMM FP32 data type

Time Breakdown

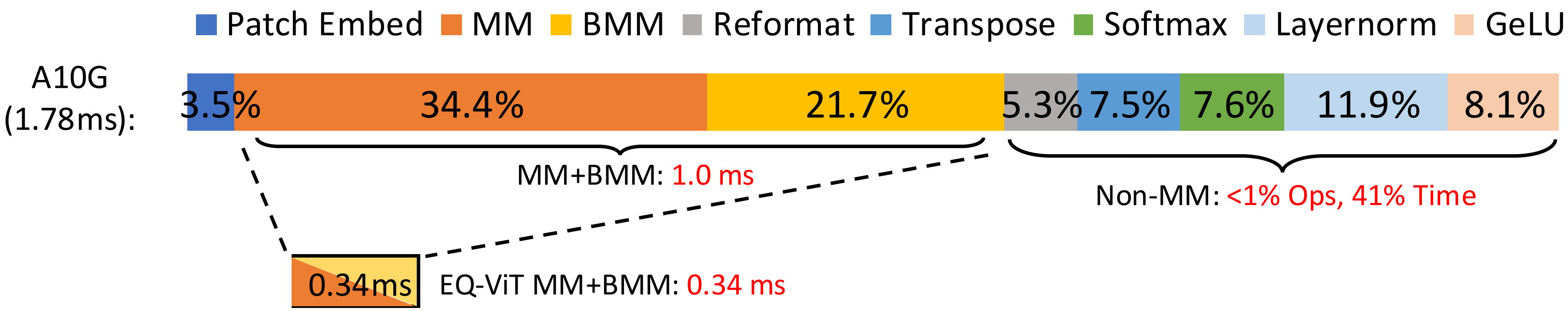
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time
- ❷ Tensor core utilization is not high enough (MM): ~23 TOPS, 16% of INT8 throughput (140TOPS)
- ❸ TensorRT adopts an implicit quantization policy: BMM FP32 data type

Time Breakdown

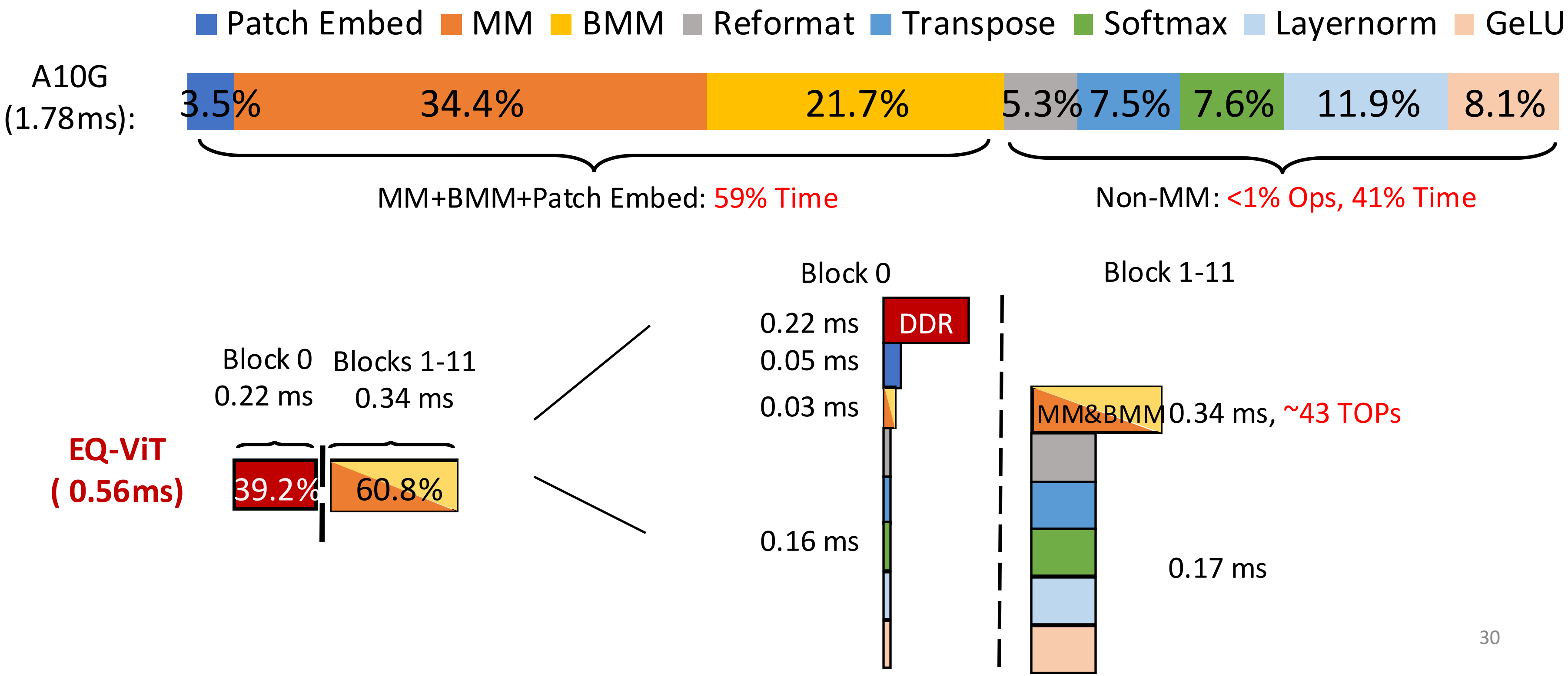
- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



- ❶ Non-MM kernels with <1% operations, take about 41% time
- ❷ Tensor core utilization is not high enough (MM): ~23 TOPS, 16% of INT8 throughput (140TOPS)
- ❸ TensorRT adopts an implicit quantization policy: BMM FP32 data type

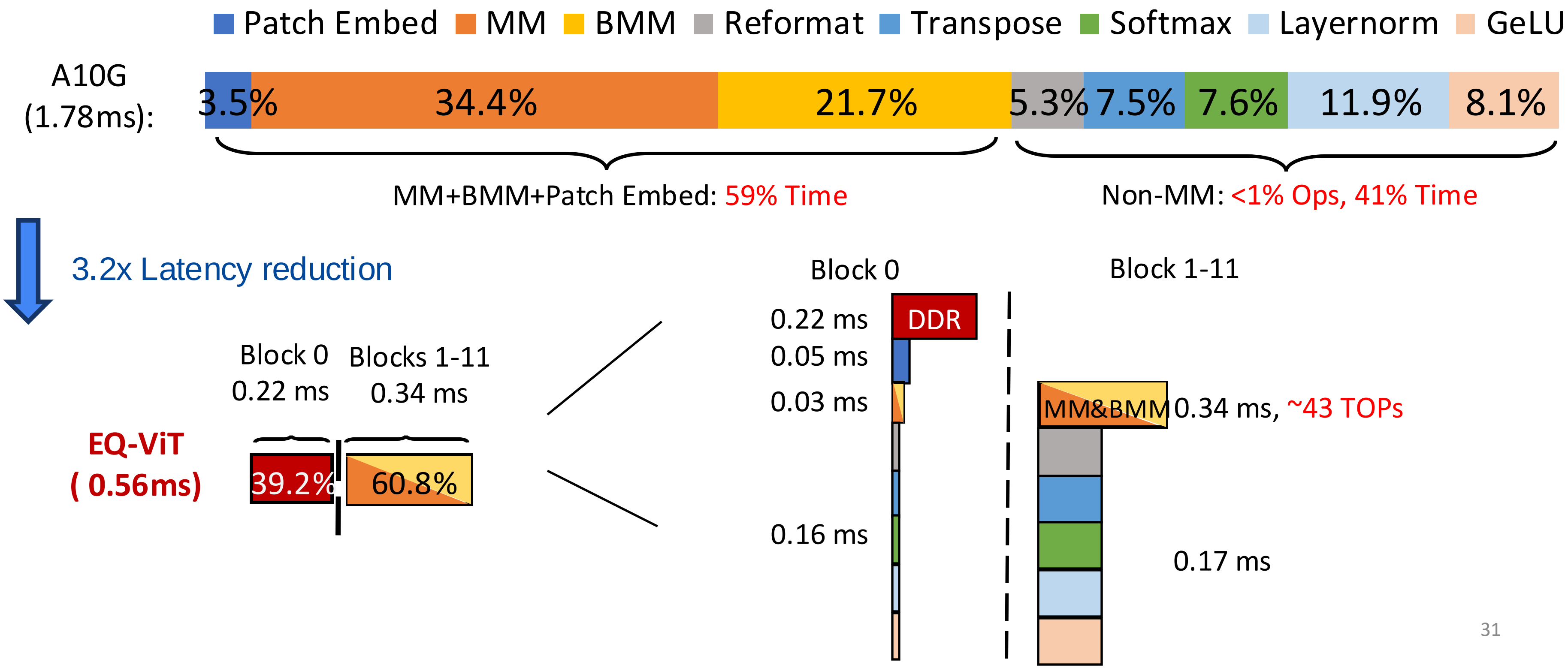
Time Breakdown

- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



Time Breakdown

- Time breakdown of EQ-ViT on Versal and TensorRT on A10G GPU for DeiT-T



EQ-ViT Framework Overview

EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

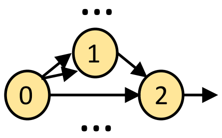
- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design

EQ-ViT Framework Overview

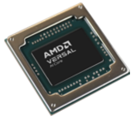
- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design

ViT Models



HW Capability

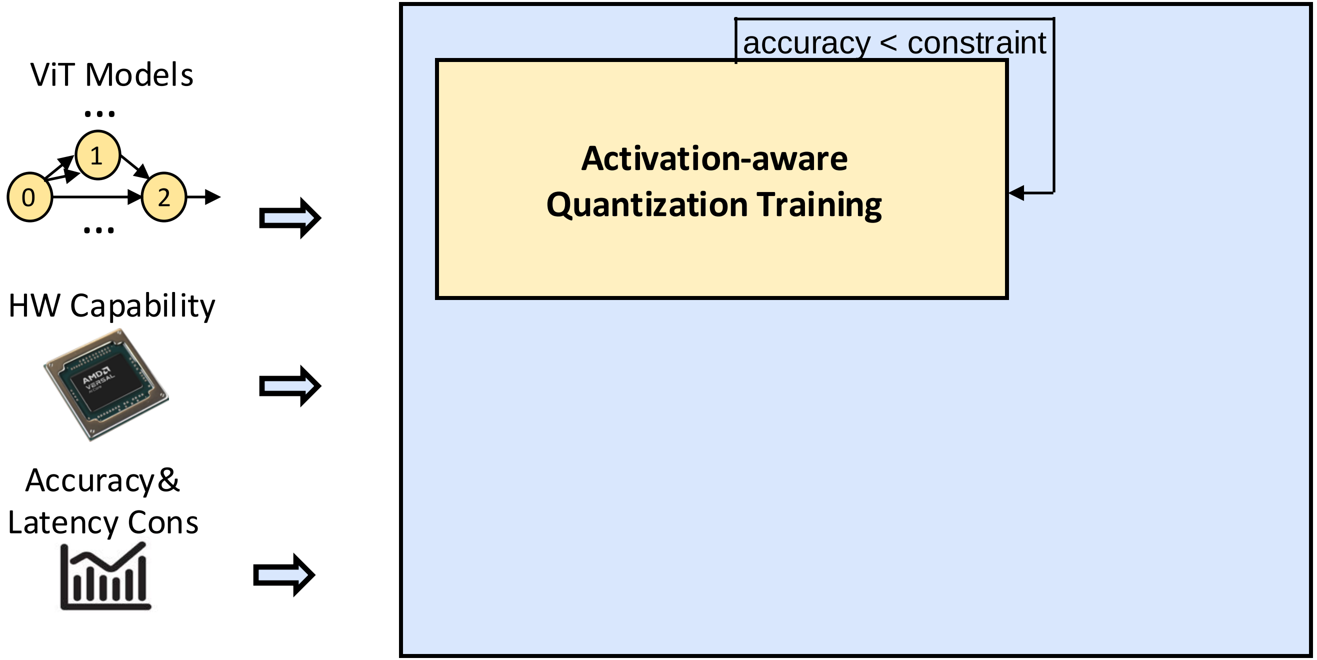


Accuracy&
Latency Cons

EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

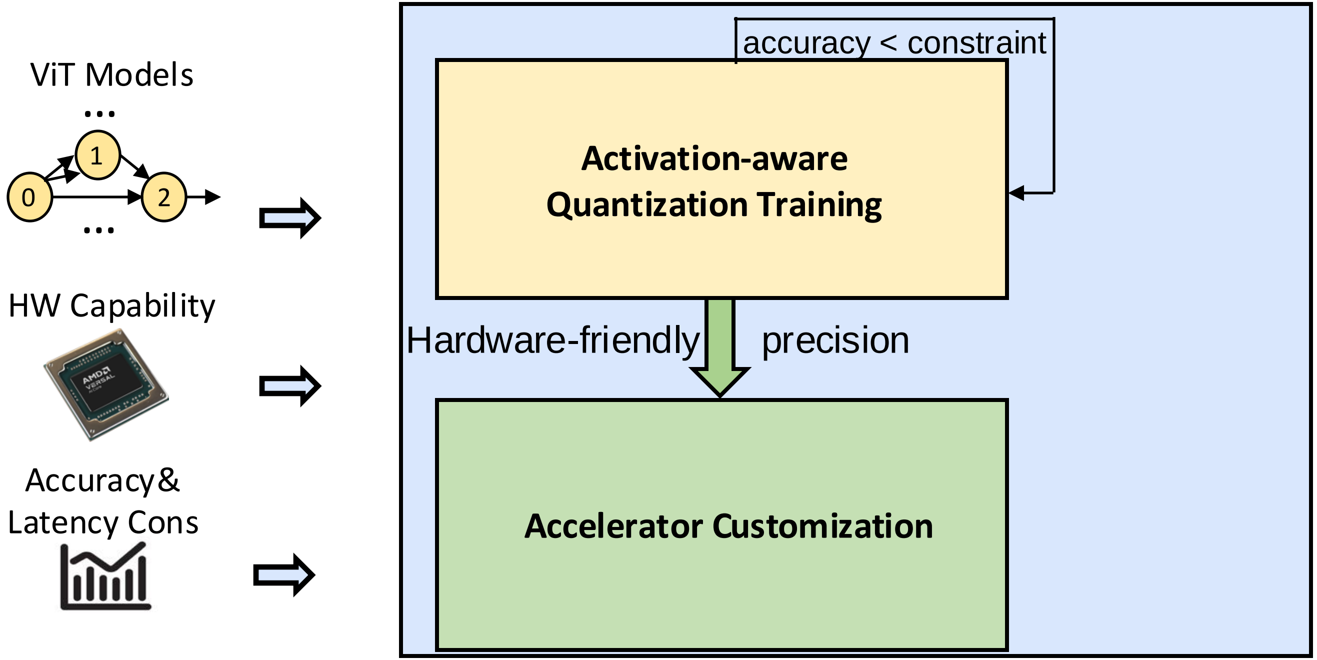
- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

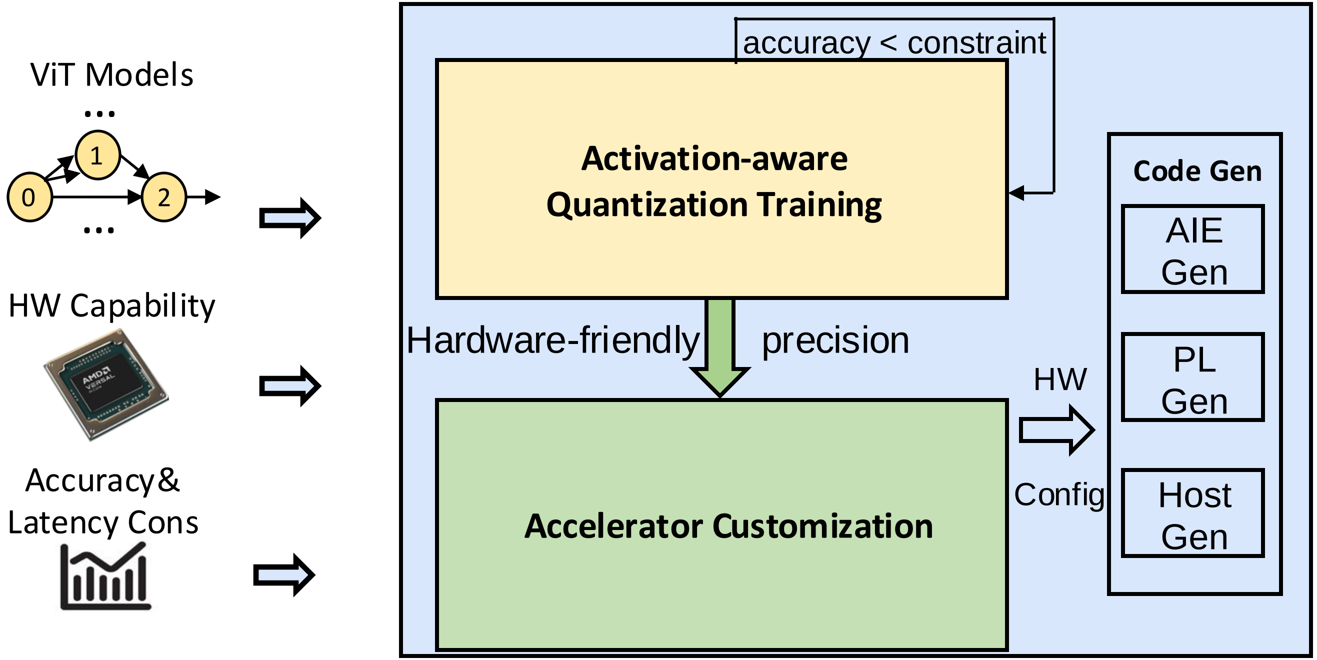
- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

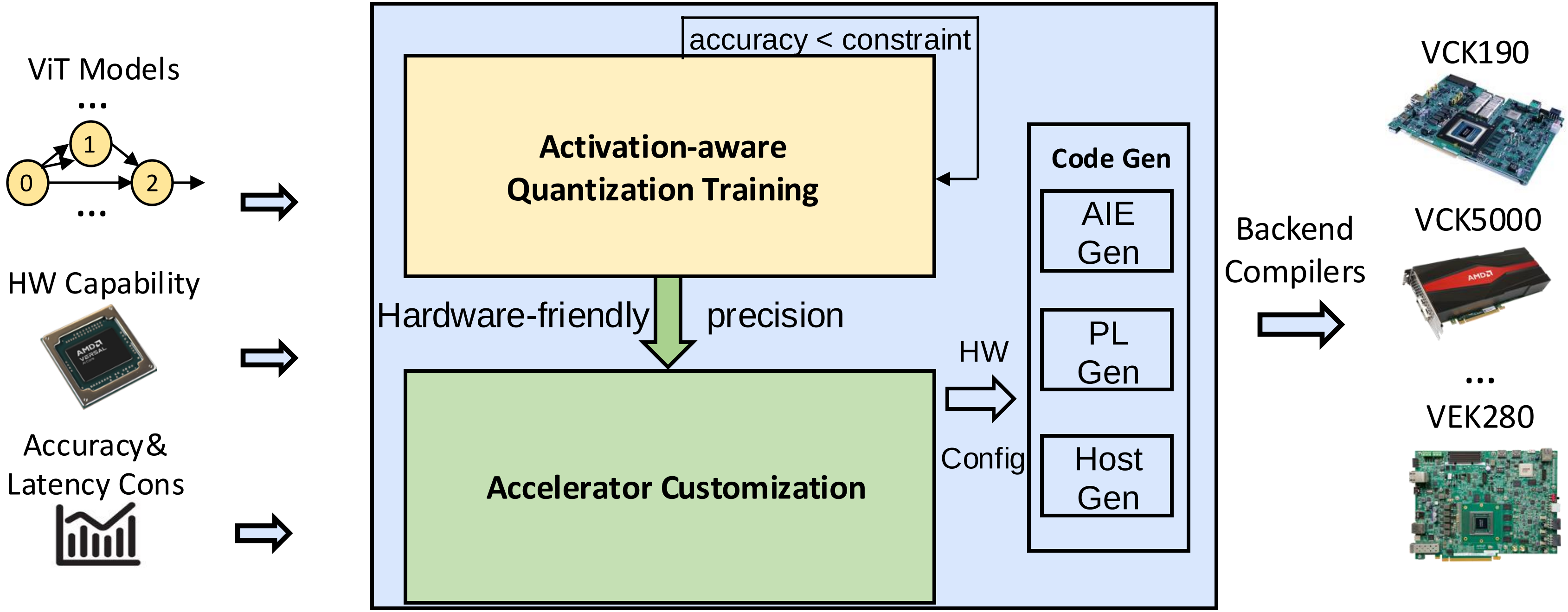
- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



EQ-ViT Framework Overview

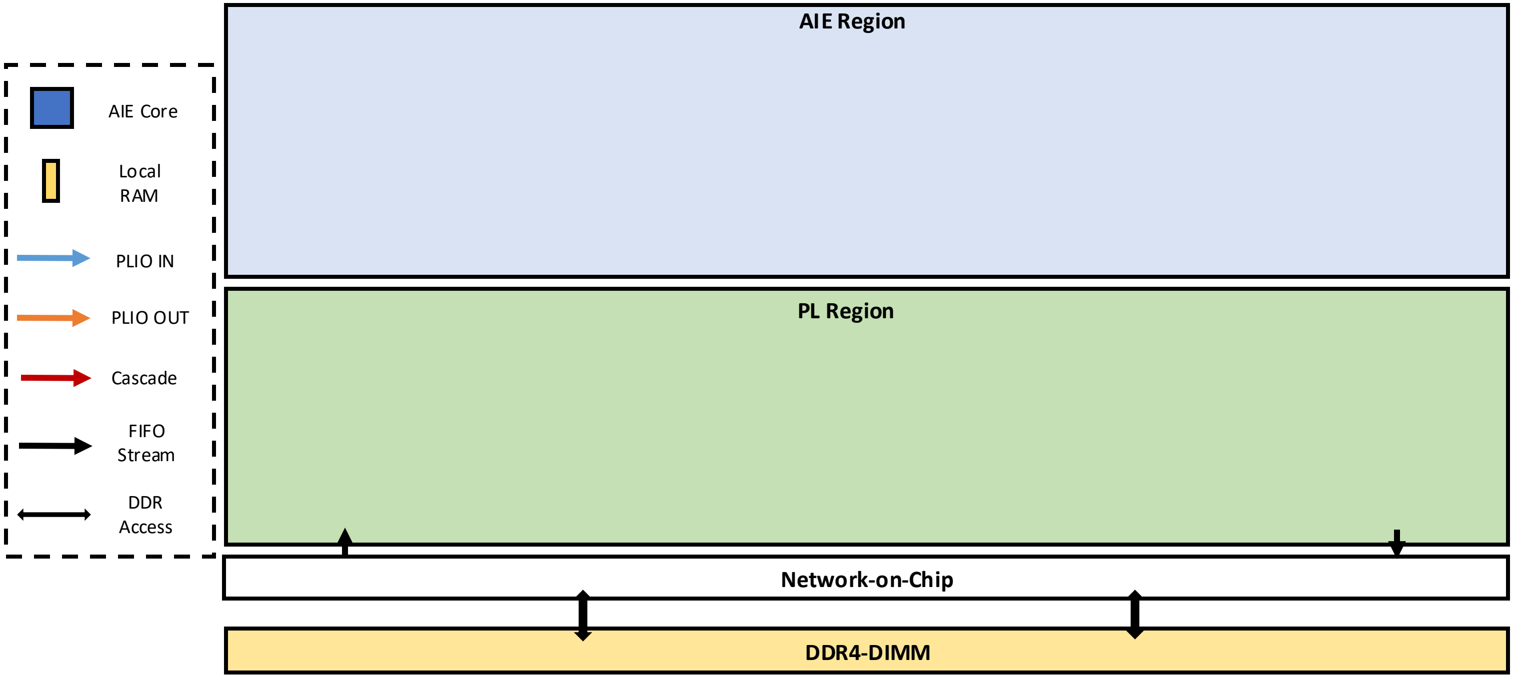
- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



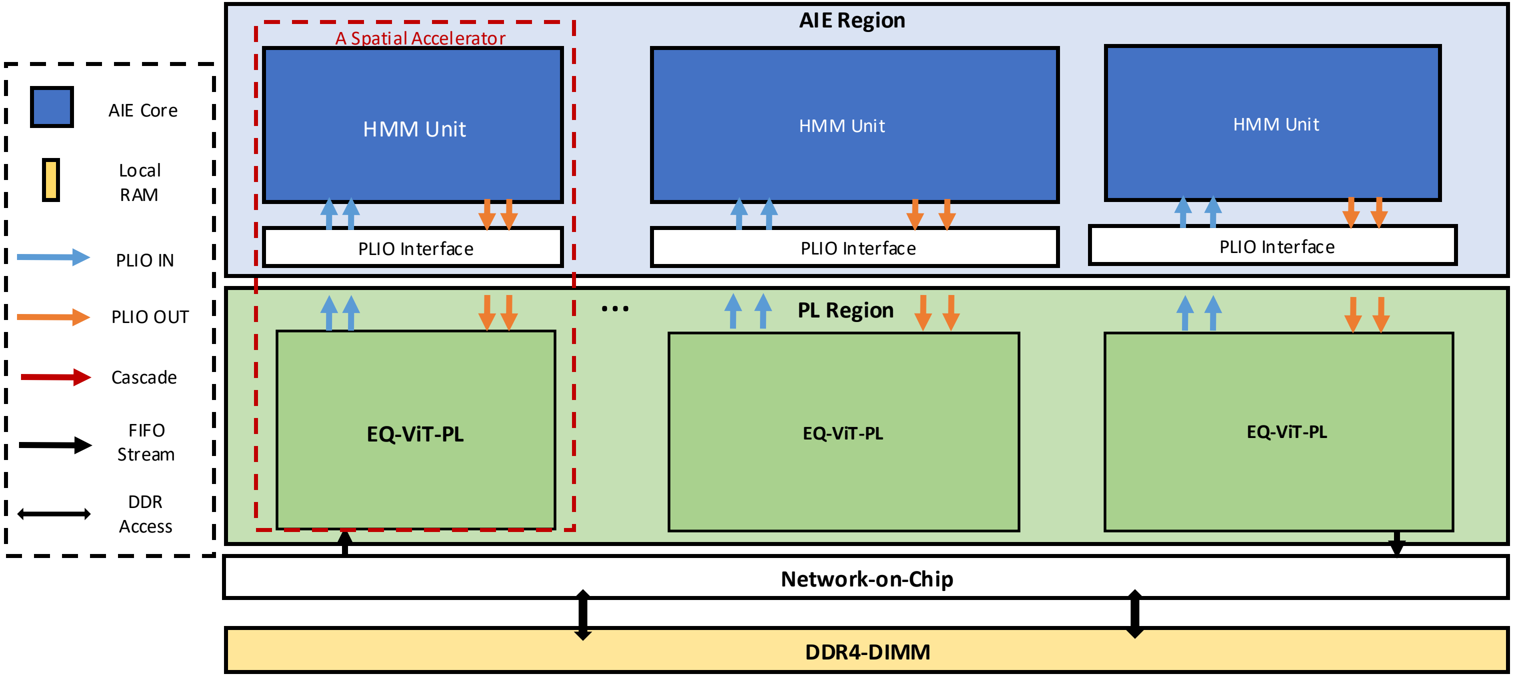
EQ-ViT Hardware Design Methodology

- Specialized MM kernel design



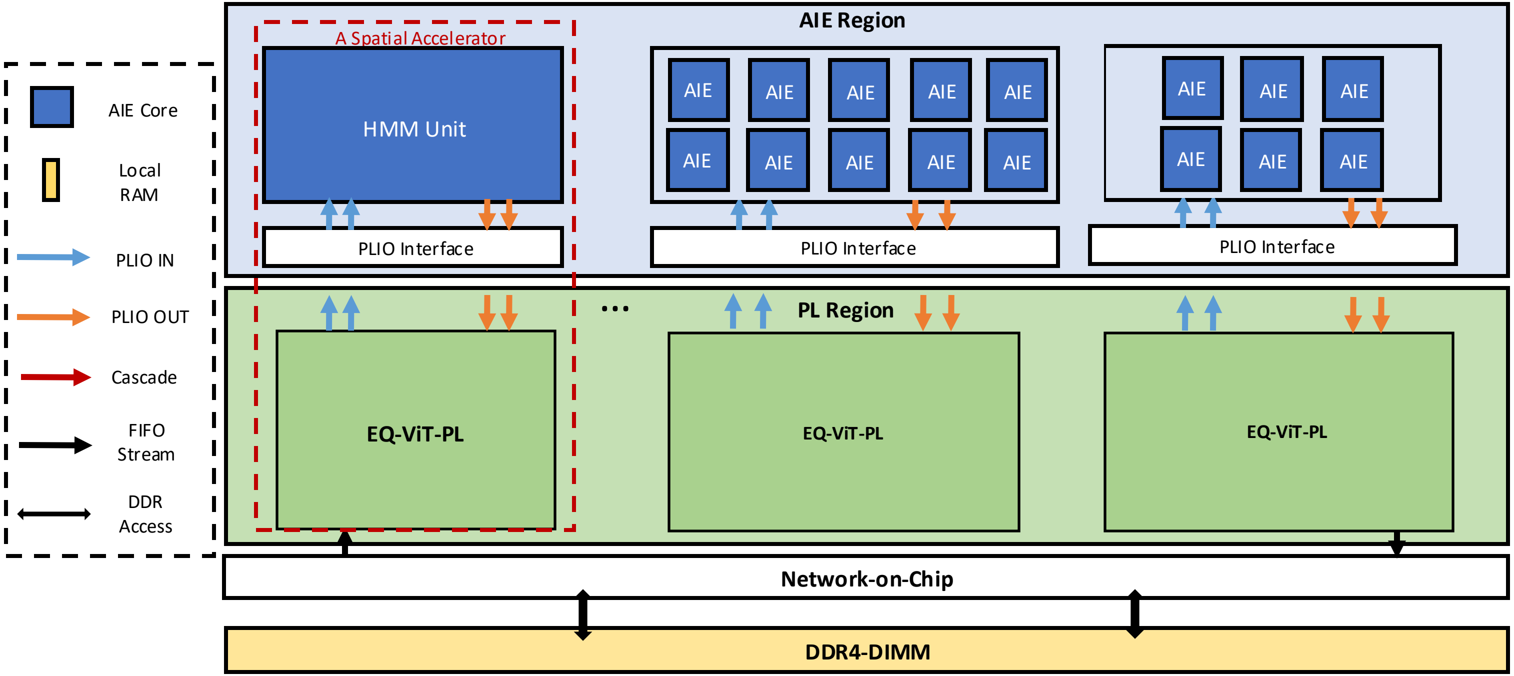
EQ-ViT Hardware Design Methodology

- Specialized MM kernel design



EQ-ViT Hardware Design Methodology

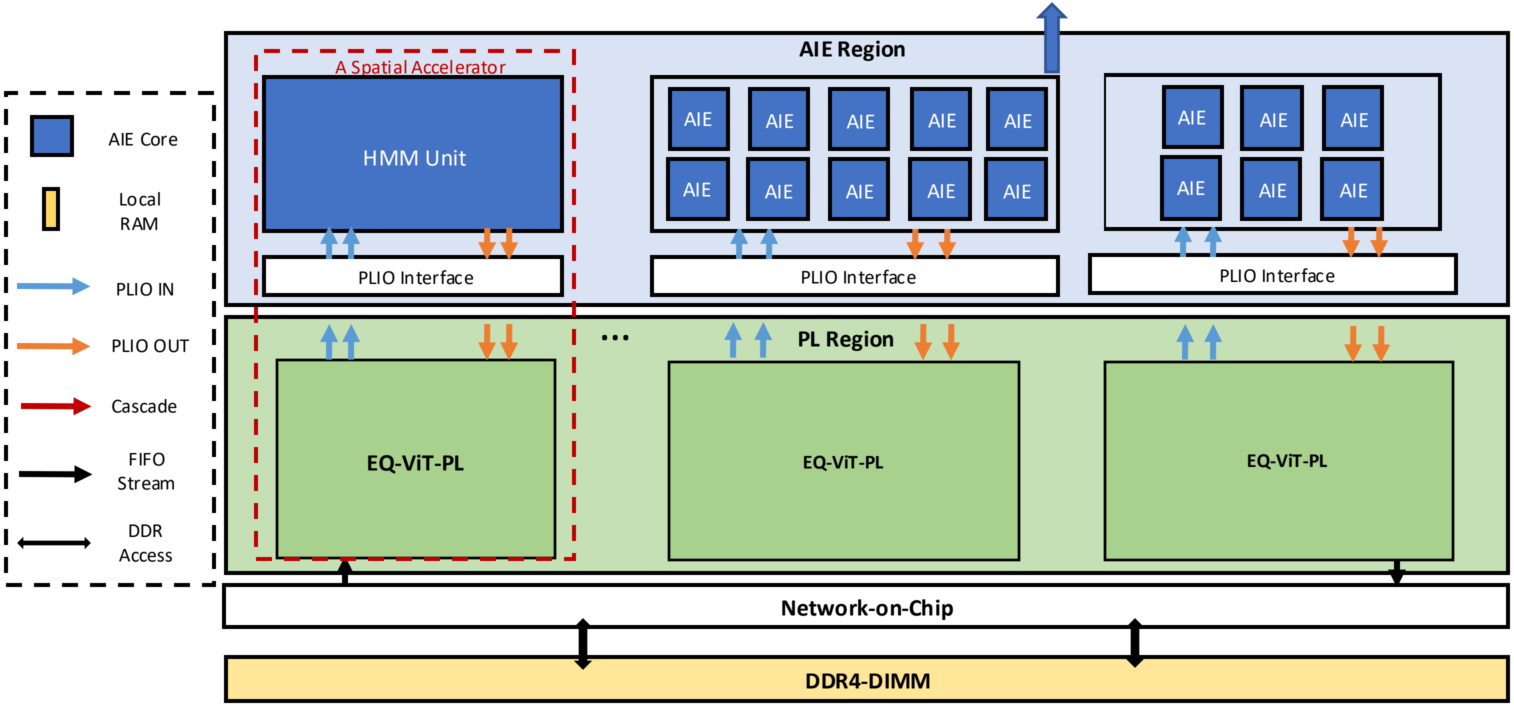
- Specialized MM kernel design



EQ-ViT Hardware Design Methodology

- Specialized MM kernel design

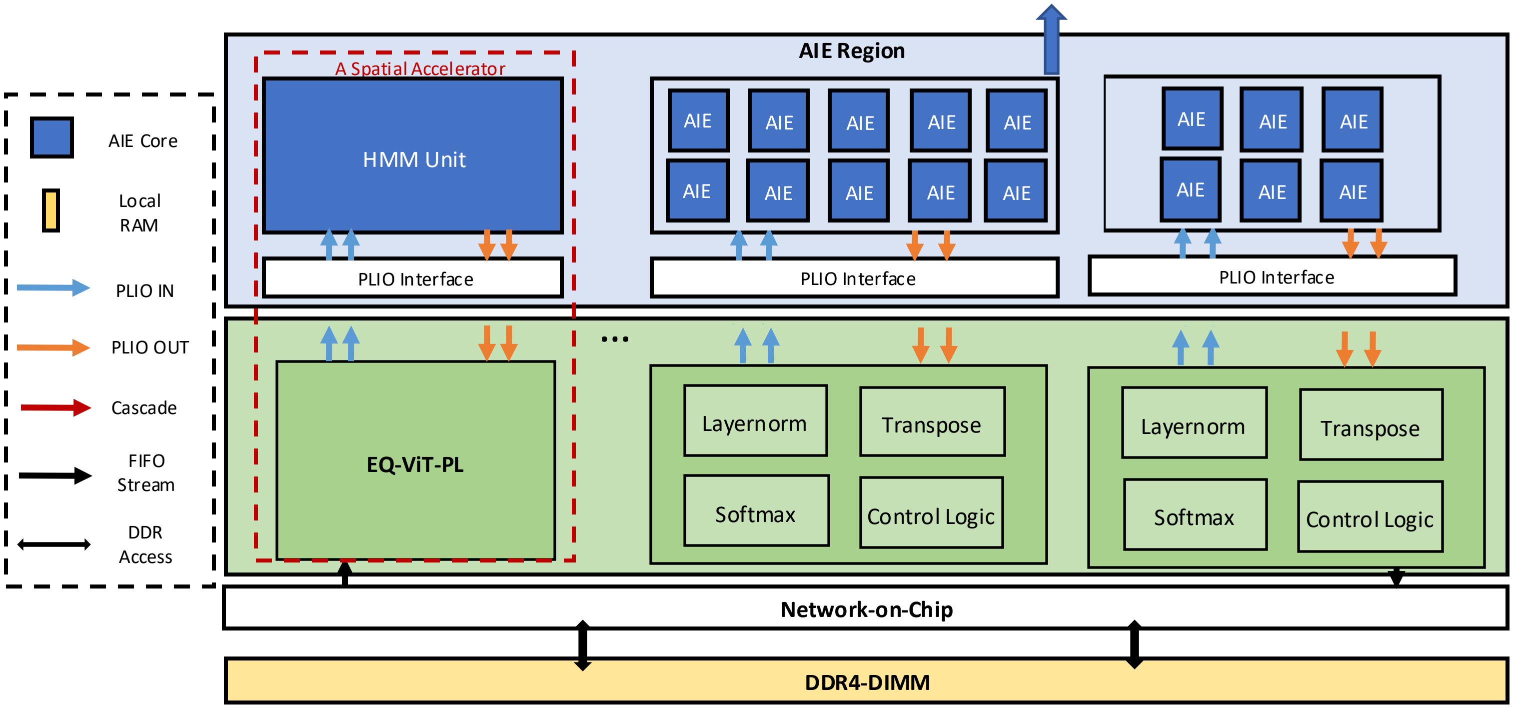
3D-Parallelism on two hierarchies: 3D-AIE Array (A, B, C), 3D-SIMD Instruction (PI, PK, PJ)



EQ-ViT Hardware Design Methodology

- Specialized MM kernel design

3D-Parallelism on two hierarchies: 3D-AIE Array (A, B, C), 3D-SIMD Instruction (PI, PK, PJ)



EQ-ViT Hardware Design Methodology

- Fine-grain pipelined non-MM kernel design

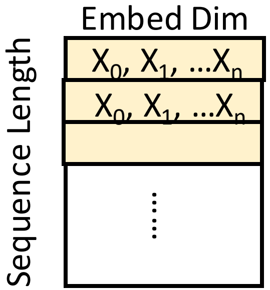
EQ-ViT Hardware Design Methodology

- Fine-grain pipelined non-MM kernel design
  - Non-linear Kernels

EQ-ViT Hardware Design Methodology

- Fine-grain pipelined non-MM kernel design

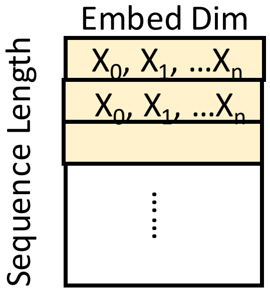
  - Non-linear Kernels



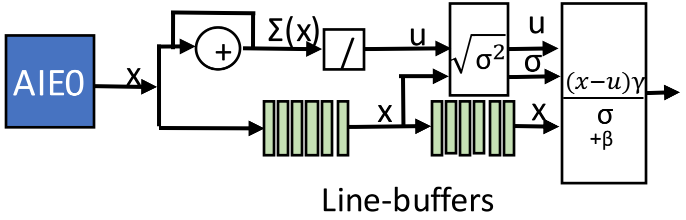
$$LN = \frac{(x-u)}{\sigma} * \gamma + \beta \qquad \sigma = \sqrt{\Sigma(x-u)^2} \qquad u = \frac{\Sigma(x)}{n}$$

EQ-ViT Hardware Design Methodology

- Fine-grain pipelined non-MM kernel design
  - Non-linear Kernels

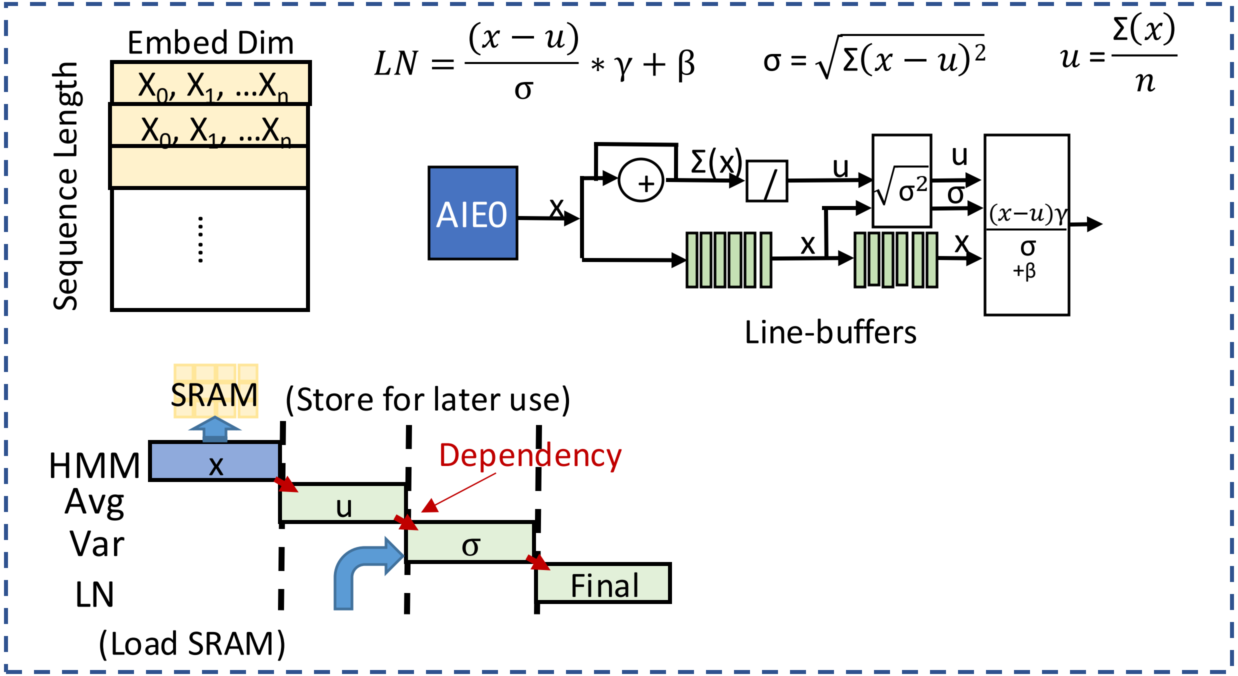


$$LN = \frac{(x-u)}{\sigma} * \gamma + \beta \qquad \sigma = \sqrt{\Sigma(x-u)^2} \qquad u = \frac{\Sigma(x)}{n}$$



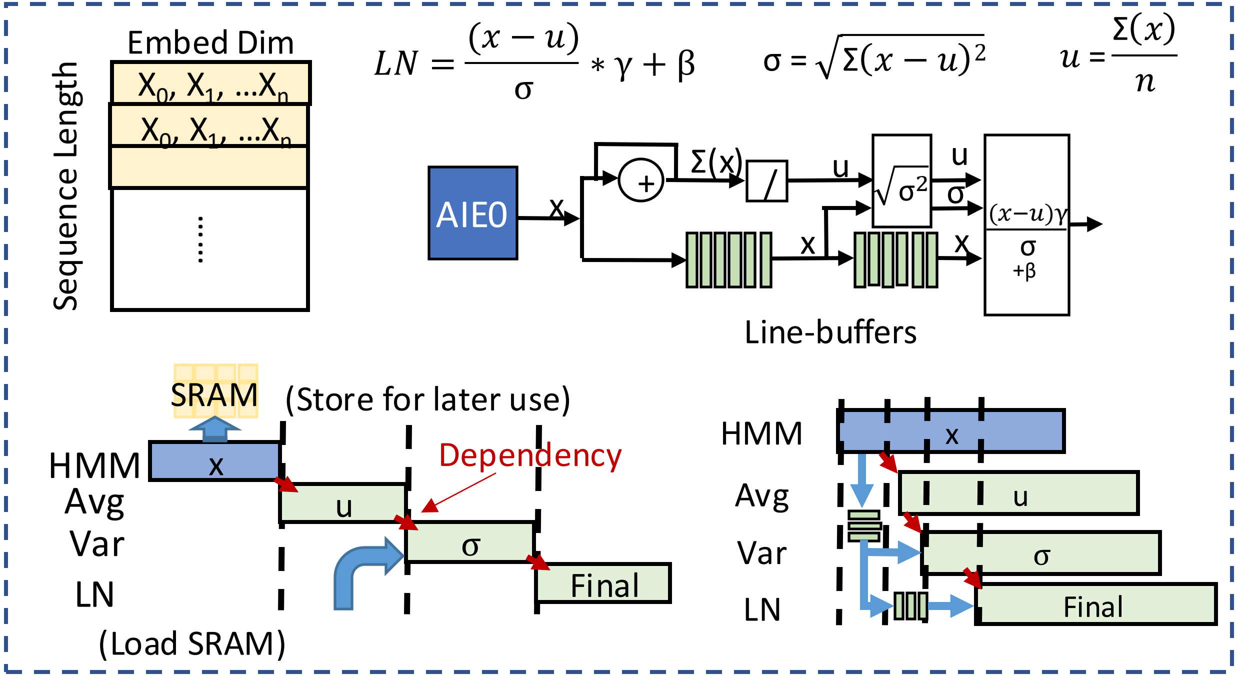
EQ-ViT Hardware Design Methodology

- Fine-grain pipelined non-MM kernel design
  - Non-linear Kernels



EQ-ViT Hardware Design Methodology

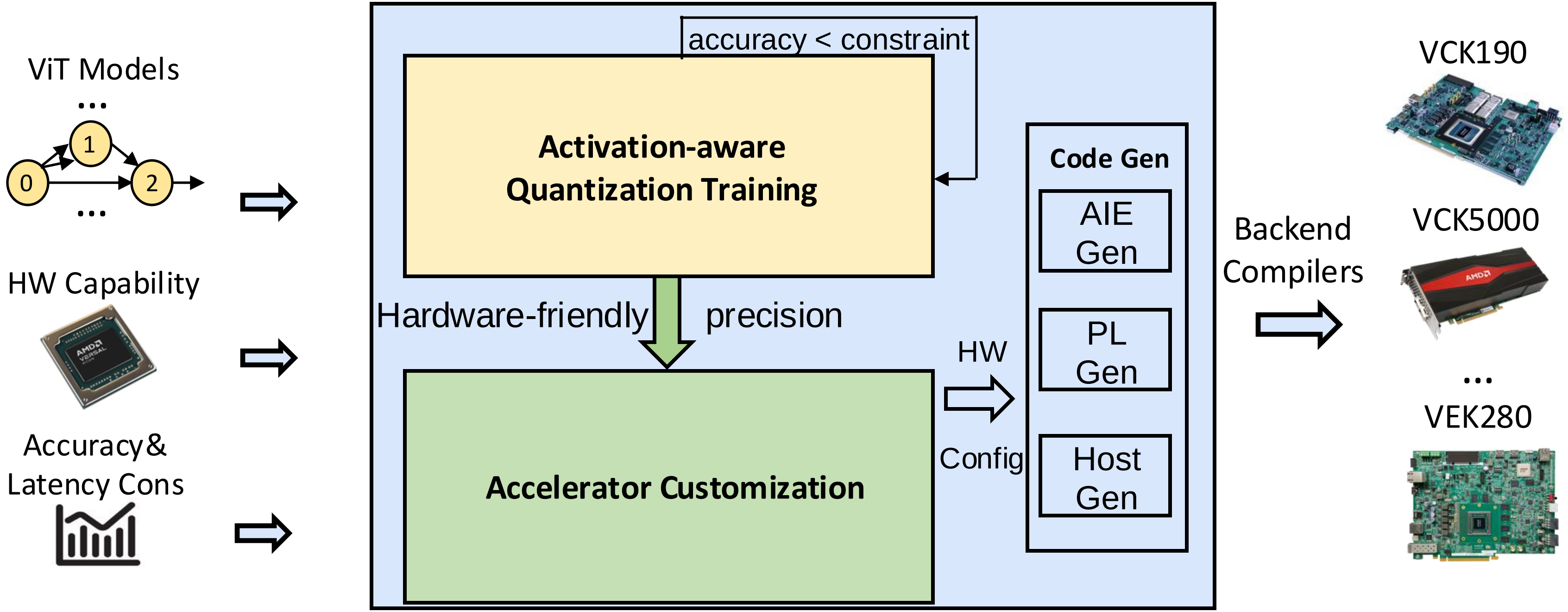
- Fine-grain pipelined non-MM kernel design
  - Non-linear Kernels



EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

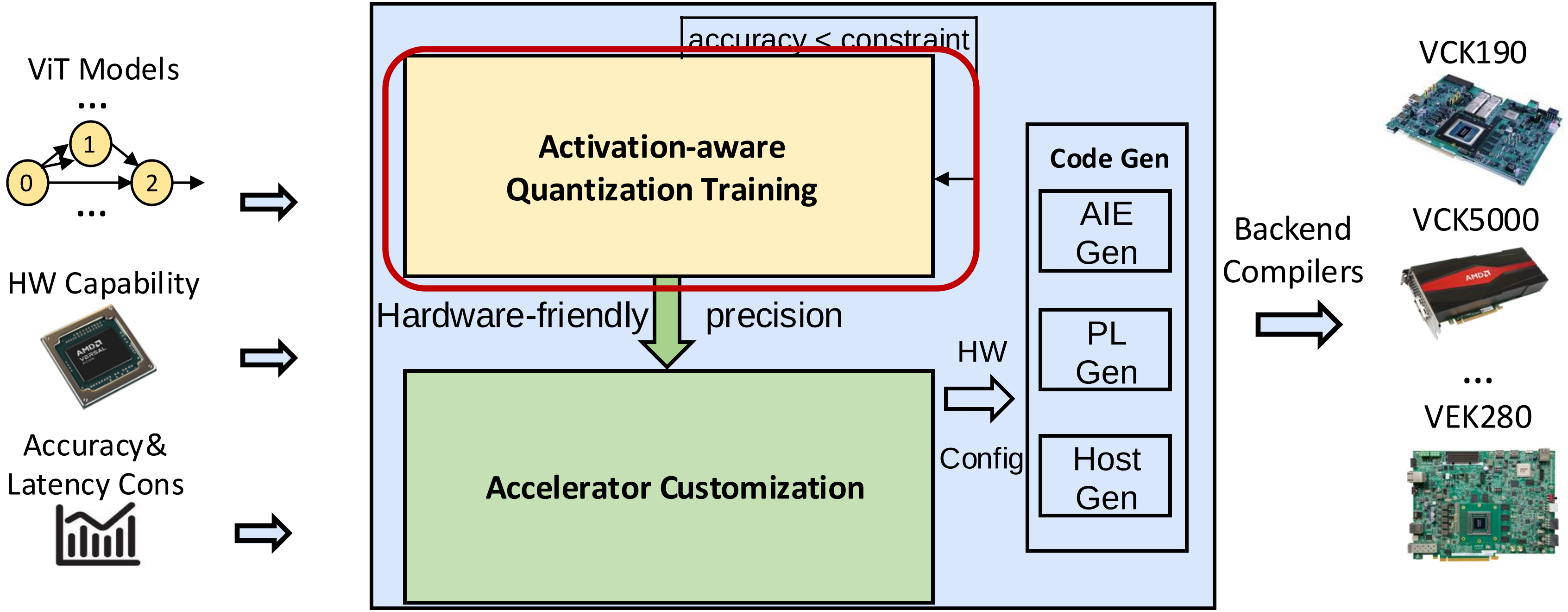
- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



EQ-ViT Framework Overview

- **Inputs**
  1) Transformer models
  2) Accuracy constraint
  3) Latency constraint
  4) Hardware constraints

- **Outputs**
  1) Quantization strategy
  2) Hardware optimization strategy
  3) Automatic generated hardware design



Can We Quantize ViTs into low-bit (e.g. 8) for enhanced Accuracy?

Can We Quantize ViTs into low-bit (e.g. 8) for enhanced Accuracy?

Quantization Algorithm:

- ViT Quantization
  - No papers quantize ViTs into 8-bit with higher acc

Method	#Bits	DeiT-T [43]	DeiT-S [43]	DeiT-B [43]	Swin-T [33]	Swin-S [33]
Full Precision	32/32/32	72.21	79.85	81.85	81.35	83.2
PTQ						
MinMax	8/8/8	70.94	75.05	78.02	64.38	74.37
EMA	8/8/8	71.17	75.71	78.82	70.81	75.05
Percentile	8/8/8	71.47	76.57	78.37	78.78	78.12
OMSE	8/8/8	71.3	75.03	79.57	79.3	78.96
Bit-Split	8/8/8	–	77.06	79.42	–	–
PTQ for ViT	8/8/8	–	77.47	80.48	–	–
FQ-ViT	8/8/8	71.61	79.17	81.2	80.51	82.71

Can We Quantize ViTs into low-bit (e.g. 8) for enhanced Accuracy?

EMBEDDED SYSTEMS WEEK

Quantization Algorithm:

- ViT Quantization
  - No papers quantize ViTs into 8-bit with higher acc

Method	#Bits	DeiT-T [43]	DeiT-S [43]	DeiT-B [43]	Swin-T [33]	Swin-S [33]
Full Precision	32/32/32	72.21	79.85	81.85	81.35	83.2
PTQ						
MinMax	8/8/8	70.94	75.05	78.02	64.38	74.37
EMA	8/8/8	71.17	75.71	78.82	70.81	75.05
Percentile	8/8/8	71.47	76.57	78.37	78.78	78.12
OMSE	8/8/8	71.3	75.03	79.57	79.3	78.96
Bit-Split	8/8/8	–	77.06	79.42	–	–
PTQ for ViT	8/8/8	–	77.47	80.48	–	–
FQ-ViT	8/8/8	71.61	79.17	81.2	80.51	82.71

We analyze ViT's data distribution to figure it out.

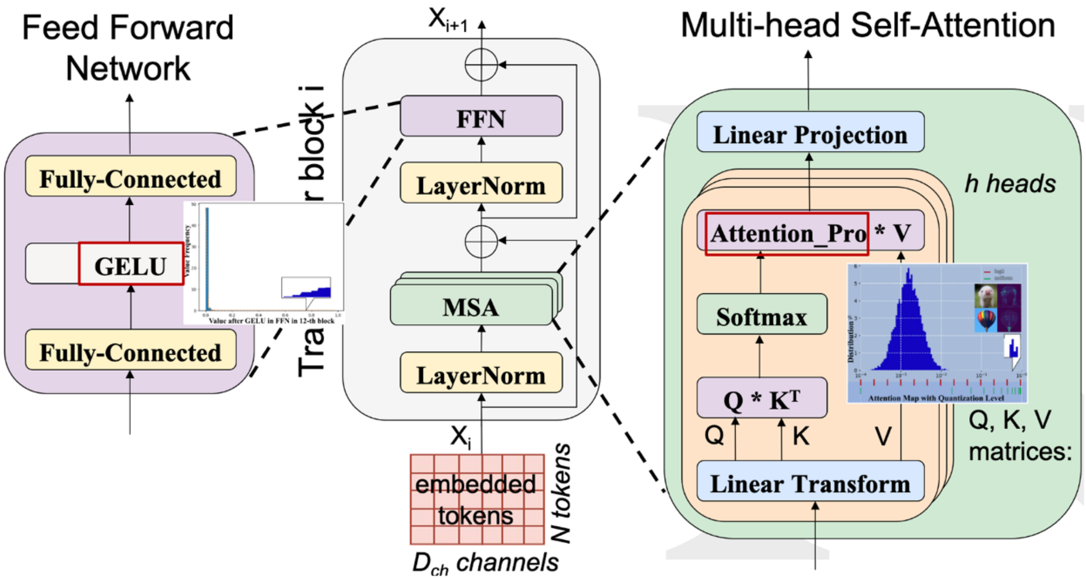
EQ-ViT Data Analysis

- Two Special Data Distribution inside ViTs
  - Long-Tail Distribution

EQ-ViT Data Analysis

Long-Tail Distribution: Attention Matrix & Act After GELU

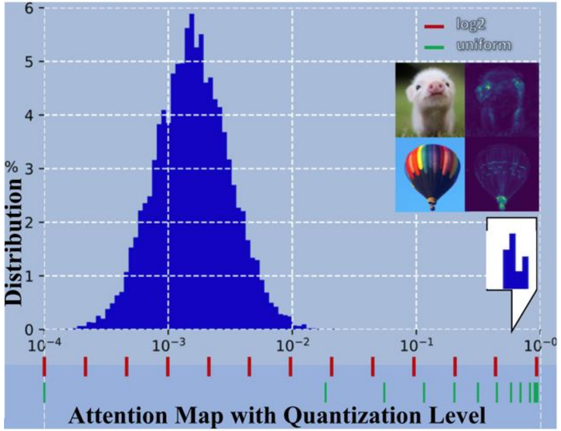
- Two Special Data Distribution inside ViTs
  - Long-Tail Distribution

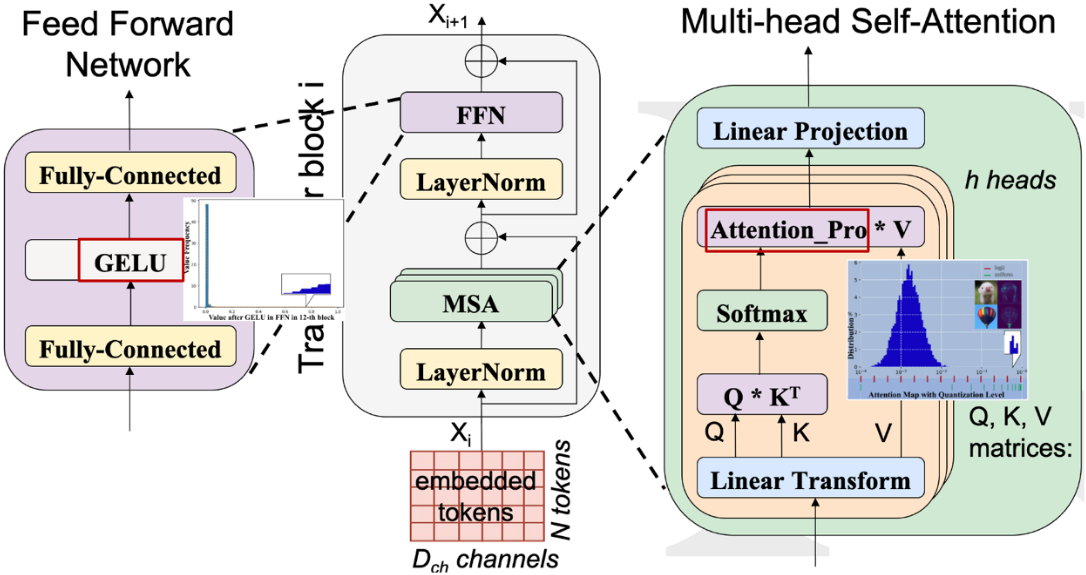


EQ-ViT Data Analysis

Long-Tail Distribution: Attention Matrix & Act After GELU

- Two Special Data Distribution inside ViTs
  - Long-Tail Distribution





EQ-ViT Data Analysis

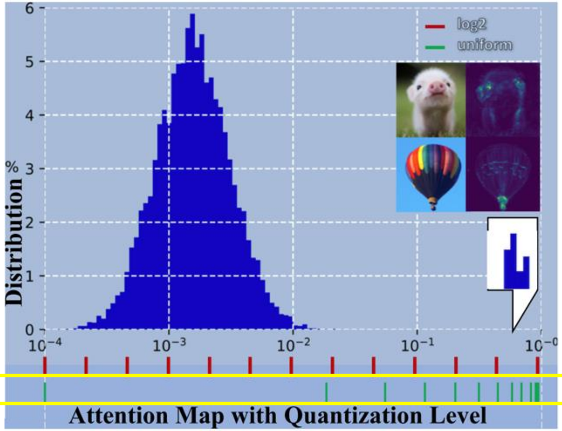
Long-Tail Distribution: Attention Matrix & Act After GELU

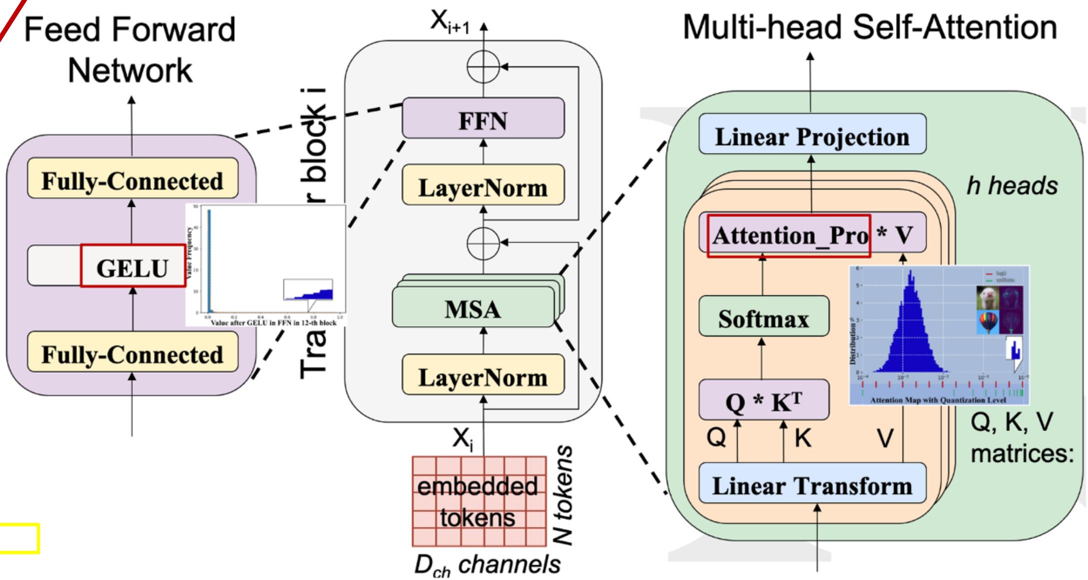
- Two Special Data Distribution inside ViTs
  - Long-Tail Distribution

Using by Many Works…

Uniform Quantization

One Value





EQ-ViT Data Analysis

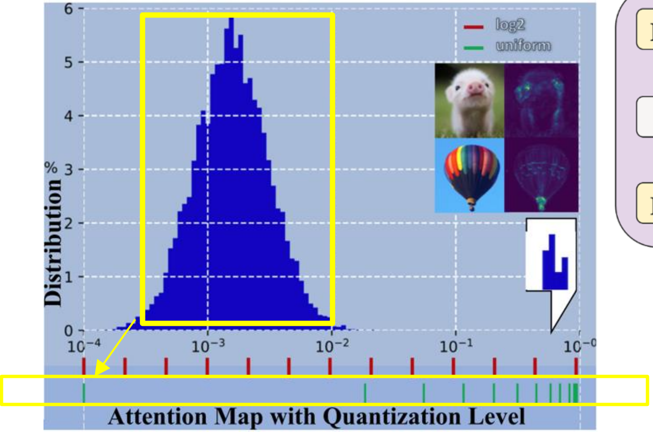
Long-Tail Distribution: Attention Matrix & Act After GELU

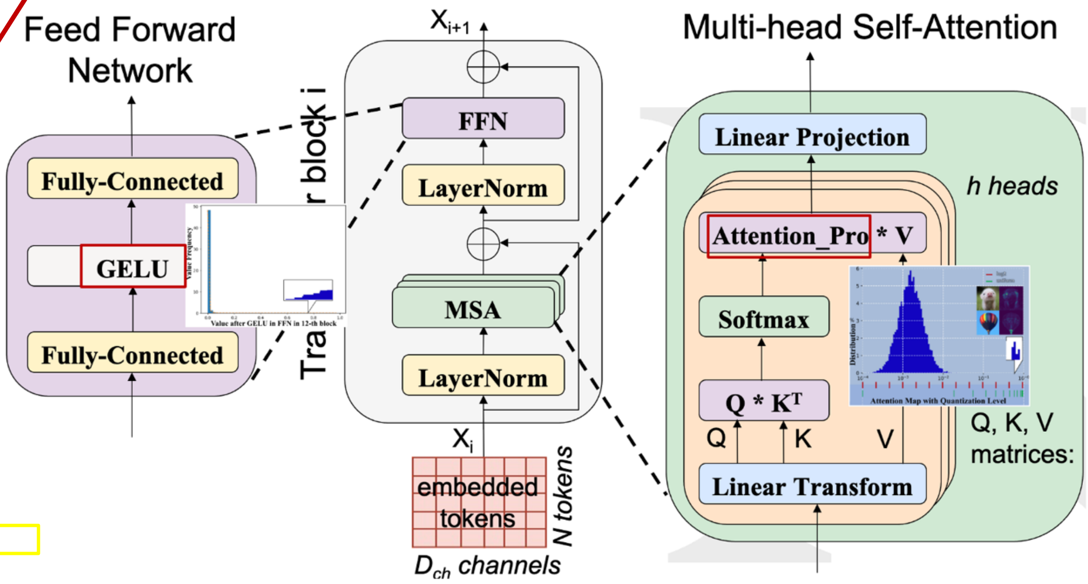
- Two Special Data Distribution inside ViTs
  - Long-Tail Distribution

Using by Many Works…

Uniform Quantization

One Value





EQ-ViT Data Analysis

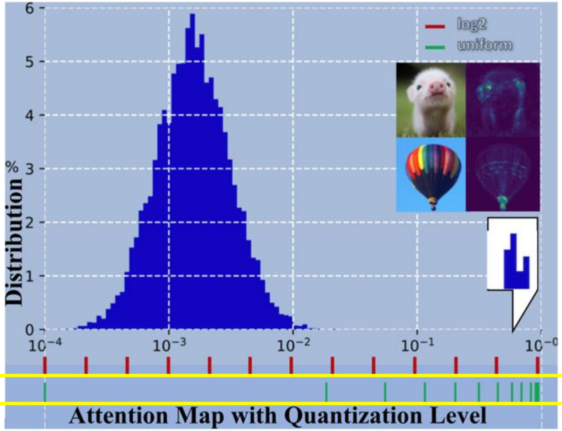
Long-Tail Distribution: Attention Matrix & Act After GELU

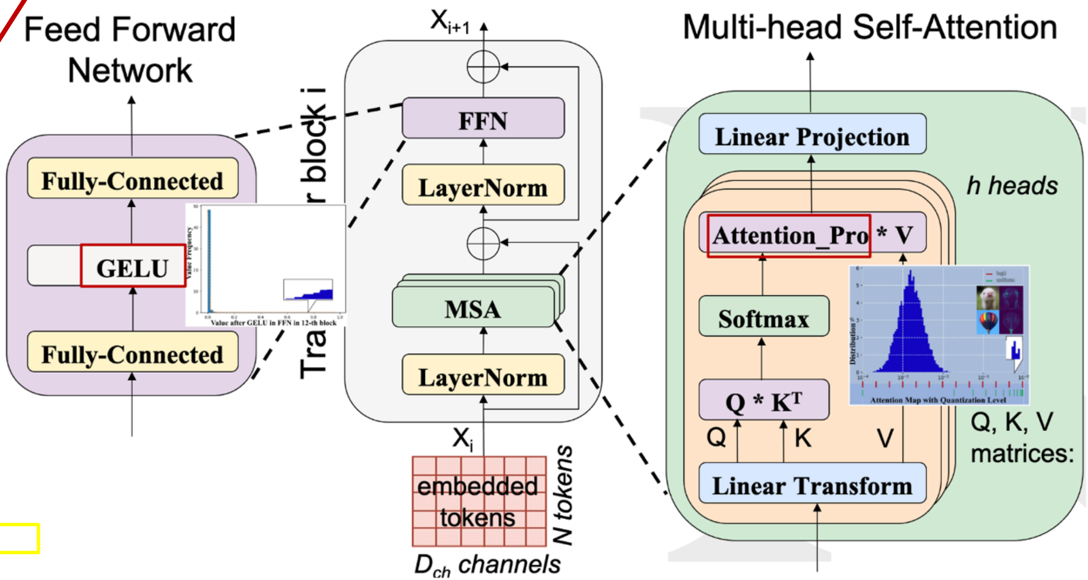
- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution

Using by Many Works…

Uniform Quantization

Information Loss



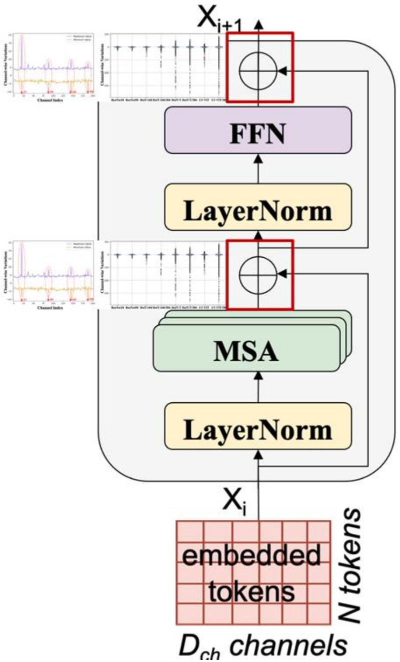


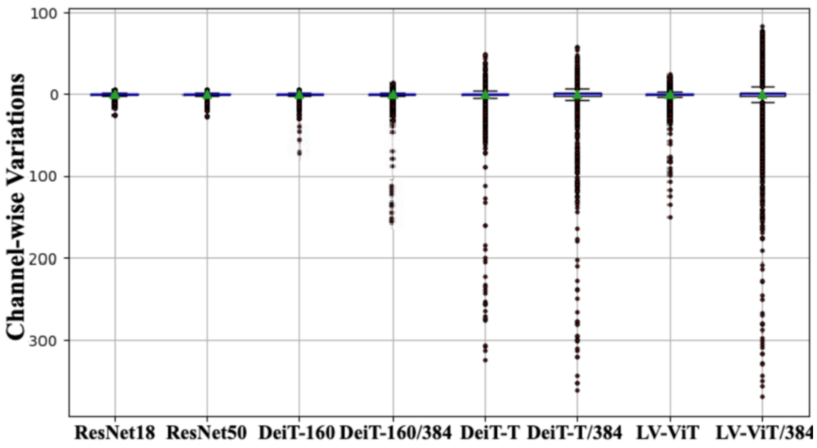
EQ-ViT Data Analysis

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

EQ-ViT Data Analysis

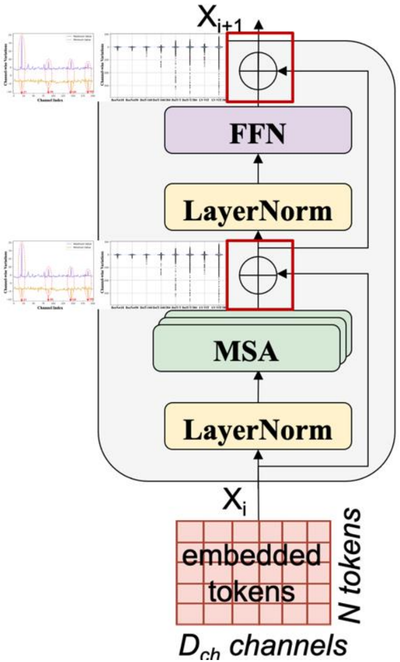
- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

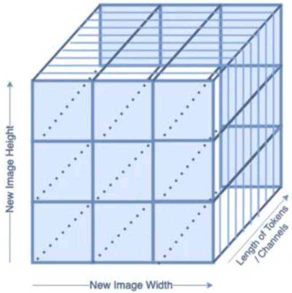


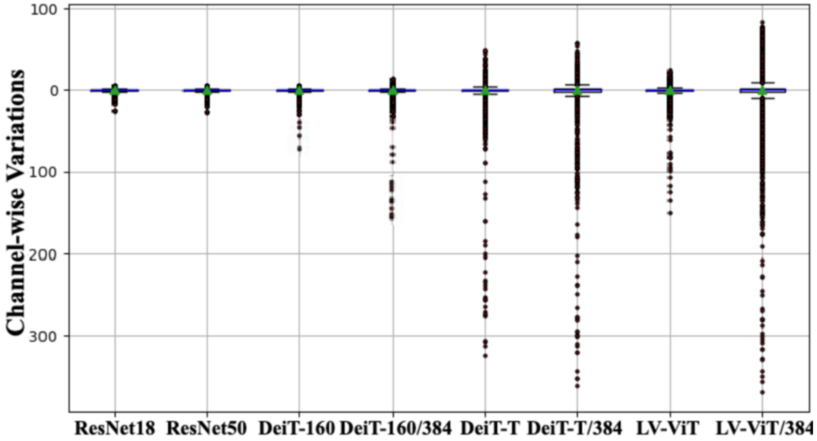


EQ-ViT Data Analysis

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

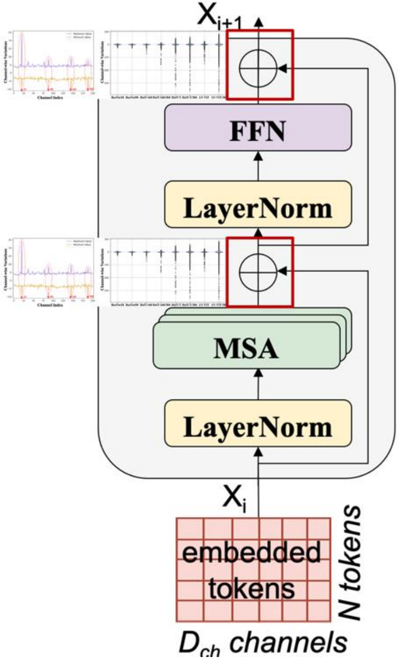


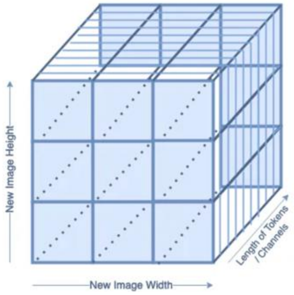


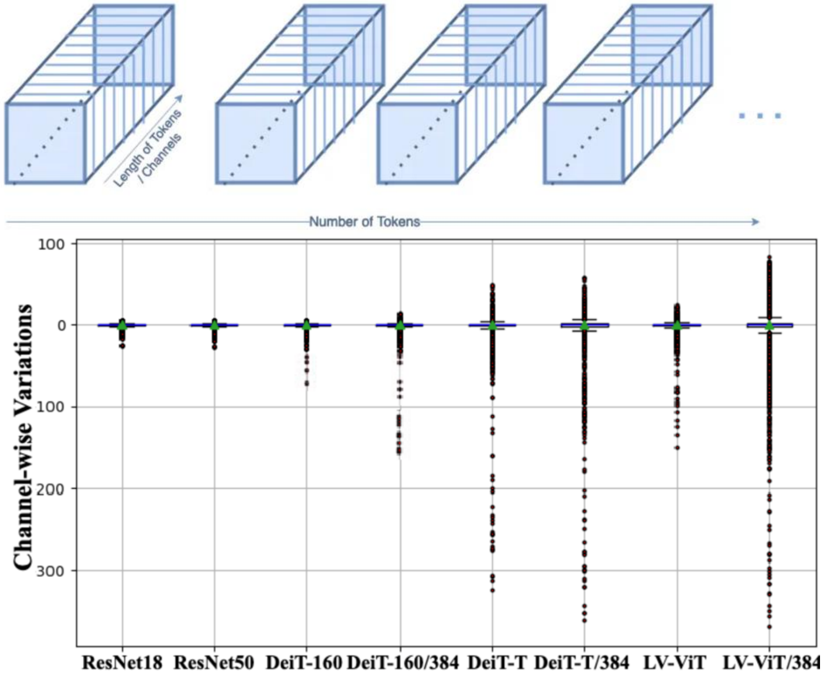


EQ-ViT Data Analysis

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

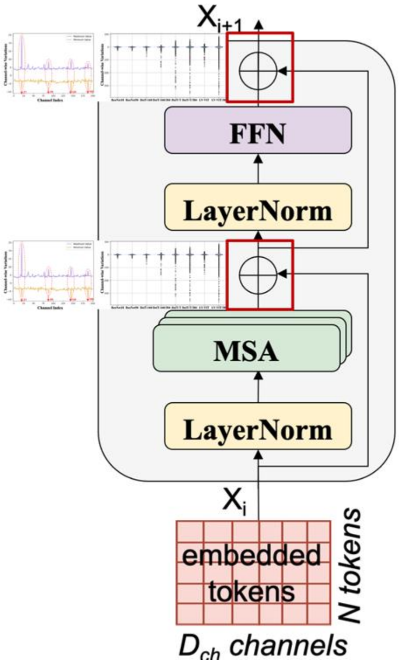


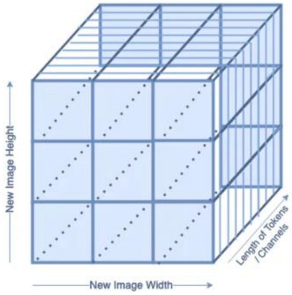


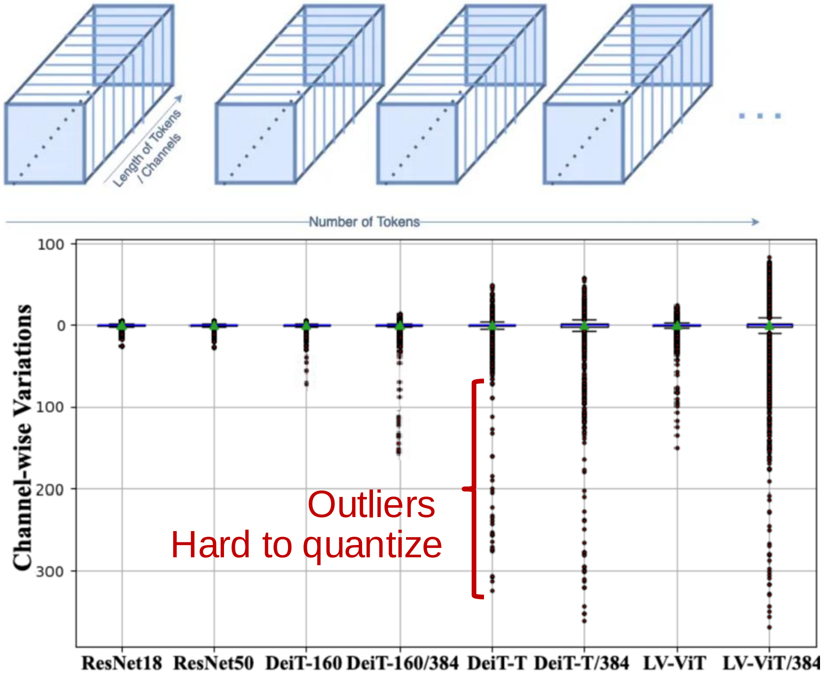


EQ-ViT Data Analysis

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers



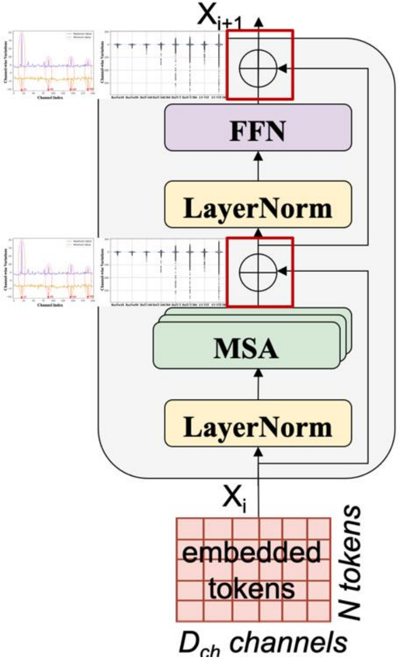


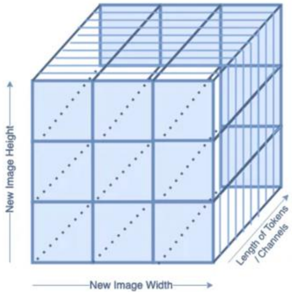


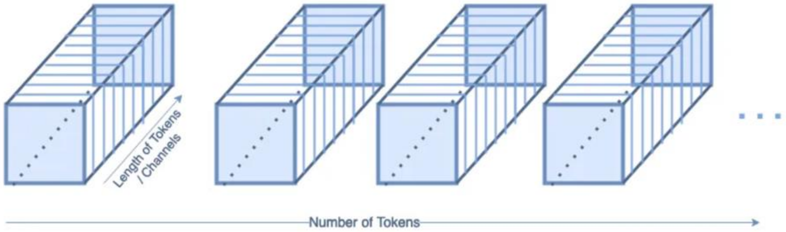
EQ-ViT Data Analysis

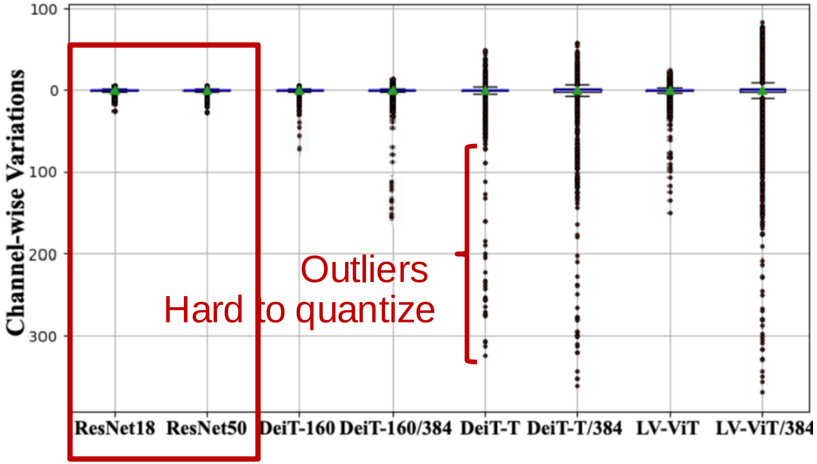
Long-Tail Distribution: Attention Matrix & Act After GELU

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers



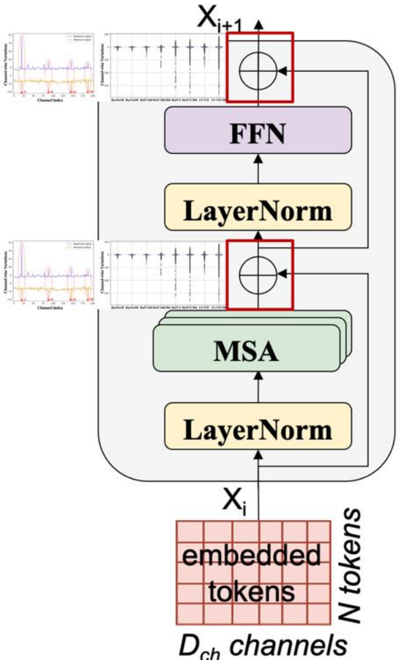


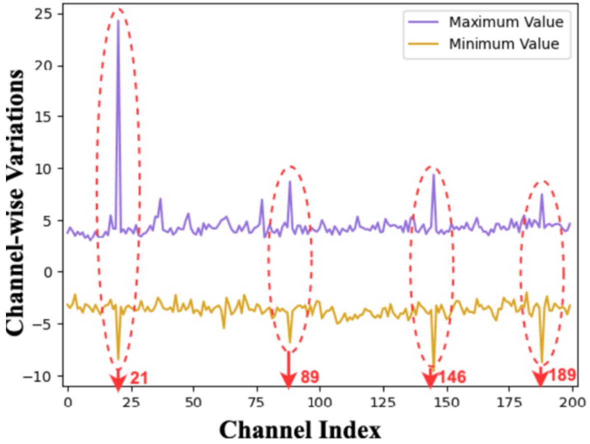


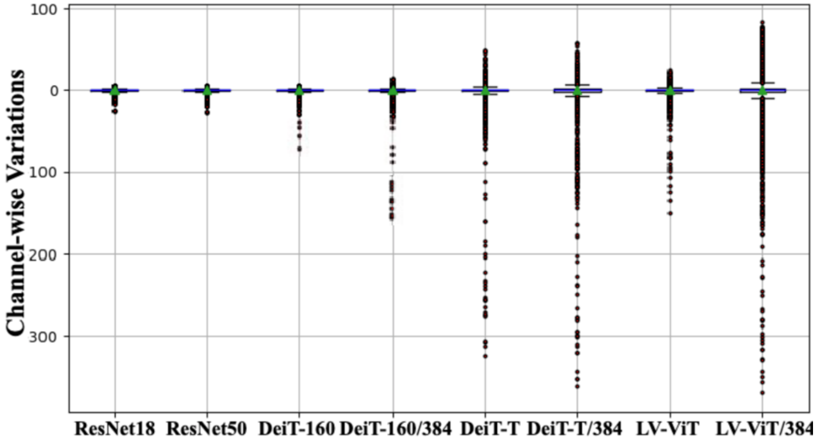


EQ-ViT Data Analysis

- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

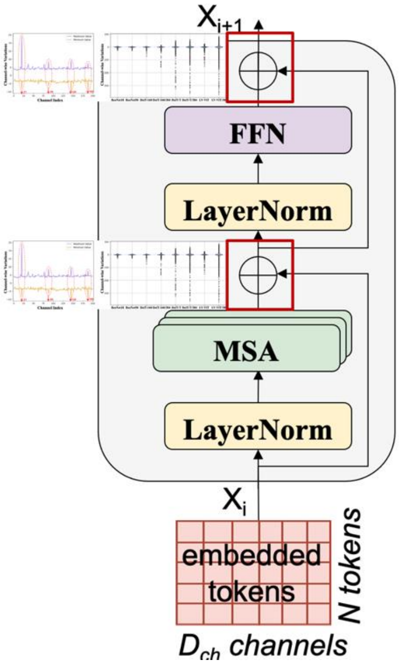


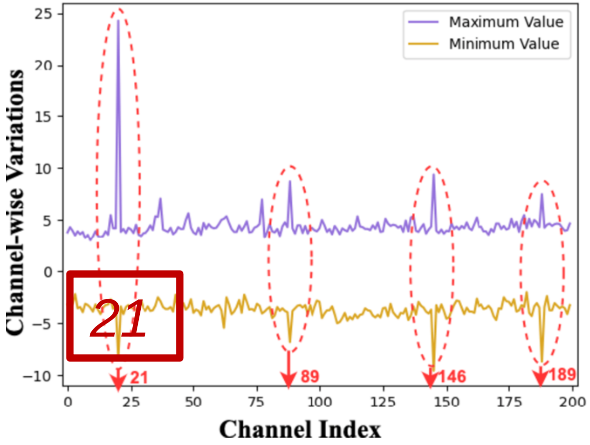


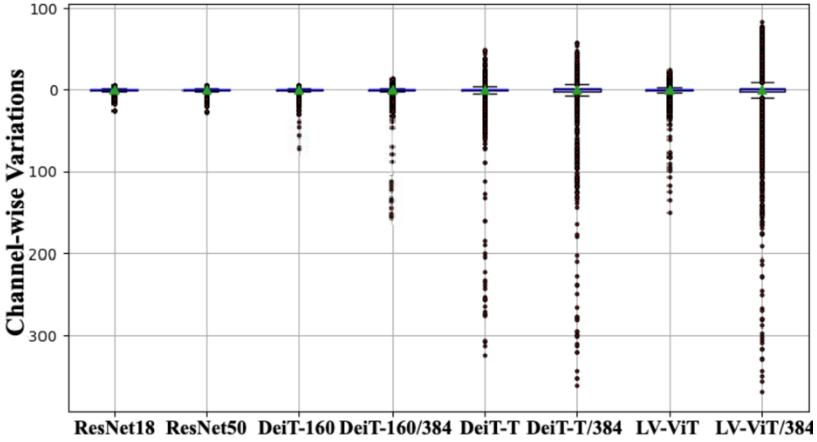


EQ-ViT Data Analysis

- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

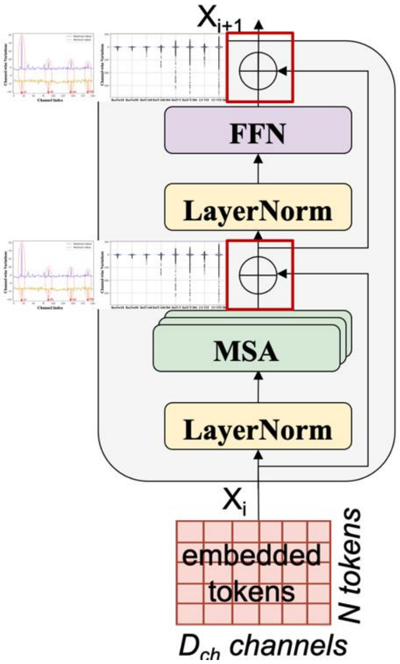


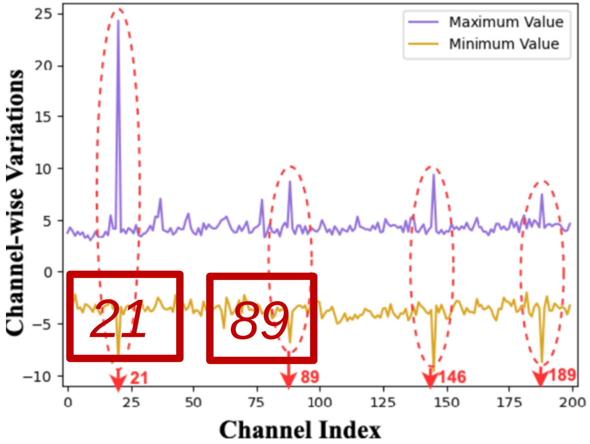


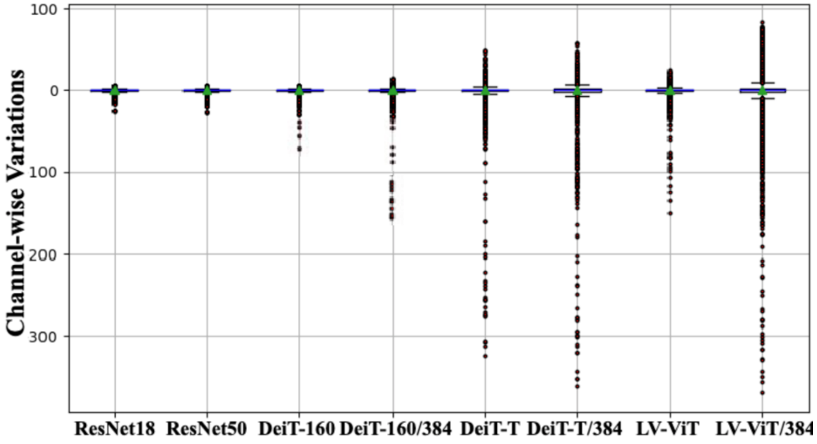


EQ-ViT Data Analysis

- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

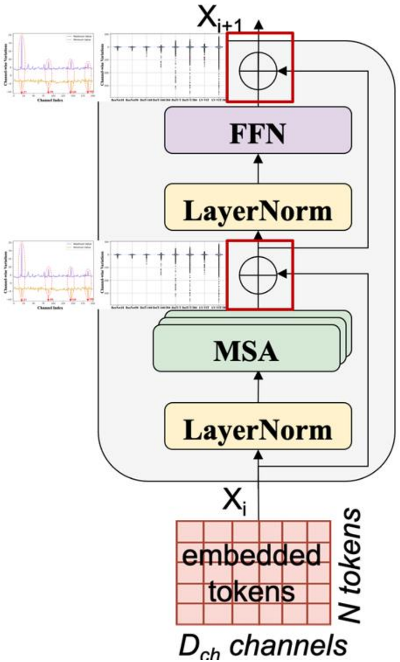


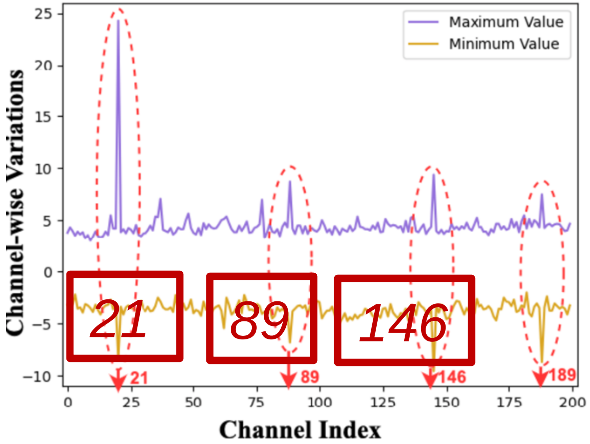


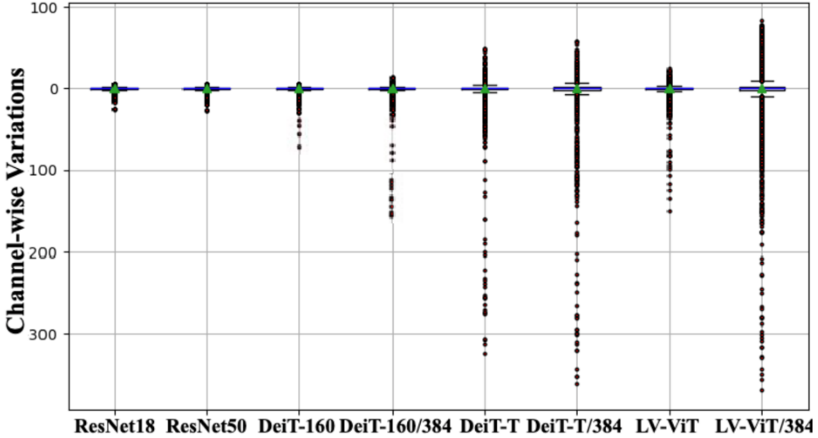


EQ-ViT Data Analysis

- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

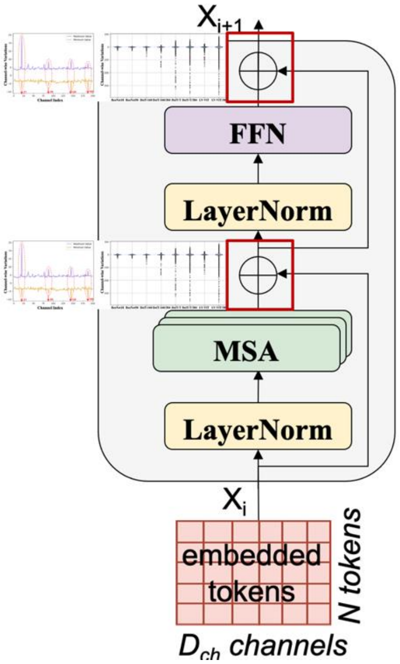


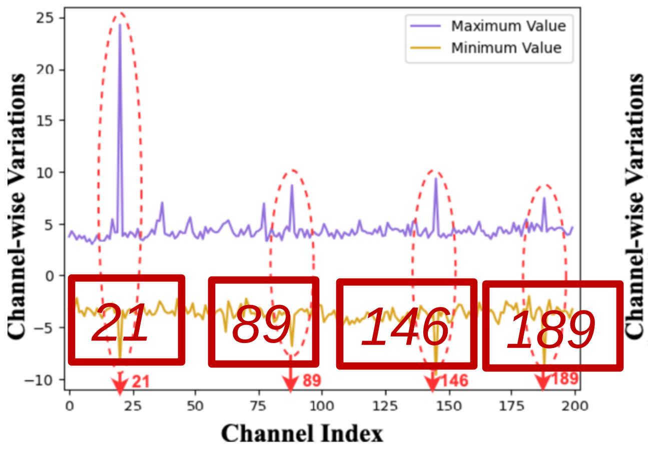


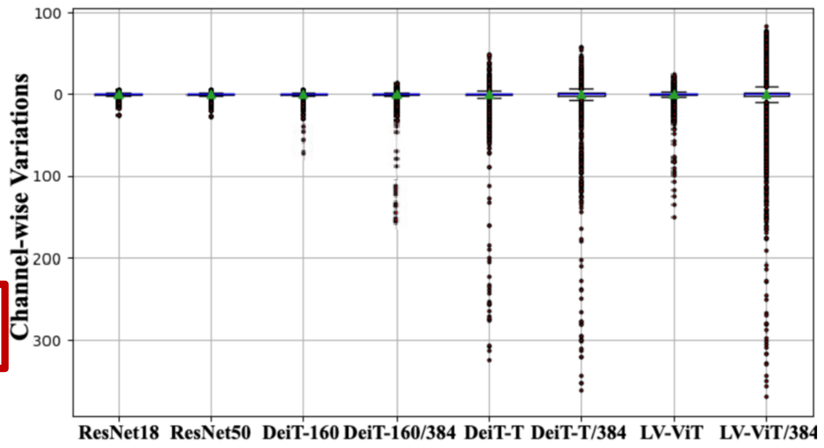


EQ-ViT Data Analysis

- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers





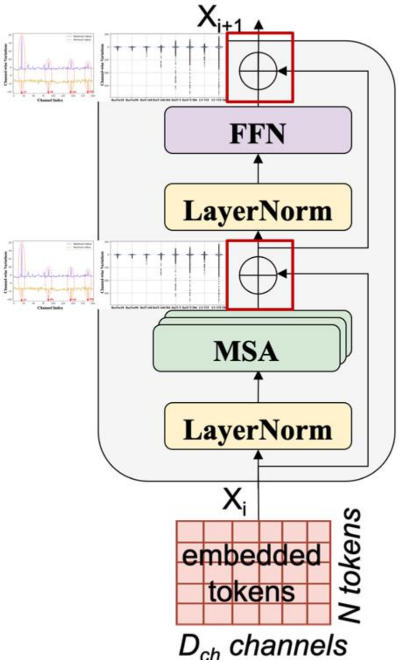


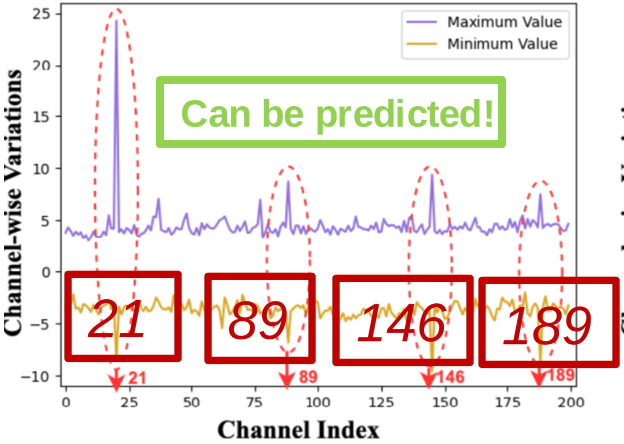
EQ-ViT Data Analysis

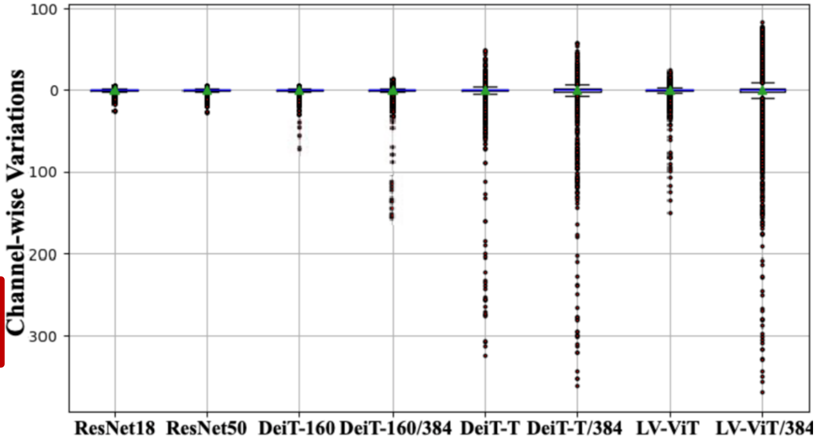
- Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers

Long-Tail Distribution: Attention Matrix & Act After GELU

Channel-wise Outlier: Fixed Layer & Fixed Channel & Fixed Data Range

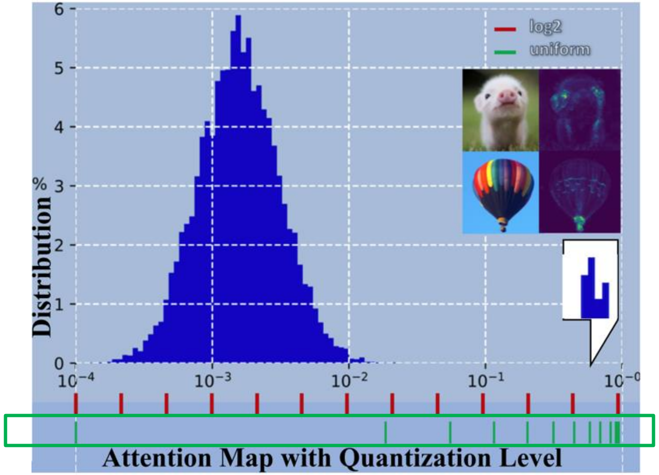






EQ-ViT Software Solution

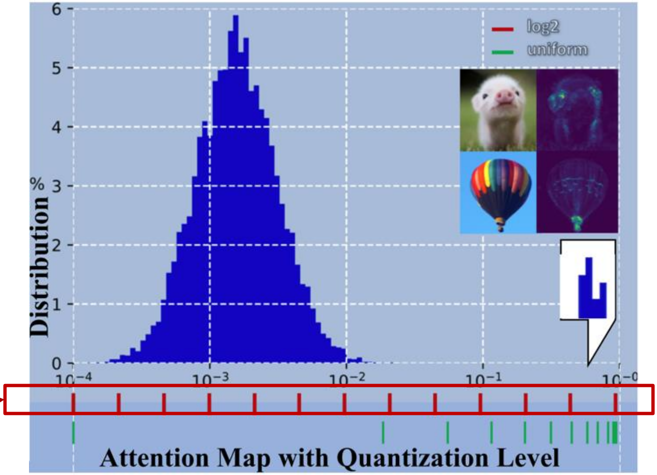
- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization



EQ-ViT Algorithm

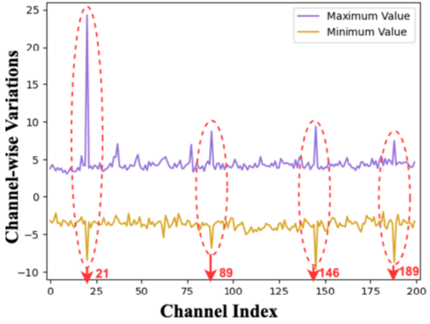
- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization

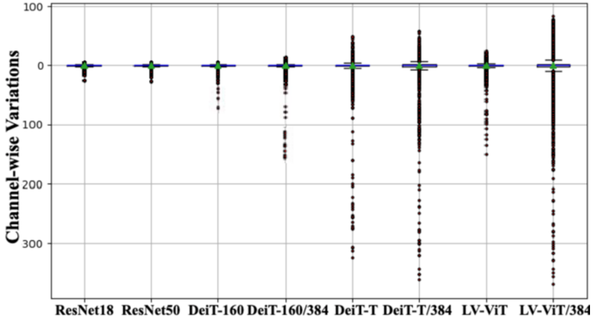
Log2 has 7~8 values to cover this large data range instead of only 1.



EQ-ViT Algorithm

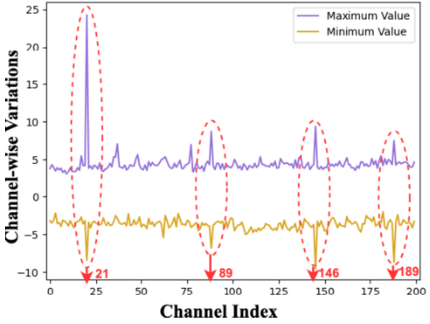
- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ 2^X Adaption

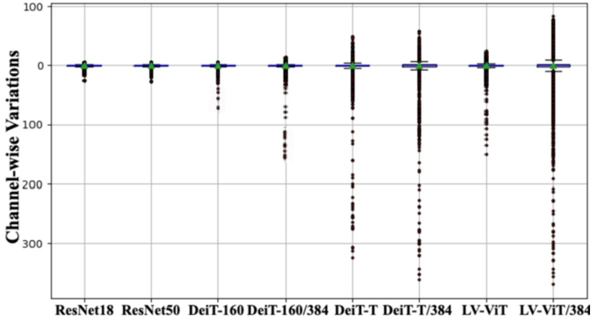




EQ-ViT Algorithm

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ $\mathbf{2^{\wedge}X}$ Adaption

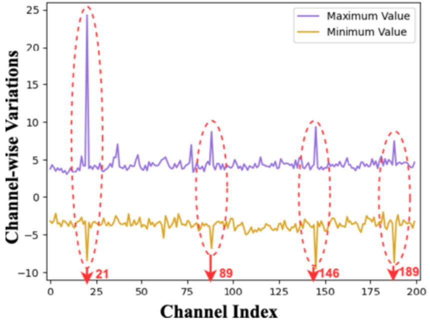


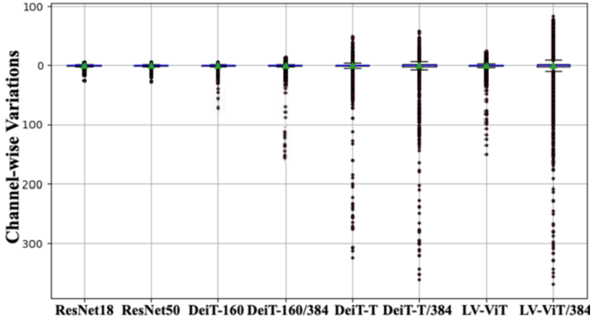


Layer-wise Uniform Quantization with 2^x

EQ-ViT Algorithm

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ 2^X Adaption





2^x Can be efficiently supported by Bitshift on FPGA board.

Layer-wise Uniform Quantization with 2^x

EQ-ViT Algorithm

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ 2^X Adaption
  - w/ Token Pruning Regularization

EQ-ViT Algorithm

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ $\mathbf{2^{\wedge}}\boldsymbol{X}$ Adaption
  - w/ Token Pruning Regularization

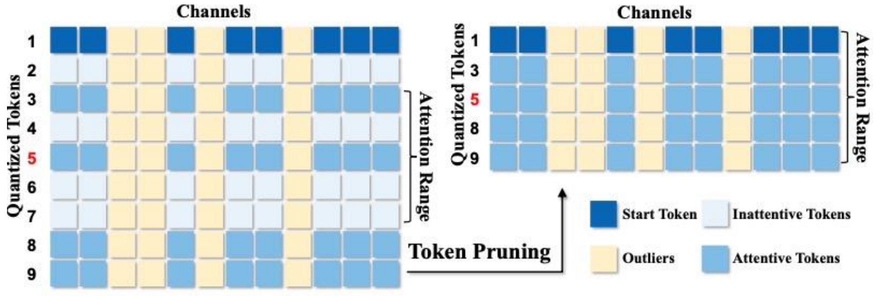


Figure 4: Activation Quantization With Token Pruning.

EQ-ViT Algorithm

- Two Specific Data Distribution inside ViTs
  - Long-Tail Distribution
  - Substantial Outliers
- Sub-8-bit: Activation-aware Full Quantization
  - Log2 Quantization
  - Outlier-aware Training w/ 2^X Adaption
  - w/ Token Pruning Regularization

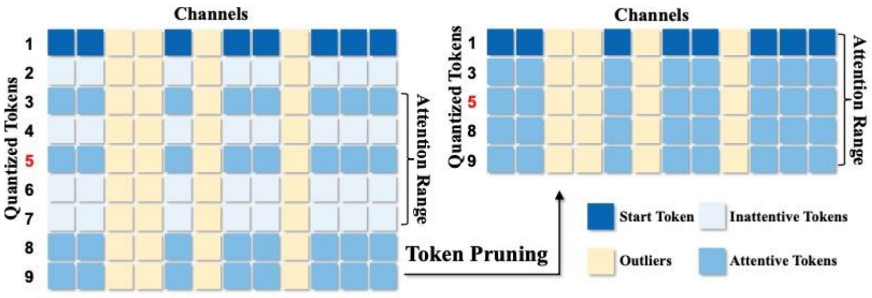
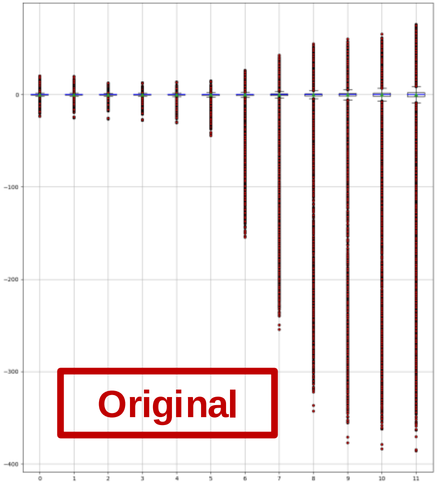
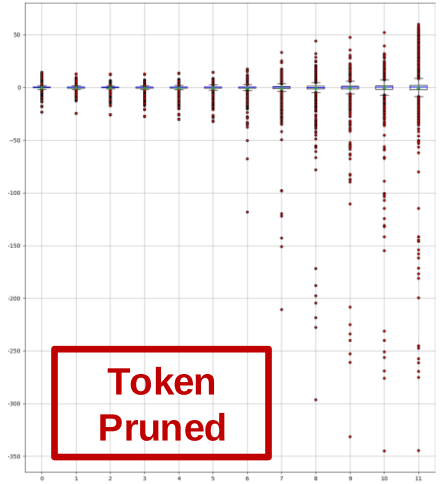


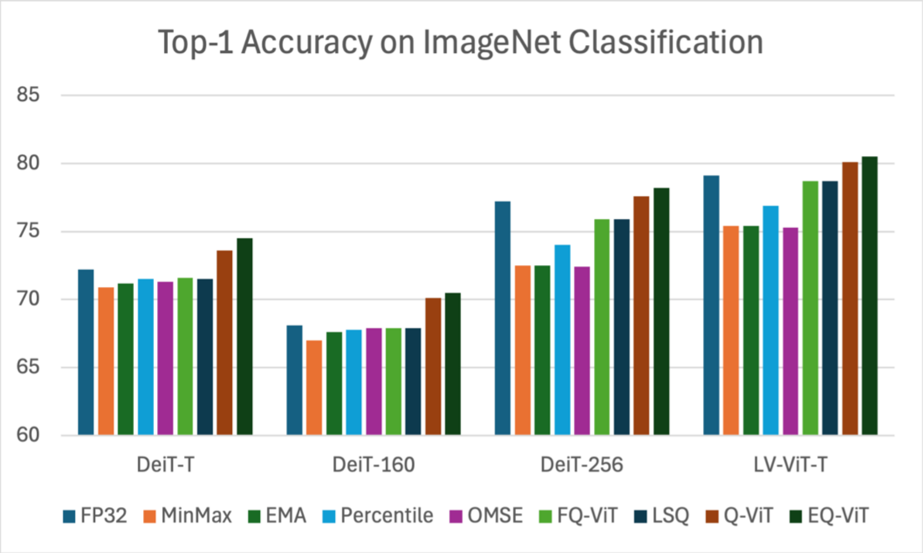
Figure 4: Activation Quantization With Token Pruning.

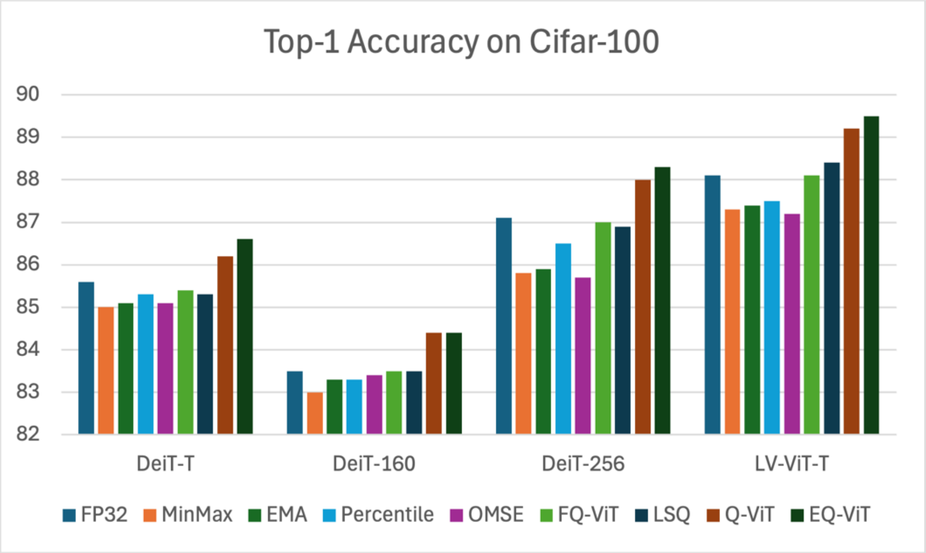




Experiment Results

- Application accuracy performance



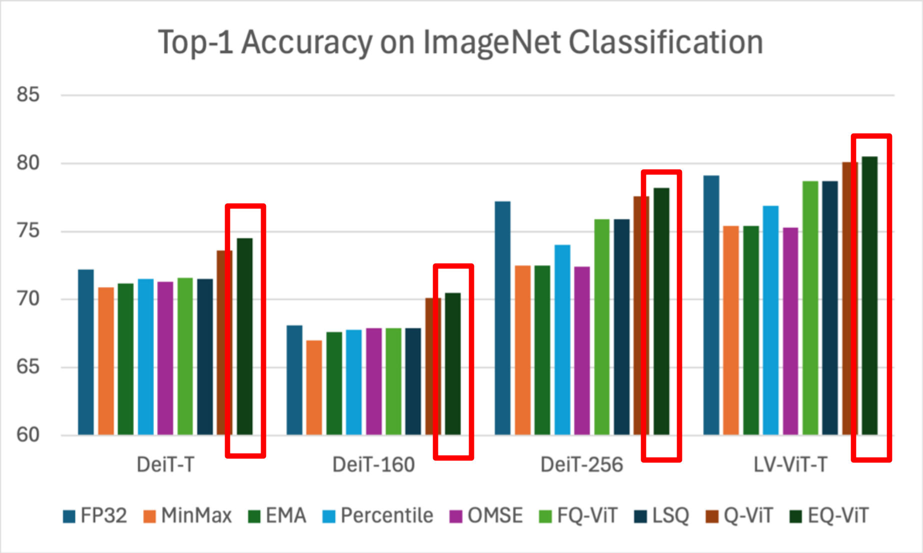


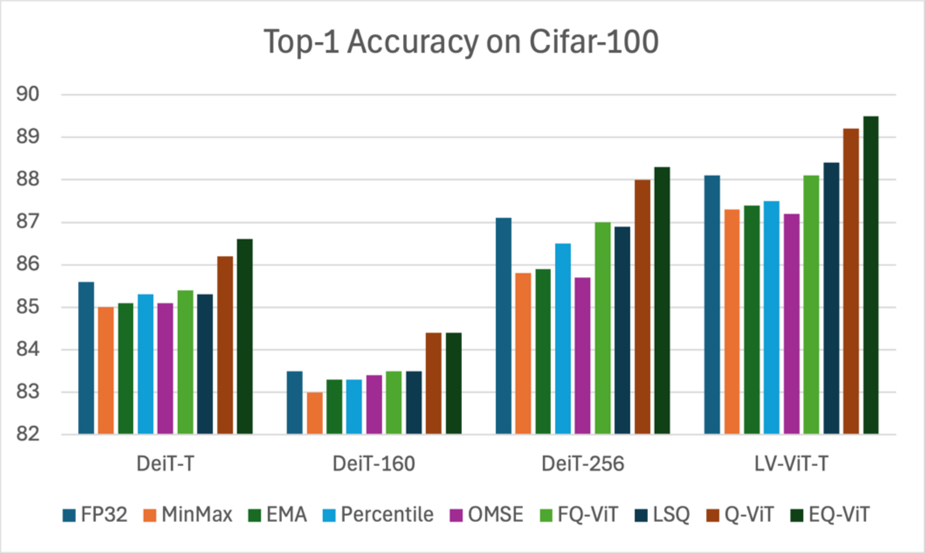
On ImageNet: EQ-ViT can enhance task accuracy up to 2.4% over the baseline, better up to 6.2% higher than other SOTA;

On Cifar-100: EQ-ViT can enhance task accuracy up to 1.4% over the baseline, better up to 1.8% higher than other SOTA.

Experiment Results

- Application accuracy performance



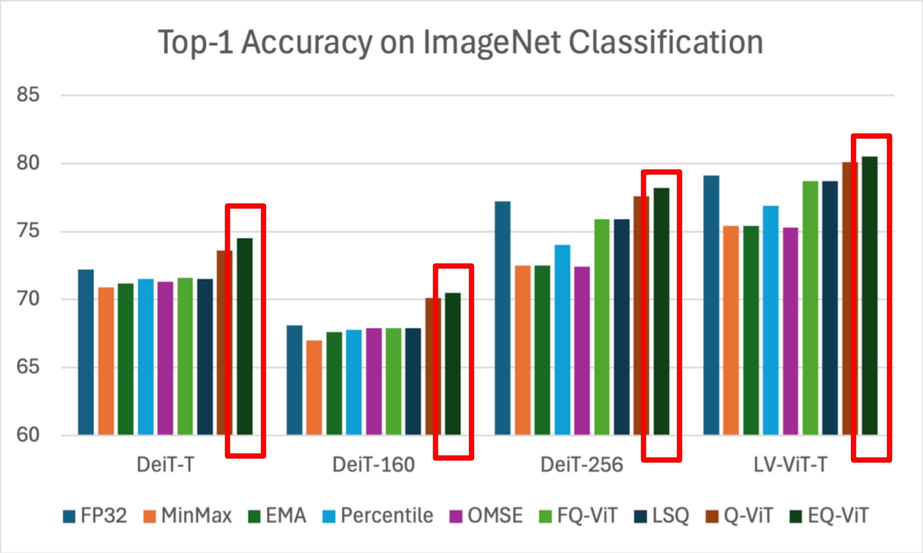


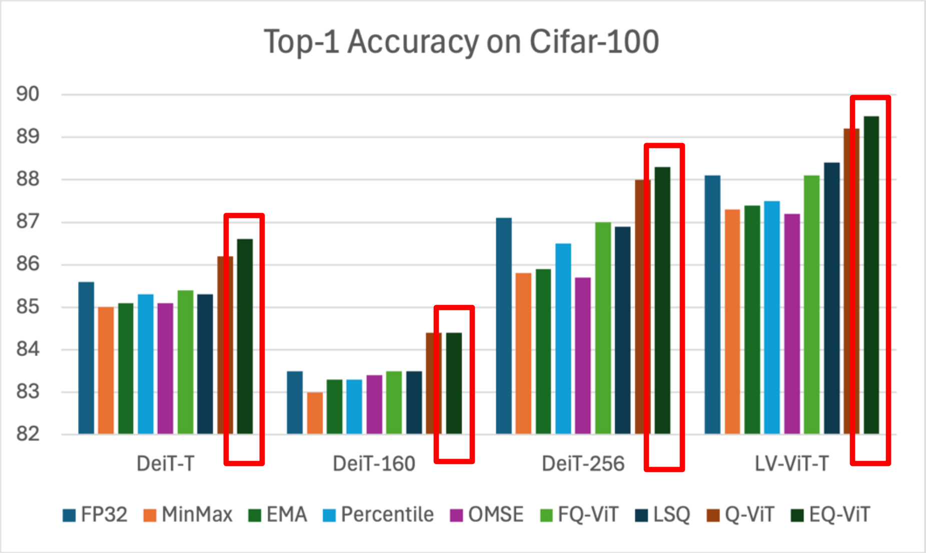
On ImageNet: EQ-ViT can enhance task accuracy up to 2.4% over the baseline, better up to 6.2% higher than other SOTA;

On Cifar-100: EQ-ViT can enhance task accuracy up to 1.4% over the baseline, better up to 1.8% higher than other SOTA.

Experiment Results

- Application accuracy performance





On ImageNet: EQ-ViT can enhance task accuracy up to 2.4% over the baseline, better up to 6.2% higher than other SOTA;

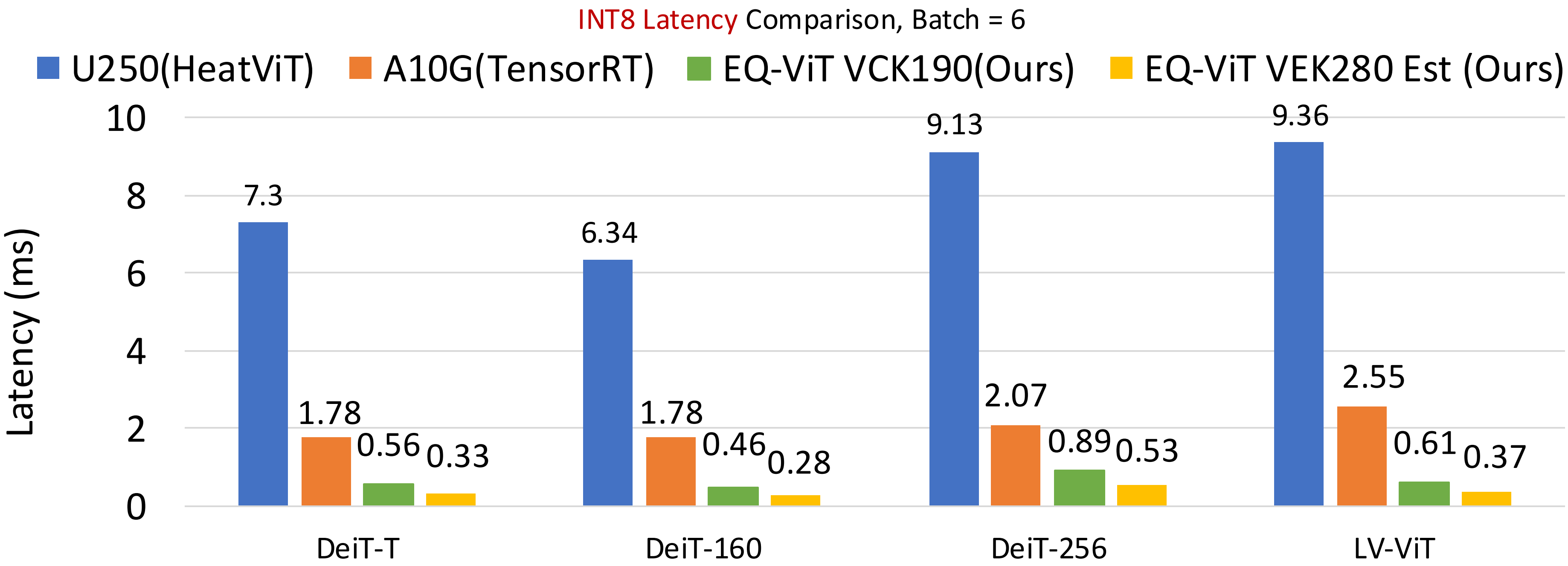
On Cifar-100: EQ-ViT can enhance task accuracy up to 1.4% over the baseline, better up to 1.8% higher than other SOTA.

Experiment Results

- Hardware performance comparisons across different solutions

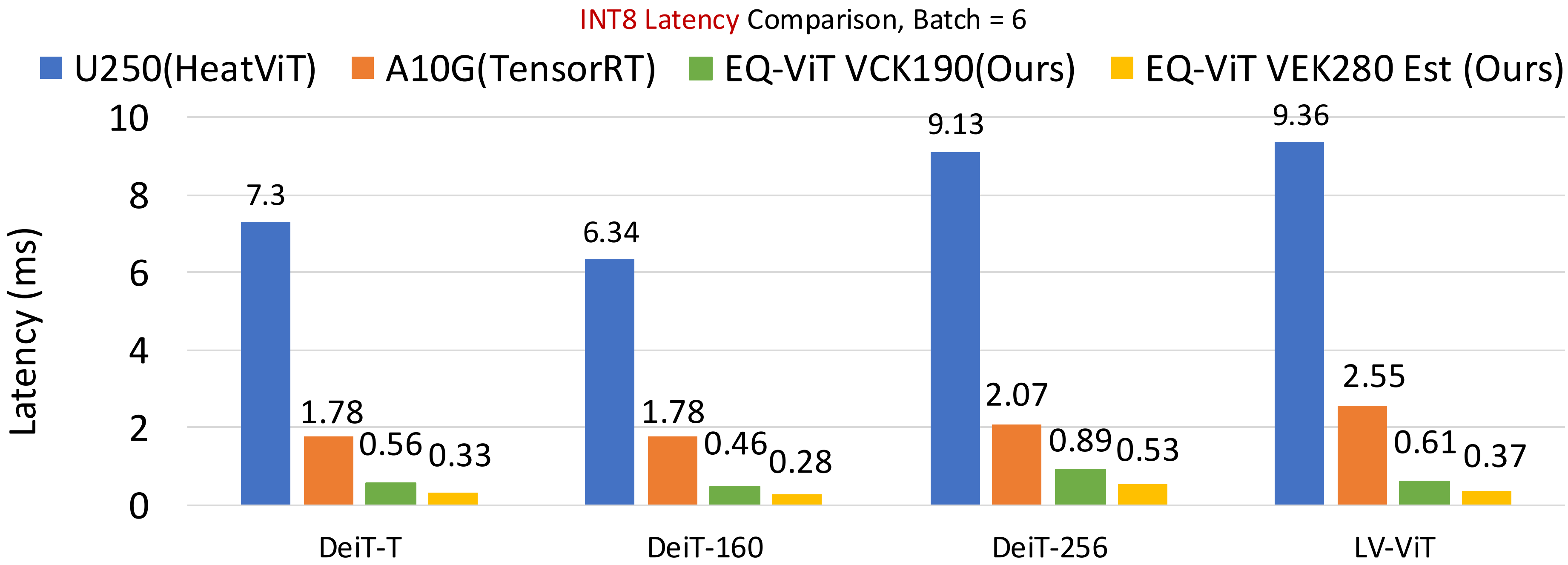
Experiment Results

- Hardware performance comparisons across different solutions



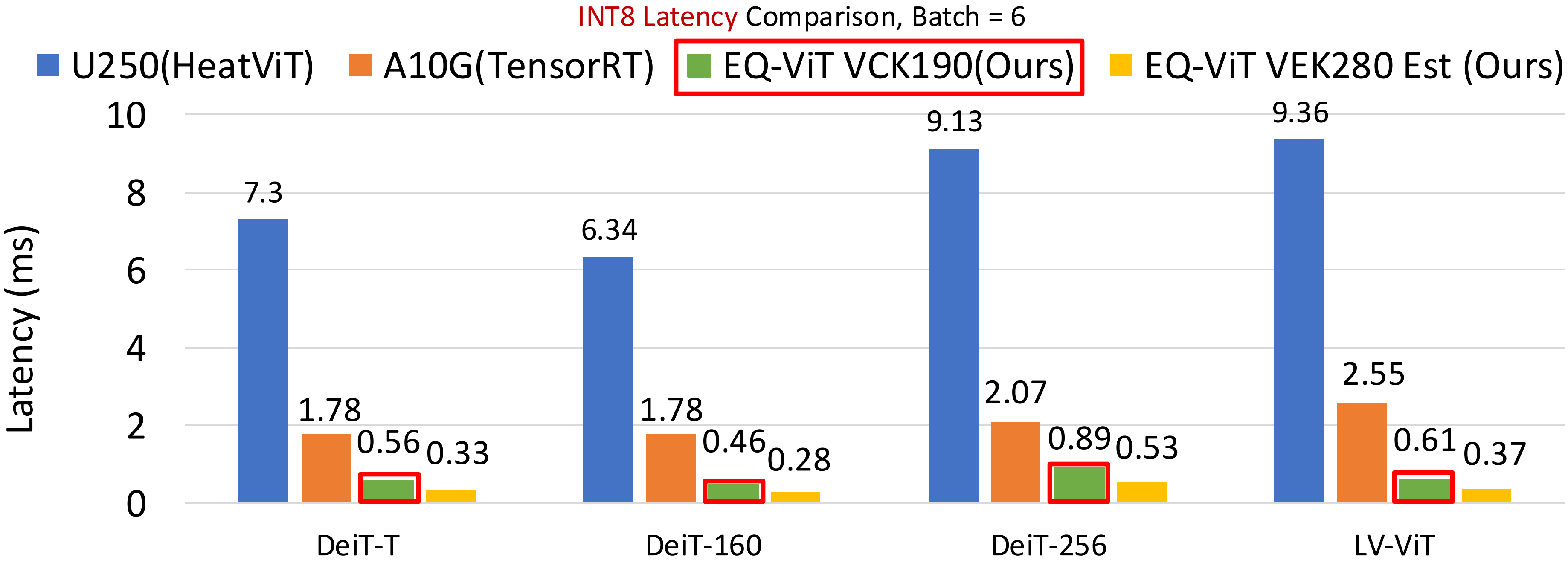
Experiment Results

- Hardware performance comparisons across different solutions
  - EQ-ViT on VCK190 achieves 13.1x and 3.4x average latency reduction compared with U250, A10G



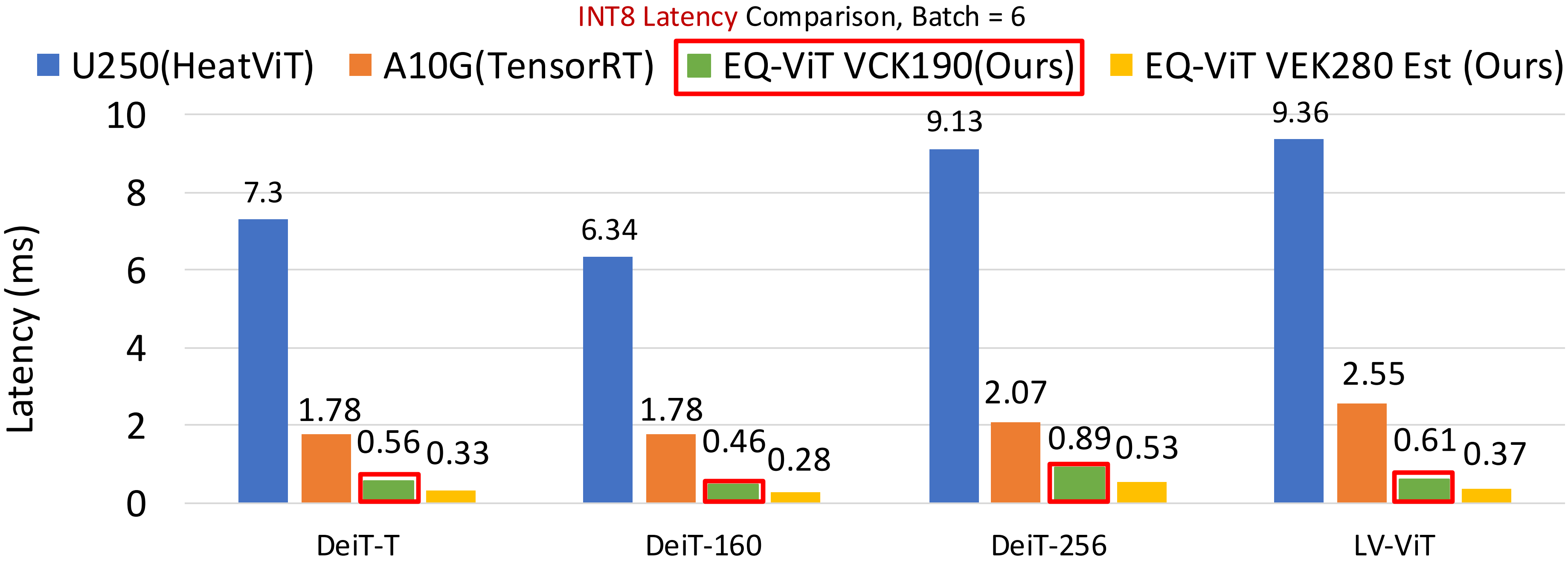
Experiment Results

- Hardware performance comparisons across different solutions
  - EQ-ViT on VCK190 achieves 13.1x and 3.4x average latency reduction compared with U250, A10G



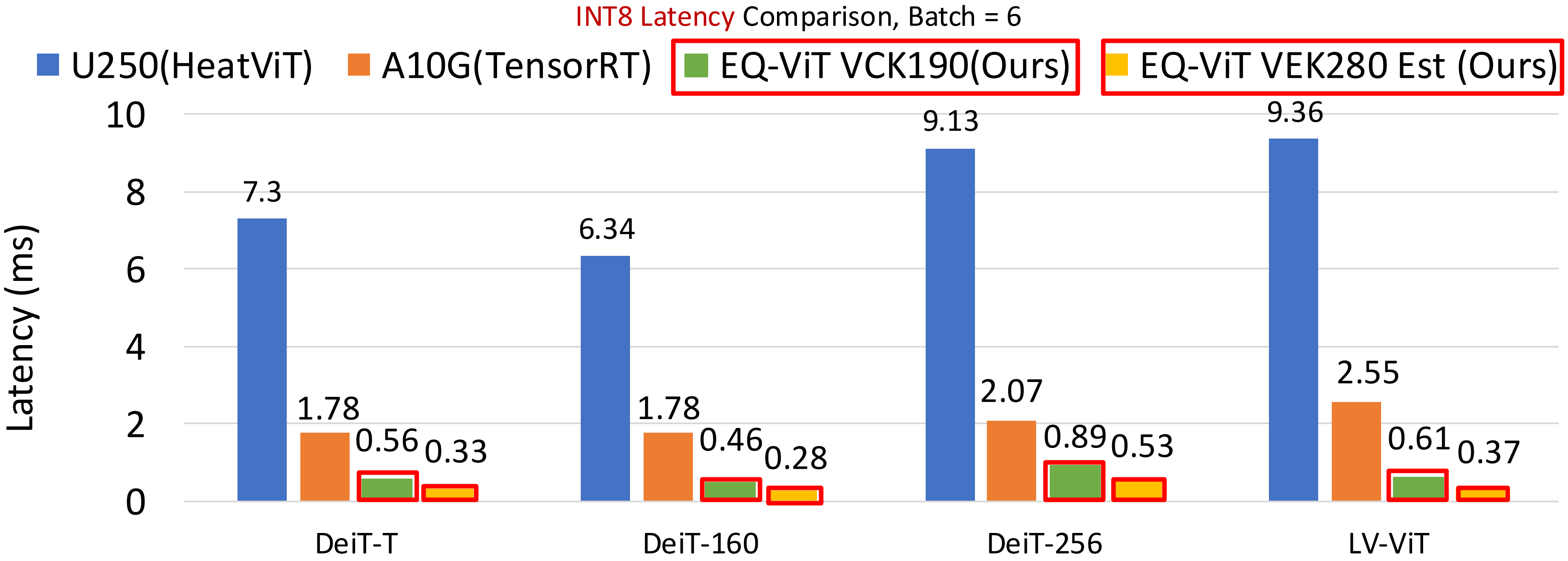
Experiment Results

- Hardware performance comparisons across different solutions
  - EQ-ViT on VCK190 achieves 13.1x and 3.4x average latency reduction compared with U250, A10G
  - Estimation of EQ-ViT on VEK280 shows an another 1.7x average latency reduction over VCK190



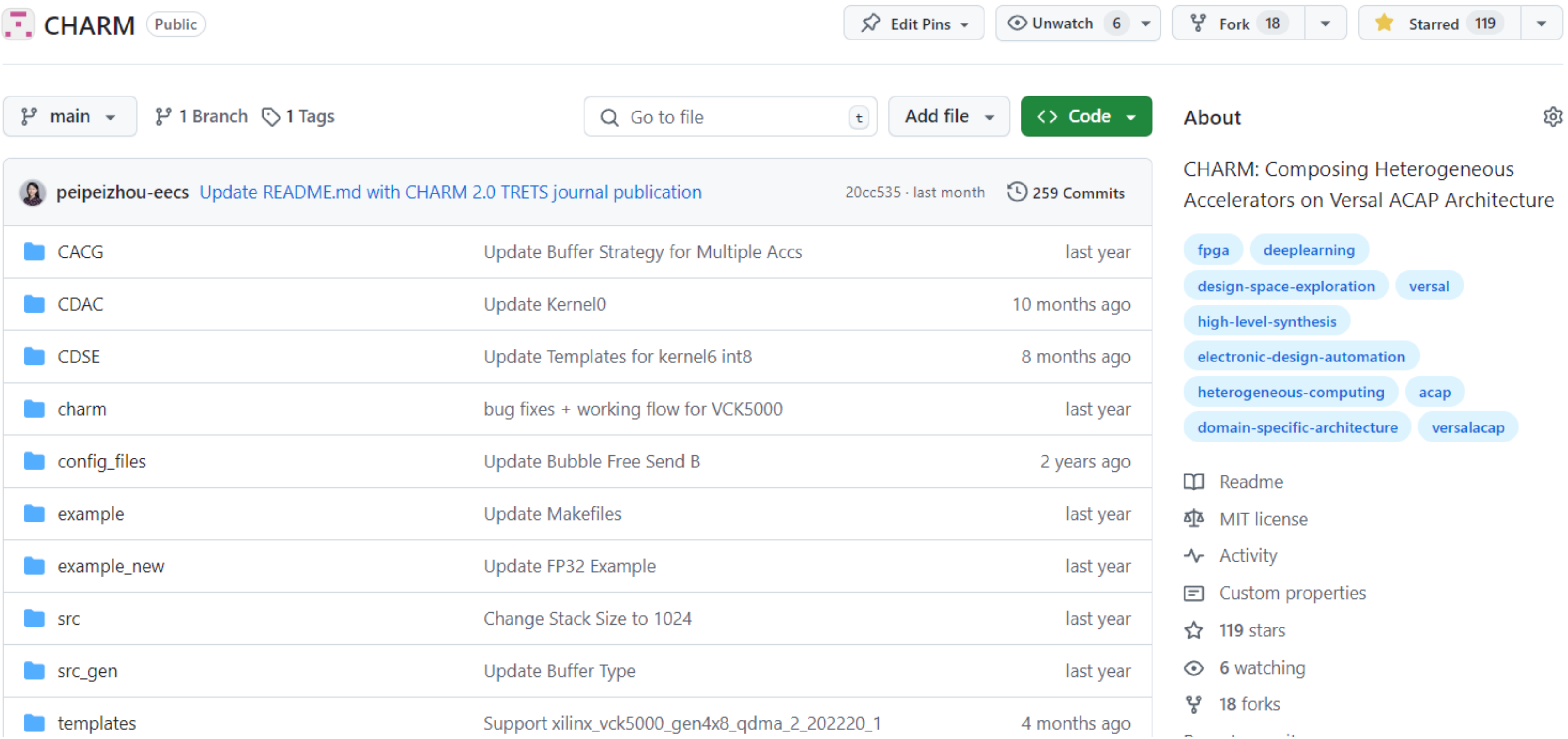
Experiment Results

- Hardware performance comparisons across different solutions
  - EQ-ViT on VCK190 achieves 13.1x and 3.4x average latency reduction compared with U250, A10G
  - Estimation of EQ-ViT on VEK280 shows an another 1.7x average latency reduction over VCK190



Open-Source Tool

- GitHub Link: https://github.com/arc-research-lab/CHARM



Thank You & Welcome to Questions

EQ-ViT: Algorithm-Hardware Co-Design for End-to-End Acceleration of Real-Time Vision Transformer Inference on Versal ACAP Architecture

P. Dong*, J. Zhuang* , Z. Yang , S. Ji , Y. Li, D. Xu, H. Huang, J. Hu,
A.K. Jones, Y. Shi, Y. Wang, P. Zhou

*Co-first authors

Massachusetts Institute of Technology; Brown University; Northeastern University;
North Carolina State University; Syracuse University;
University of Maryland, College Park; University of Pittsburgh; University of Notre Dame

National Science Foundation