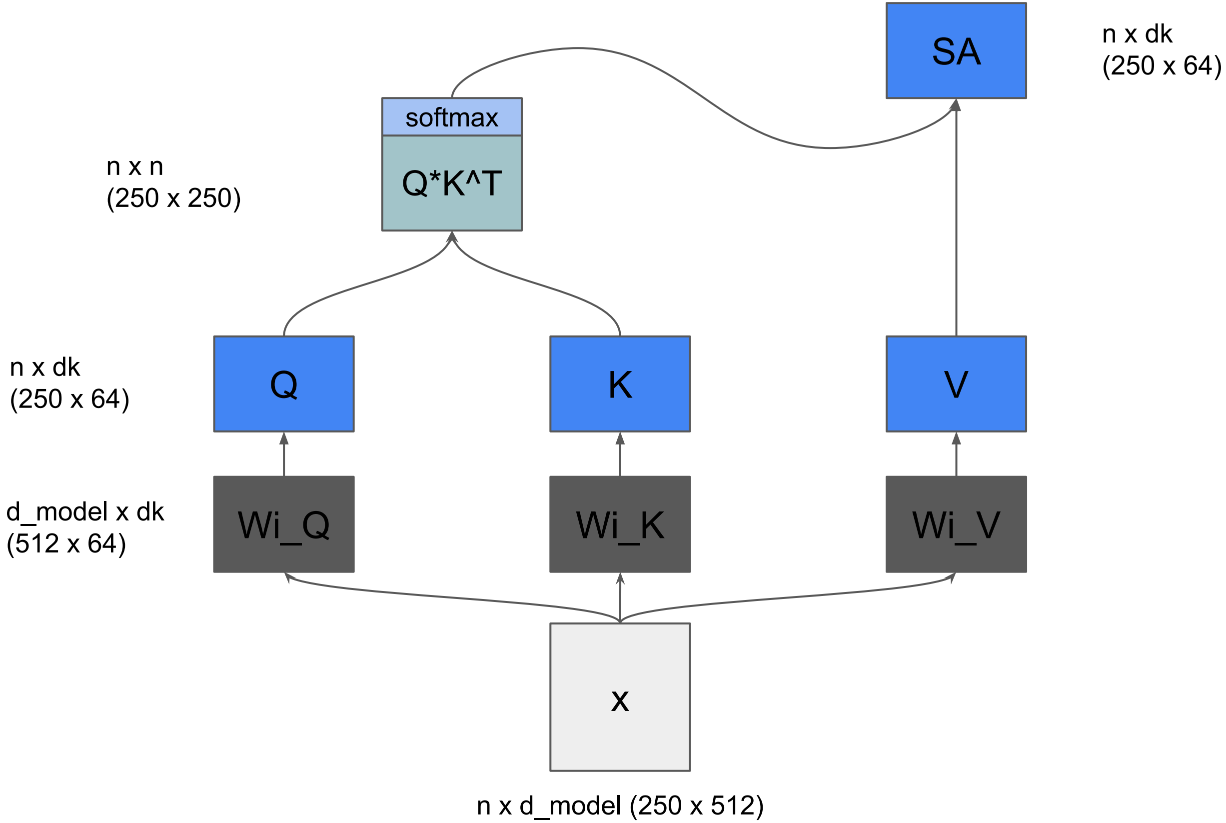
Week 3

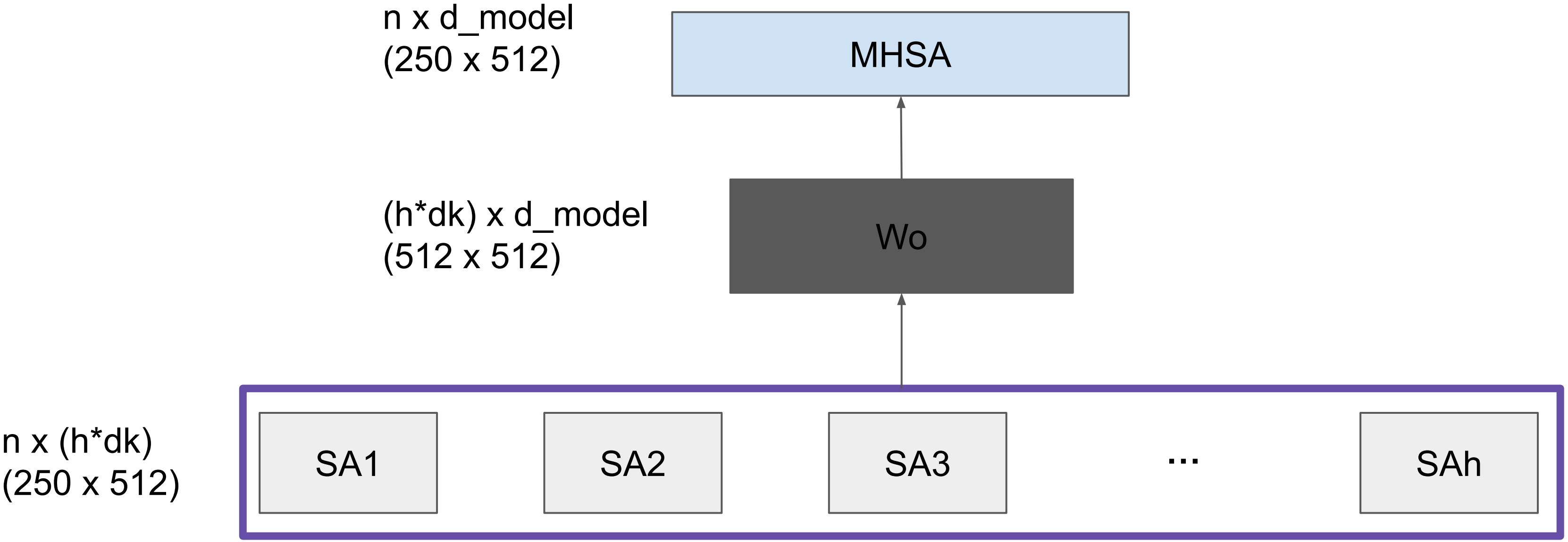
SA (n: sequence length, dk: Q,K,V channels, d_model: embeddings)



SA

- Input
  - Wi (Q, K, V) = d_model x dk = 512 x 64 (dv = dk)
  - X = n x d_model = 25000 x 512 (n: batch size)
- Output
  - Q, K, V = n x dk = 25000 x 64
  - Q * K^T = n x n = 25000 x 25000
  - sa(Q, K, V) = **n x dk = 25000 x 64**

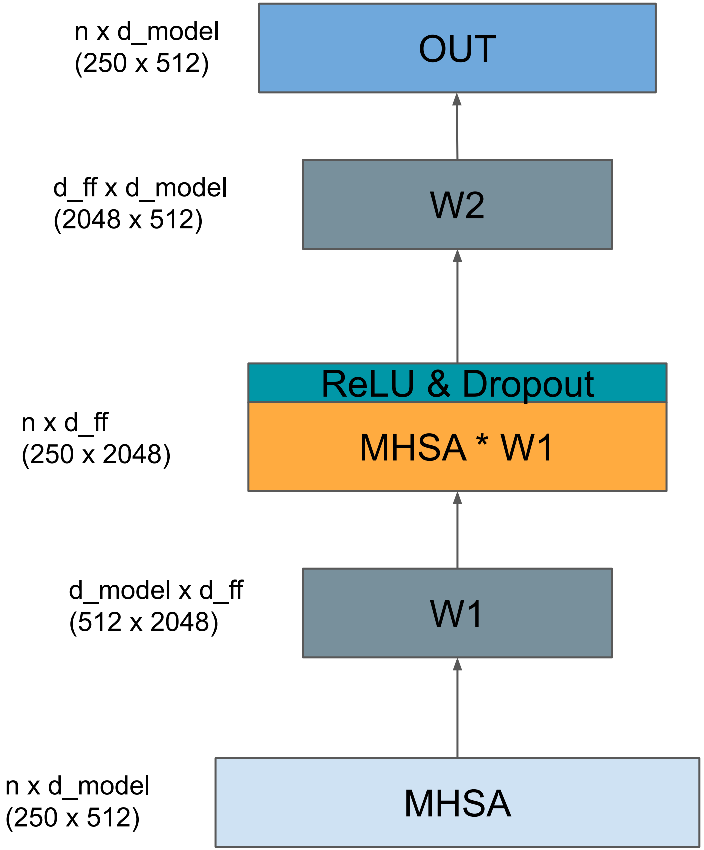
MHSA


n x d_model
(250 x 512)
MHSA
(h*dk) x d_model
(512 x 512)
Wo
n x (h*dk)
(250 x 512)
SA1
SA2
SA3
...
SAh

MHSA

- Input
  - SA_i = n x dk = 25000 x 64, h = 8
  - Wo = h*dk x d_model = 512 x 512
- Output
  - [SA_1 SA_2 … SA_h] * Wo = (n x h*dk) x (h*dk x d_model) = **n x d_model = 25000 x 512**

FFN


n x d_model
(250 x 512)
OUT
d_ff x d_model
(2048 x 512)
W2
ReLU & Dropout
n x d_ff
(250 x 2048)
MHSA * W1
d_model x d_ff
(512 x 2048)
W1
n x d_model
(250 x 512)
MHSA

FFN

- Input
  - W_1 = d_model x d_ff = 512 x 2048, W_2 = d_ff x d_model = 2048 x 512
  - X (from MHSA) = n x d_model = 25000 x 512
- Output
  - X * W_1 = (n x d_model) x (d_model x d_ff) = n x d_ff = 25000 x 2048
  - _ * W_2 = (n x d_ff) x (d_ff x d_model) = **n x d_model = 25000 x 512**

Apply Nvidia’s softmax to Ara

mi: local maximum, xi: current element, di: sum

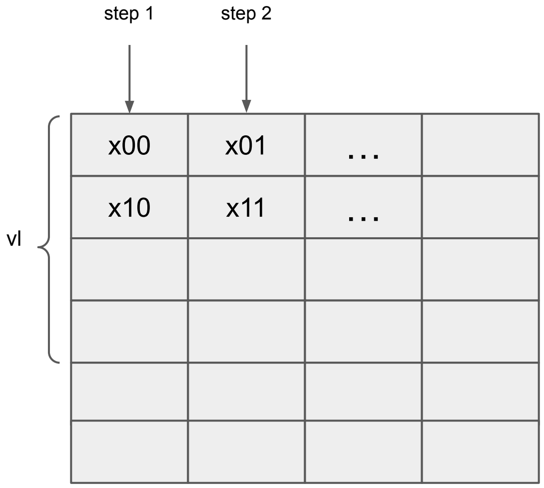
- Loop 1:
  - mi (max)
  - di (sub, shift, add)
- Loop 2:
  - reciprocal (1/dv)
  - sub, shift, mult

Algorithm

Implemented in PPU

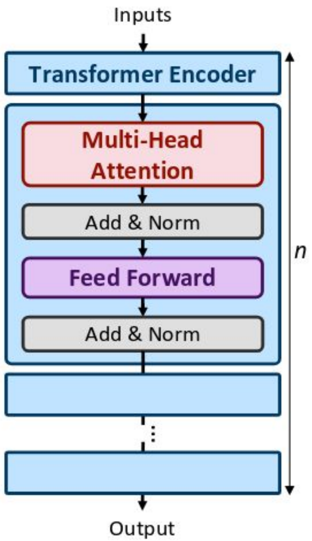
for i $\leftarrow 1, V$ **do**
$m_i \leftarrow \mathrm{IntMax}(m_{i-1}, x_i)$
$y_i \leftarrow 2^{x_i - m_i}$
$d_i \leftarrow d_{i-1} \gg (m_i - m_{i-1})$
$d_i \leftarrow d_i + y_i$
end for

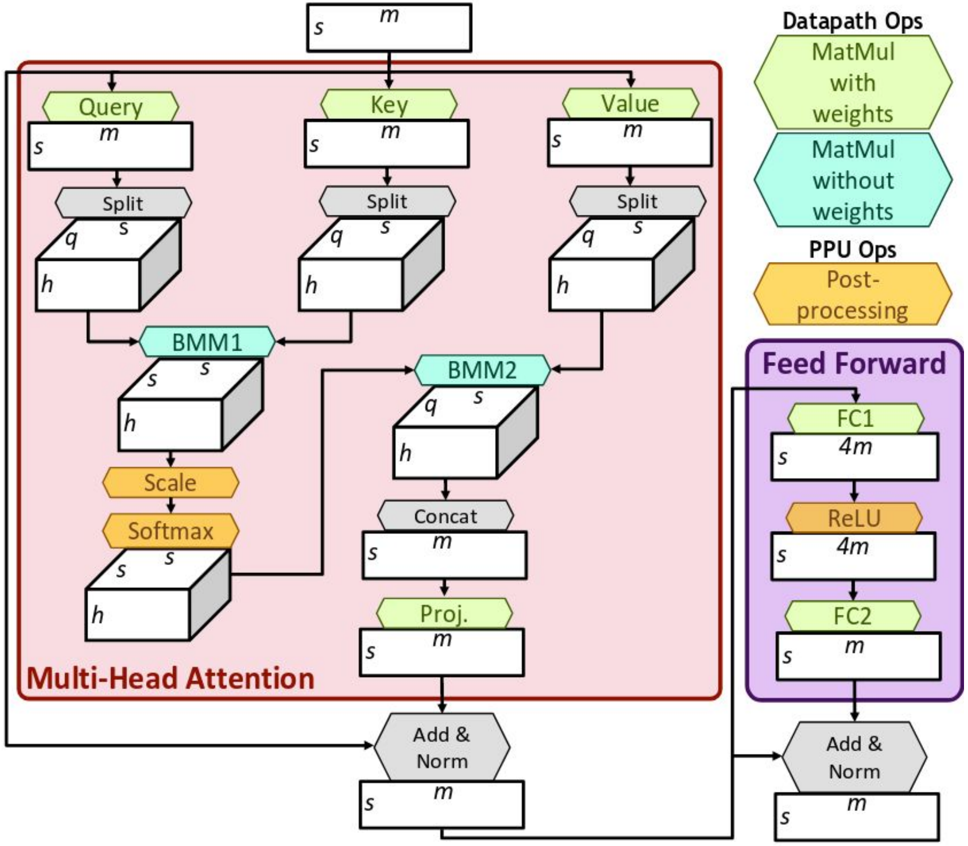
for i $\leftarrow 1, V$ **do**
$y_i \leftarrow \frac{y_i \gg (m_v - m_i)}{d_v}$
end for



M x N

A 17–95.6 TOPS/W Deep Learning Inference Accelerator with Per-Vector Scaled 4-bit Quantization for Transformers in 5nm, Ben Keller Rangharajan Venkatesan Steve Dai Stephen Tell Brian Zimmer William Dally Tom Gray Brucek Khailany





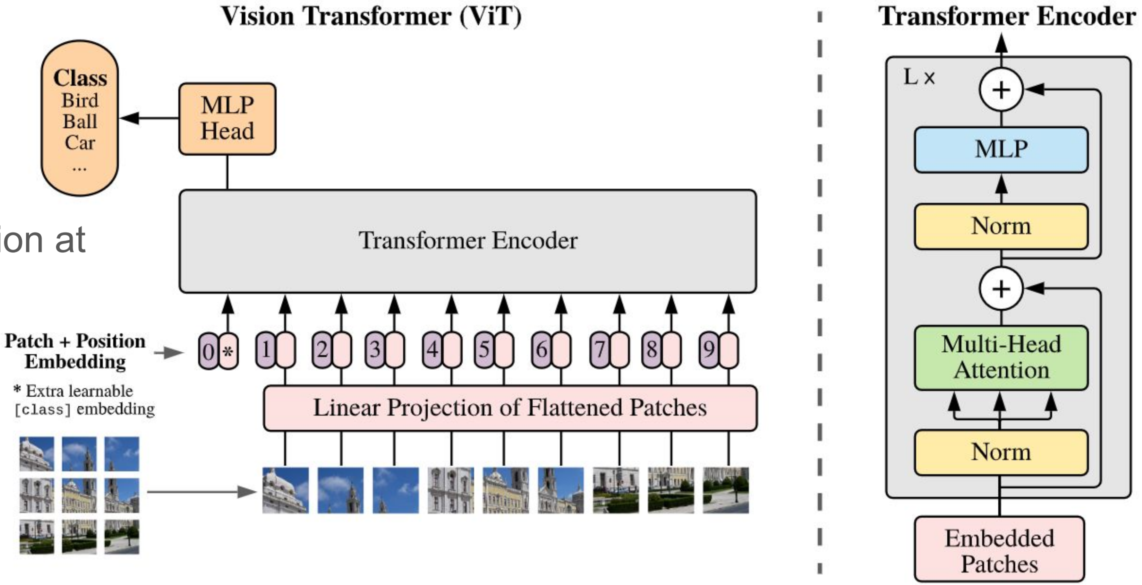
	Bert-Large	Bert-Base
Model (m)	1024	768
Heads (h)	16	12
Sequence* (s)	384	384
Query size (q)	64	64
Layers (n)	24	12
MACs ($\times 10^9$)	123	35

*Sequence varies with workload

A 17–95.6 TOPS/W Deep Learning Inference Accelerator with Per-Vector Scaled 4-bit Quantization for Transformers in 5nm, Ben Keller Rangharajan Venkatesan Steve Dai Stephen Tell Brian Zimmer William Dally Tom Gray Brucek Khailany

Transformer in CV (ViT)

- 2D images to 1D Input
  - Image H x W x C
  - Patches N x (P^2 * C)
  - Prepend class token Xclass, transformed as class prediction at the output (not necessary)
  - Positional encoding, similar accuracy in 1D and 2D



$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C)\times D},\, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$$

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby

Transformer in CV (ViT)

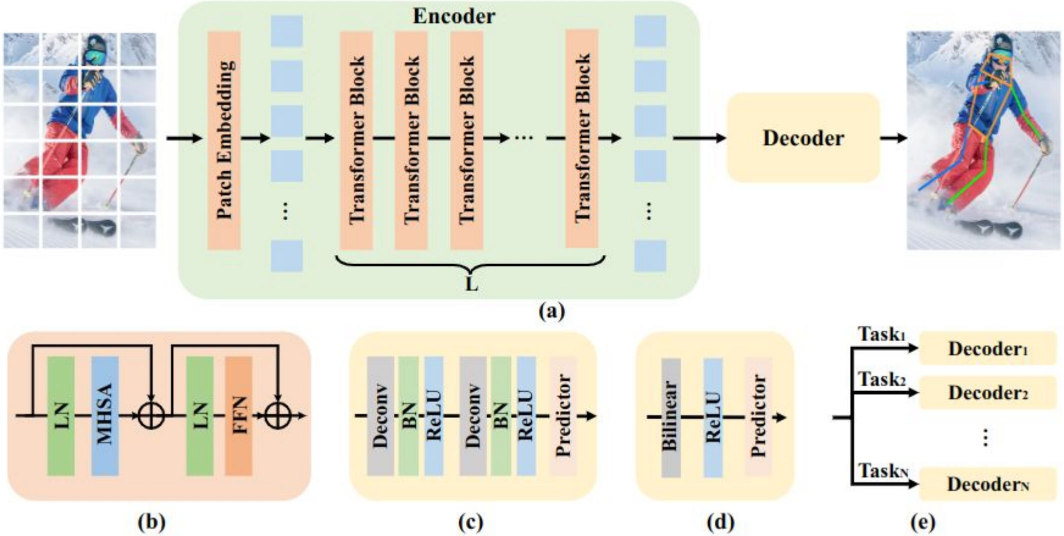
- Transformer pre-trained on JFT-300M outperforms ResNet, and requires less expenses
- Better to extract long-range information
- Hybrid Architecture
  - Input could be feature map from CNN

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Pose Estimation (ViTPose)

- Plain ViT (no CNN)
- MIM pretraining as backbones
- Simple structure:
  - No cross-attention in decoder
  - Deconvolution or bilinear interpolation
- Window-based attention
  - Relative position embedding
  - Shift-window (broadcast info between windows)
  - Pooling window
  - *Swin-Transformer*



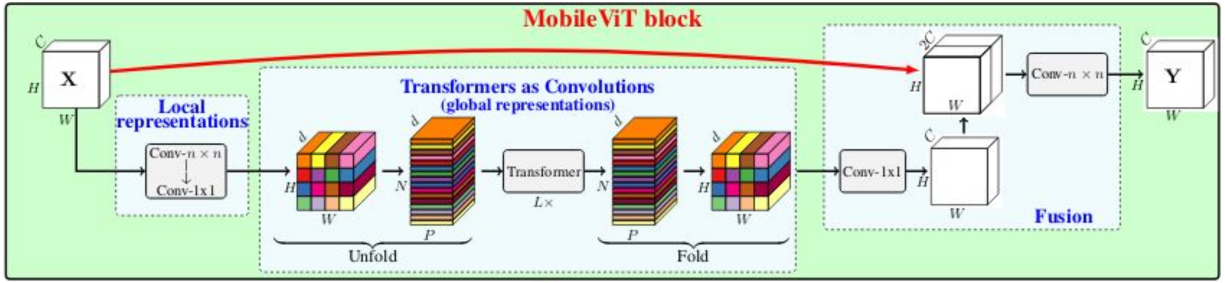
ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation Yufei Xu1, Jing Zhang1, Qiming Zhang1, Dacheng Tao2,1 1School of Computer Science, The University of Sydney, Australia 2JD Explore Academy, China

ViTPose

Model	Backbone	Params (M)	Speed (fps)	Input Resolution	Feature Resolution	COCO val AP	COCO val AR	COCO test-dev AP	COCO test-dev AR
SimpleBaseline [44]	ResNet-152	60	829	256x192	1/32	73.5	79.0	-	-
HRNet [38]	HRNet-W32	29	916	256x192	1/4	74.4	78.9	-	-
HRNet [38]	HRNet-W32	29	428	384x288	1/4	75.8	81.0	74.9	80.1
HRNet [38]	HRNet-W48	64	649	256x192	1/4	75.1	80.4	-	-
HRNet [38]	HRNet-W48	64	309	384x288	1/4	76.3	81.2	75.5	80.5
UDP [19]	HRNet-W48	64	309	384x288	1/4	77.2	82.0	-	-
TokenPose-L/D24 [28]	HRNet-W48	28	602	256x192	1/4	75.8	80.9	75.1	80.2
TransPose-H/A6 [46]	HRNet-W48	18	309	256x192	1/4	75.8	80.8	75.0	-
HRFormer-B [48]	HRFormer-B	43	158	256x192	1/4	75.6	80.8	-	-
HRFormer-B [48]	HRFormer-B	43	78	384x288	1/4	77.2	82.0	76.2	81.2
ViTPose-B	ViT-B	86	944	256x192	1/16	75.8	81.1	75.1	80.3
ViTPose-B*	ViT-B	86	944	256x192	1/16	77.5	82.6	76.4	81.5
ViTPose-L	ViT-L	307	411	256x192	1/16	78.3	83.5	77.3	82.4
ViTPose-L*	ViT-L	307	411	256x192	1/16	79.1	84.1	77.8	82.8
ViTPose-H	ViT-H	632	241	256x192	1/16	79.1	84.1	78.1	83.1
ViTPose-H*	ViT-H	632	241	256x192	1/16	79.8	84.8	78.4	83.4

Light-Weight ViT (MobileViT)

- CNNs: local feature extraction and less sensitivity to data augmentation
- Transformers: input-adaptive weighting and global processing



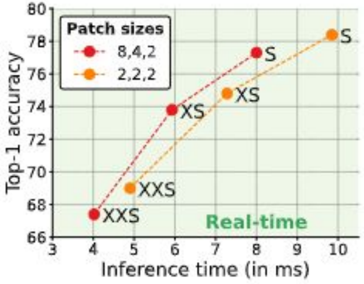
- Standard CNN: 1. Unfolding 2. Local Processing 3. Folding
- This work: replace local processing with transformer

MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer, Sachin Mehta, Mohammad Rastegari

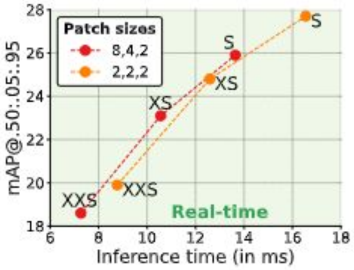
Light-Weight ViT (MobileViT)

Model	# Params. ⇓	Top-1 ⇑
MobileNetv1	2.6 M	68.4
MobileNetv2	2.6 M	69.8
MobileNetv3	2.5 M	67.4
ShuffleNetv2	2.3 M	69.4
ESPNetv2	2.3 M	69.2
MobileViT-XS (Ours)	2.3 M	**74.8**

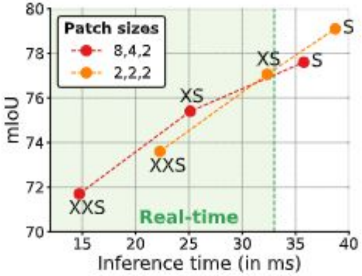
Model	# Params. ⇓	Top-1 ⇑
DenseNet-169	14 M	76.2
EfficientNet-B0	5.3 M	76.3
ResNet-101	44.5 M	77.4
ResNet-101-SE	49.3 M	77.6
MobileViT-S (Ours)	5.6 M	**78.4**



(a) Classification @ 256 × 256



(b) Detection @ 320 × 320

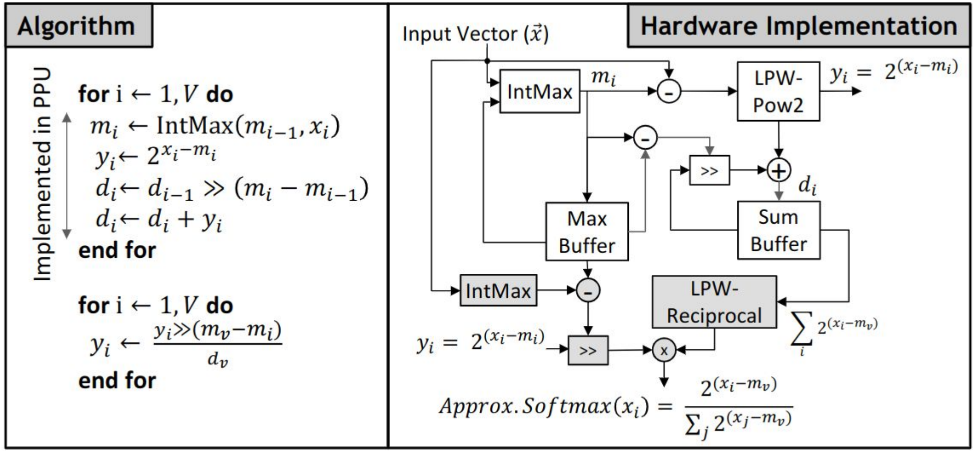


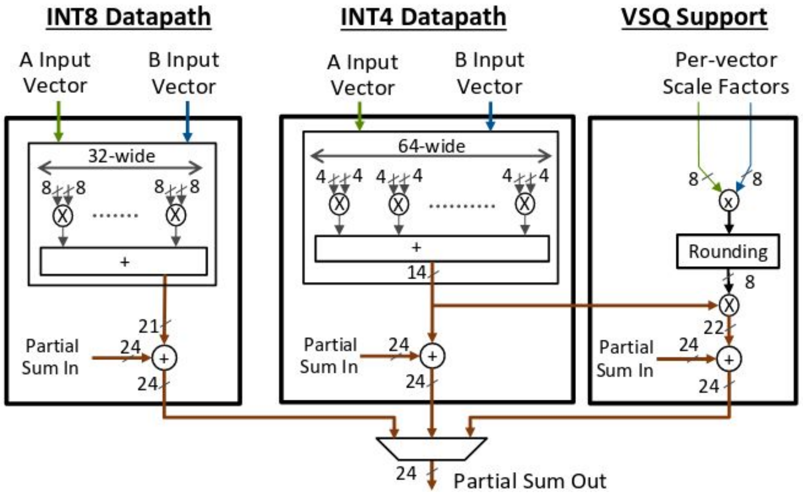
(c) Segmentation @ 512 × 512

Model	# Params.	Top-1
MobileViT-XXS	1.3 M	69.0
MobileViT-XS	2.3 M	74.8
MobileViT-S	5.6 M	78.4

Nvidia’s Transformer Accelerator

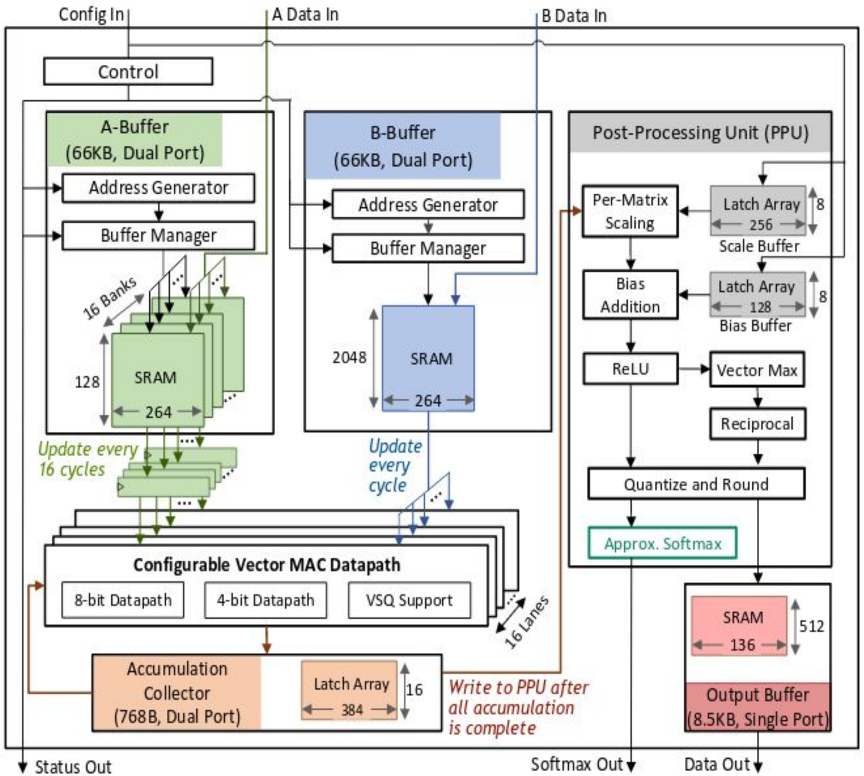
- Per-vector scaled quantization (VSQ)
- softmax: base 2 instead of e
- Replace GELU with ReLU

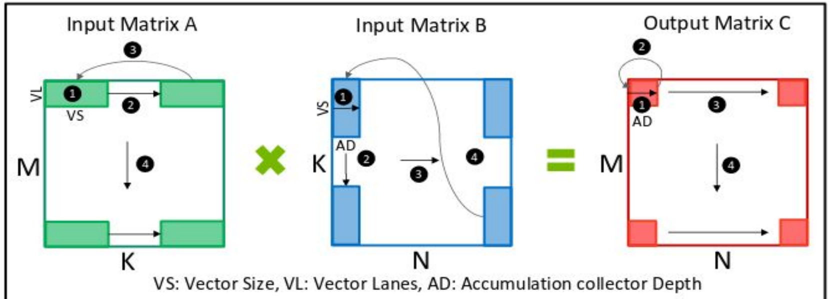




A 17–95.6 TOPS/W Deep Learning Inference Accelerator with Per-Vector Scaled 4-bit Quantization for Transformers in 5nm, Ben Keller Rangharajan Venkatesan Steve Dai Stephen Tell Brian Zimmer William Dally Tom Gray Brucek Khailany

Nvidia’s Transformer Accelerator






```
❹ for m =[0:M/VL)   // Temporal tiling along M dimension
❸ for n =[0:N/AD)   // Temporal tiling along N dimension
❷ for k =[0:K/VS)   // Output stationary
❶ for a =[0:AD)     // A input stationary
  for l = [0:VL)    // Spatial B input activation reuse
  for v = [0:VS)    // Spatial output partial sum reuse
     compute_MAC
```

Dataset, Task	SQuAD v1.1, Reading Comprehension												ImageNet, Image Classification					
Network	BERT-Base						BERT-Large						DeIT-Small			DeIT-Base		
Sequence Length	128			384			128			384			197			197		
Baseline FP32 Accuracy (%)	87.5			87.5			90.3			90.9			79.8			81.8		
Data Bitwidth (4V = 4b VSQ)	4b	4V	8b	4b	4V	8b	4b	4V	8b	4b	4V	8b	4b	4V	8b	4b	4V	8b
Accuracy Loss (%)	80	0.7	0.7	81	0.5	0	88	1.1	1.1	89	0.8	0.1	29	3.6	0.7	25	1.3	0.4
MAC Utilization (%)	-	98	99	-	98	99	-	98	99	-	98	99	-	94	96	-	97	98
Throughput (inferences/s)	-	88	45	-	28	14	-	25	13	-	8.1	4.1	-	210	108	-	56	28
Energy Eff. (inferences/s/W)	-	1.7k	745	-	539	235	-	502	216	-	160	69	-	3.5k	1.5k	-	1.0k	406