DaViT: Dual Attention Vision Transformer

Soumen Mondal & Siddhant Gole
23m2157@iitb.ac.in, 23m2154@iitb.ac.in

Guide: Prof. Biplab Banarjee

Indian Institute of Technology Bombay

August 23, 2024

Introduction

What is DaViT?

- Dual Attention Vision Transformer (DaViT) is a ViT architecture that focuses on capturing global context in images while maintaining computational efficiency.
- Uses spatial window attention with *spatial tokens*, and channel group attention with *channel tokens*. The two forms of attention are complementary (hence the name dual) and alternatively arranged.

DaViT vs Standard ViT

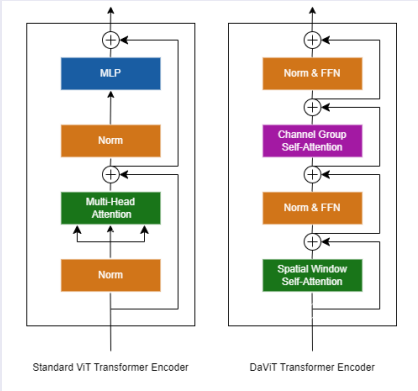


Image Source: Team members - Soumen and Siddhant

Dual Attention Mechanism

Ordinary Dual Attention

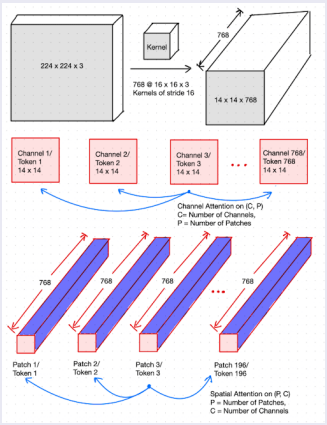


Image Source: Team members - Soumen and Siddhant

Optimized Dual Attention

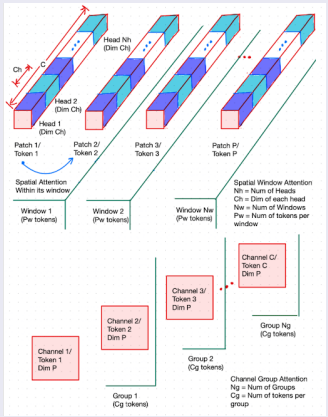


Image Source: Team members - Soumen and Siddhant

Dual Attention Mechanism [contd.]

Spatial Window Attention

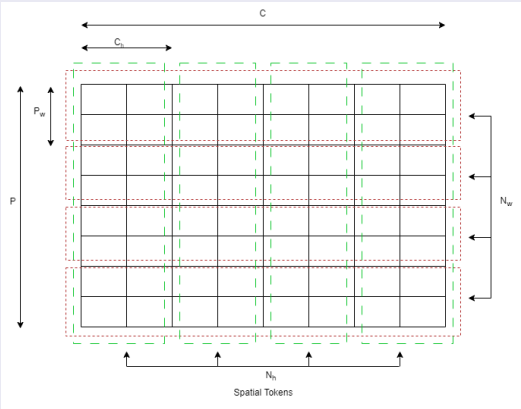


Image Source: Team members - Soumen and Siddhant

Channel Group Attention

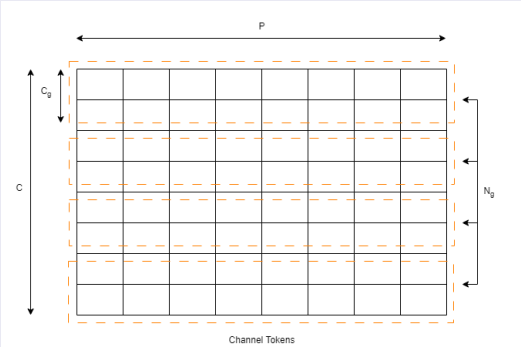


Image Source: Team members - Soumen and Siddhant

Model Architecture

Encoder Architecture

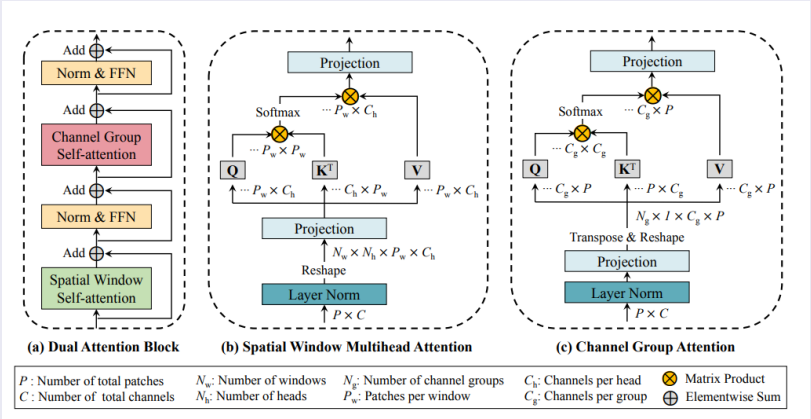


Image Source: DaViT authors - Ding et. al.

Complexity Analysis

Spatial Window Attention

$$\underset{P_w \times C}{[\mathbf{Q_j}]} = \underset{P_w \times C}{[\mathbf{X_j}]}\ \underset{C \times C}{[\mathbf{W^Q}]} \implies \text{takes } P_w C^2 \text{ mults}$$

$$\underset{P_w \times C}{[\mathbf{K_j}]} = \underset{P_w \times C}{[\mathbf{X_j}]}\ \underset{C \times C}{[\mathbf{W^K}]} \implies \text{takes } P_w C^2 \text{ mults}$$

$$\underset{P_w \times C}{[\mathbf{V_j}]} = \underset{P_w \times C}{[\mathbf{X_j}]}\ \underset{C \times C}{[\mathbf{W^V}]} \implies \text{takes } P_w C^2 \text{ mults}$$

$$\underset{P_w \times P_w}{[AttnMap([\mathbf{Q_j}],[\mathbf{K_j}])]} = softmax\left(\frac{[\mathbf{Q_j}][\mathbf{K_j^T}]}{\sqrt{C}}\right) \implies \text{takes } P_w^2 C \text{ mults}$$

$$\underset{P_w \times C}{[\mathbf{Z_j}]} = \underset{P_w \times C}{[Attn([\mathbf{Q_j}],[\mathbf{K_j}],[\mathbf{V_j}])]} = \underset{P_w \times P_w}{[AttnMap([\mathbf{Q_j}],[\mathbf{K_j}])]}\ \underset{P_w \times C}{[\mathbf{V_j}]} \implies \text{takes } P_w^2 C \text{ mults}$$

$$\underset{P \times C}{[Attn([\mathbf{Q}],[\mathbf{K}],[\mathbf{V}])]} = concat(\underset{P_w \times C}{\mathbf{Z_1}}, \underset{P_w \times C}{\mathbf{Z_2}}, \cdots, \underset{P_w \times C}{\mathbf{Z_{N_w}}})$$

Total time taken for spatial window attention with $P = P_w N_w$:

$$T = N_w \times O(3P_w C^2 + 2P_w^2 C) = O(P(2P_w C + 3C^2)) \implies \text{Linear in P!}$$

Critical Analysis

Architecture Design

- The architecture aims to balance global context and local detail, but this balance might not be optimal for all tasks. For instance, tasks that are heavily reliant on local features (e.g., fine-grained image classification) might not benefit as much from the global context modeling provided by channel attention.
- The choice of using non-overlapping windows in spatial attention might lead to a loss of information at the boundaries of these windows.
- The DaViT model does not use shared weights between the channel group attention and spatial window attention mechanisms. Instead, these two attention mechanisms operate independently, each with its own set of projection layers (weights). This increases the computational complexity if P or C is large.
- The patch embedding is obtained in 4 stages where each stage has a different kernel size and channel dimension. While each stage includes multiple DaViT blocks with spatial and channel attention, the outputs of one stage are passed to the next stage in a sequential manner without cross attending stages.

Research Gap and Potential Improvements

- Investigating cross-attention mechanisms where spatial and channel features influence each other directly could lead to improved feature representations.

Model Instantiation

Configurations

3 different network configurations are proposed:

- DaViT-Tiny: C=96, L={1,1,3,1}, $N_g = N_h = \{3, 6, 12, 24\}$
- DaViT-Small: C=96, L={1,1,9,1}, $N_g = N_h = \{3, 6, 12, 24\}$
- DaViT-Base: C=128, L={1,1,9,1}, $N_g = N_h = \{4, 8, 16, 32\}$

When more training data is involved the architectures are scaled up as follows:

- DaViT-Large: C=192, L={1,1,9,1}, $N_g = N_h = \{6, 12, 24, 48\}$
- DaViT-Huge: C=256, L={1,1,9,1}, $N_g = N_h = \{8, 16, 32, 64\}$
- DaViT-Giant: C=384, L={1,1,12,3}, $N_g = N_h = \{12, 24, 48, 96\}$

After each stage the image resolution is decreased while increasing the channel features. Patch Embedding Layer Configuration: Kernel Size: {7,2,2,2}, Stride: {4,2,2,2}

$$1 : \frac{H}{4} * \frac{W}{4} * C \implies 2 : \frac{H}{8} * \frac{W}{8} * 2C \implies 3 : \frac{H}{16} * \frac{W}{16} * 4C \implies 4 : \frac{H}{32} * \frac{W}{32} * 8C$$

Model Instantiation

Stagewise Output

Table 7. Model configurations for our DaViT. We introduce three configurations DaViT-Tiny, DaViT-Small, and DaViT-Base with different model capacities. The size of the input image is set to 224×224.

	Output Size	Layer Name	DaViT-Tiny	DaViT-Small	DaViT-Base
	56×56	Patch Embedding	kernel 7, stride 4, pad 3, $C^1 = 96$	kernel 7, stride 4, pad 3, $C^1 = 96$	kernel 7, stride 4, pad 3, $C^1 = 128$
stage 1	56×56	Dual Transformer Block	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^1 = N_g^1 = 3 \\ C_h^1 = C_g^1 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^1 = N_g^1 = 3 \\ C_h^1 = C_g^1 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^1 = N_g^1 = 4 \\ C_h^1 = C_g^1 = 32 \end{bmatrix} \times 1$
	28×28	Patch Embedding	kernel 2, stride 2, pad 0, $C^2 = 192$	kernel 2, stride 2, pad 0, $C^2 = 192$	kernel 2, stride 2, pad 0, $C^2 = 256$
stage 2	28×28	Dual Transformer Block	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^2 = N_g^2 = 6 \\ C_h^2 = C_g^2 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^2 = N_g^2 = 6 \\ C_h^2 = C_g^2 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^2 = N_g^2 = 8 \\ C_h^2 = C_g^2 = 32 \end{bmatrix} \times 1$
	14×14	Patch Embedding	kernel 2, stride 2, pad 0, $C^3 = 384$	kernel 2, stride 2, pad 0, $C^3 = 384$	kernel 2, stride 2, pad 0, $C^3 = 512$
stage 3	14×14	Dual Transformer Block	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^3 = N_g^3 = 12 \\ C_h^3 = C_g^3 = 32 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^3 = N_g^3 = 12 \\ C_h^3 = C_g^3 = 32 \end{bmatrix} \times 9$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^3 = N_g^3 = 16 \\ C_h^3 = C_g^3 = 32 \end{bmatrix} \times 9$
	7×7	Patch Embedding	kernel 2, stride 2, pad 0, $C^4 = 768$	kernel 2, stride 2, pad 0, $C^4 = 768$	kernel 2, stride 2, pad 0, $C^4 = 1024$
stage 4	7×7	Dual Transformer Block	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 24 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 24 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 32 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$

Image Source: DaViT authors - Ding et. al.

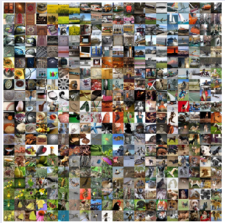
Experiments and Results

Tasks

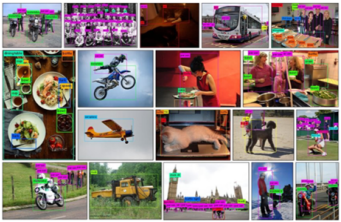
The DaViT architecture was experimented on three different tasks and datasets:

- Image Classification on the ImageNet-1K dataset.
- Object Detection on COCO-2017 object detection dataset.
- Semantic segmentation on the ADE-20K dataset.

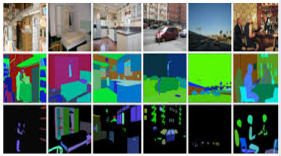
Dataset



ImageNet-1k



COCO-2017



ADE-20K

Image Source: ImageNet, COCO, ADE (from Google)

Experiments and Results - Image Classification

Training Configuration

- Batch Size: 2048
- Epochs: 300
- Optimizer: AdamW with weight decay 0.05
- Maximal gradient norm clipping: 1.0
- Stochastic depth drop rates are set as 0.1,0.2,0.4 for tiny, small, base models respectively.
- random cropping to size 224x224 is employed during training and center crop during validation

Evaluation Results

Table 1. Comparison of image classification on ImageNet-1K for different models. All models are trained and evaluated with 224×224 resolution on ImageNet-1K by default, unless otherwise noted. For fair comparison, token labeling [30] and distillation [55] are not used for all models and their counterparts. † and ‡ denote the model is evaluated with resolution of 384×384 and 512×512, respectively.

Model	#Params (M)	FLOPs (G)	Top-1 (%)
ResNet-50 [24]	25.0	4.1	76.2
DeiT-Small/16 [55]	22.1	4.5	79.8
PVT-Small [63]	24.5	3.8	79.8
ConvMixer-768/32 [57]	21.1	–	80.2
CrossViT-Small [5]	26.7	5.6	81.0
Swin-Tiny [40]	28.3	4.5	81.2
CvT-13 [65]	20.0	4.5	81.6
CoAtNet-0 [13]	25.0	4.2	81.6
CaiT-XS-24 [56]	26.6	5.4	81.8
ViL-Small [82]	24.6	5.1	82.0
PVTv2-B2 [62]	25.4	4.0	82.0
UFO-ViT-S [60]	21.0	3.7	82.0
Focal-Tiny [72]	29.1	4.9	82.2
DaViT-Tiny (Ours)	28.3	4.5	**82.8**
ResNet-101 [24]	45.0	7.9	77.4
PVT-Medium [63]	44.2	6.7	81.2
CvT-21 [65]	32.0	7.1	82.5
UFO-ViT-M [60]	37.0	7.0	82.8
Swin-Small [40]	49.6	8.7	83.1
ViL-Medium [82]	39.7	9.1	83.3
CaiT-S36 [56]	68.0	13.9	83.3
CoAtNet-1 [13]	42.0	8.4	83.3
Focal-Small [72]	51.1	9.1	83.5
CSwin-S [17]	35.0	6.9	83.6
VAN-Large [21]	44.8	9.0	83.9
UniFormer-B [32]	50.0	8.3	83.9
DaViT-Small (Ours)	49.7	8.8	**84.2**

Model	#Params (M)	FLOPs (G)	Top-1 (%)
ResNet-152 [24]	60.0	11.0	78.3
PVT-Large [63]	61.4	9.8	81.7
DeiT-Base/16 [55]	86.7	17.4	81.8
CrossViT-Base [5]	104.7	21.2	82.2
T2T-ViT-24 [75]	64.1	14.1	82.3
CPVT-Base [11]	88.0	17.6	82.3
TNT-Base [22]	65.6	14.1	82.8
ViL-Base [82]	55.7	13.4	83.2
UFO-ViT-B [50]	64.0	11.9	83.3
Swin-Base [40]	87.8	15.4	83.4
CaiT-M24 [56]	185.9	36.0	83.4
NFNet-F0 [4]	71.5	12.4	83.6
PVTv2-B5 [62]	82.0	11.8	83.8
Focal-Base [72]	89.8	16.0	83.8
CoAtNet-2 [13]	75.0	15.7	84.1
CSwin-B [17]	78.0	15.0	84.2
DaViT-Base (Ours)	87.9	15.5	**84.6**
Pre-trained on ImageNet-22k			
Swin-Large [40] †	197.0	103.9	86.4
CSWin-B [17] †	78.0	47.0	87.0
CSWin-L [17] †	173.0	96.8	87.5
CoAtNet-3 [13] ‡	168.0	107.4	87.6
DaViT-Base (Ours) †	87.9	46.4	86.9
DaViT-Large (Ours) †	196.8	103.0	87.5
Pre-trained on 1.5B image and text pairs			
DaViT-Huge (Ours) ‡	362	334	90.2
DaViT-Giant (Ours) ‡	1437	1038	**90.4**

Image Source: DaViT authors - Ding et. al.

Experiments and Results - Object Detection

Training Configuration

- DaViT is used as backbone to RetinaNet and MaskRCNN
- Multi-scale training startegy is used just like SWIN transformer that uses random resizing of images during training.
- Two different training schedules: 1x schedule with 12 epochs and 3x schedule with 36 epochs
- Optimizer: AdamW with initial learning rate 10^{-4} and weight decay 0.05.
- Stochastic depth drop 0.1, 0.2, 0.3 for tiny, small, base models respectively

Evaluation Results

Table 2. Comparisons with CNN and Transformer baselines and SoTA methods on COCO object detection. The box mAP (AP^b) and mask mAP (AP^m) are reported for RetinaNet and Mask R-CNN trained with 1× schedule. FLOPs are measured by 800 × 1280. More detailed comparisons with 3× schedule are in Table 5.

Backbone	FLOPs (G)	RetinaNet AP^b	Mask R-CNN AP^b	Mask R-CNN AP^m
ResNet-50 [24]	239/260	36.3	38.0	34.4
PVT-Small [63]	226/245	40.4	40.4	37.8
ViL-Small [82]	252/174	41.6	41.8	38.5
Swin-Tiny [40]	245/264	42.0	43.7	39.8
Focal-Tiny [72]	265/291	43.7	44.8	41.0
DaViT-Tiny (Ours)	244/263	**44.0**	**45.0**	**41.1**
ResNeXt101-32x4d [69]	319/340	39.9	41.9	37.5
PVT-Medium [63]	283/302	41.9	42.0	39.0
ViL-Medium [82]	339/261	42.9	43.4	39.7
Swin-Small [40]	335/354	45.0	46.5	42.1
Focal-Small [72]	367/401	45.6	47.4	42.8
DaViT-Small (Ours)	332/351	**46.0**	**47.7**	**42.9**
ResNeXt101-64x4d [69]	473/493	41.0	42.8	38.4
PVT-Large [63]	345/364	42.6	42.9	39.5
ViL-Base [82]	443/365	44.3	45.1	41.0
Swin-Base [40]	477/496	45.0	46.9	42.3
Focal-Base [72]	514/533	46.3	47.8	43.2
DaViT-Base (Ours)	471/491	**46.7**	**48.2**	**43.3**

Image Source: DaViT authors - Ding et. al.

Experiments and Results - Semantic Segmentation

Training Configuration

- Segmentation method: UpperNet; Backbone: DaViT
- rescale images to size 512x512
- Batch size: 16 and model is trained for 160k iterations

Evaluation Results

Table 4. Comparison with SoTA methods for semantic segmentation on ADE20K [83] val set. Single-scale evaluation is used. FLOPs are measured by 512 × 2048.

Backbone	Method	#Params (M)	FLOPs (G)	mIoU (%)
Swin-Tiny [40]	UperNet [67]	60	945	44.5
PVT-Large [63]	SemanticFPN [36]	65	318	44.8
HRNet-w48 [60]	OCRNet [78]	71	664	45.7
Focal-Tiny [72]	UperNet [67]	62	998	45.8
XCiT-S12/16 [1]	UperNet [67]	52	–	45.9
Twins-SVT-Small [10]	UperNet [67]	54	912	46.2
DaViT-Tiny (Ours)	UperNet [67]	60	940	**46.3**
ResNet-101 [24]	UperNet [67]	86	1029	44.9
XCiT-S24/16 [1]	UperNet [67]	73	–	46.9
Swin-Small [40]	UperNet [67]	81	1038	47.6
Twins-SVT-Base [10]	UperNet [67]	88	1044	47.7
Focal-Small [72]	UperNet [67]	85	1130	48.0
ResNeSt-200 [81]	DLab.v3+ [6]	88	1381	48.4
DaViT-Small (Ours)	UperNet [67]	81	1030	**48.8**
Swin-Base [40]	UperNet [67]	121	1188	48.1
XCiT-M24/8 [1]	UperNet [67]	109	–	48.4
Twins-SVT-Large [10]	UperNet [67]	133	1188	48.8
ViT-Hybrid [45]	DPT [45]	124	1231	49.0
Focal-Base [72]	UperNet [67]	126	1354	49.0
DaViT-Base (Ours)	UperNet [67]	121	1175	**49.4**

Image Source: DaViT authors - Ding et. al.

Comparison to Swin and DEiT

Proposed Idea in SWIN

Use shifted window multi-head attention i.e. in every subsequent layer shift the window by some degree to capture cross-window attention.

Proposed Idea in DEiT

Uses Distillation technique by introducing a distillation token that the model tries to learn from a pre-trained CNN's output such as ResNet-50 which acts as the teacher model.

Comparison to Swin and DEiT

Attention Map

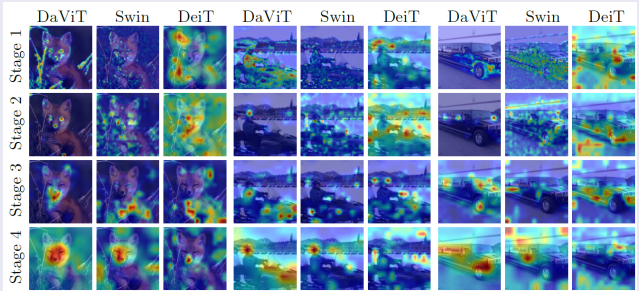


Image Source: DaViT authors - Ding et. al.

Why this architecture?

Different Design Techniques

- Apply window attention first
- Apply channel attention first
- Both attention parallely arranged

Table 6. Quantitative comparisons of different dual attention layouts on ImageNet.

Model	#Params (M)	FLOPs (G)	Top-1 (%)
Window → Channel	28.3	4.5	**82.8**
Channel → Window	28.3	4.5	82.6
Hybrid (parallel)	28.3	4.5	82.6

Image Source: DaViT authors - Ding et. al.

Is Channel Attention needed?

Affect of Channel Attention in different stages

Table 5. The effect of channel group attention at different network stages. Taking a transformer layout with all spatial window attention blocks as the baseline (n/a), we replace two spatial attention blocks at different stages with a dual attention block to show its effectiveness. The first two spatial attention blocks are selected in the third stage to compare with other stages fairly.

Stage	#Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
n/a	28.3	4.6	81.1	95.6
1	28.3	4.5	81.7	95.9
2	28.3	4.5	82.2	96.1
3	28.3	4.6	82.1	96.0
4	28.3	4.6	81.9	95.9
1–4	28.3	4.6	**82.8**	**96.2**

Image Source: DaViT authors - Ding et. al.

Affect of layers in each stage

Table 10. Impact of the change of model depth. We gradually reduce the number of transformer layers at the third stage from the original 3 (6 in Swin [40] and Focal [72]) to 2 (4) and further 1 (2).

Depths	Model	#Params. (M)	FLOPs (G)	Top-1 (%)
2-2-2-2	Swin [40]	21.2	3.1	78.7
	Focal [72]	21.7	3.4	79.9
1-1-1-1	DaViT (ours)	21.2	3.1	**80.2**
2-2-4-2	Swin [40]	24.7	3.8	80.2
	Focal [72]	25.4	4.1	81.4
1-1-2-1	DaViT (ours)	24.7	3.8	**81.8**
2-2-6-2	Swin [40]	28.3	4.5	81.2
	Focal [72]	29.1	4.9	82.2
1-1-3-1	DaViT (ours)	28.3	4.5	**82.8**

Image Source: DaViT authors - Ding et. al.

Conclusion

- The paper proposes two attention mechanisms Spatial and Channel attention.
- Spatial attention captures fine grained features through local interactions.
- While channel attention builds upon these local interactions to capture global context.

Thank You!