DaViT: Dual Attention Vision Transformers – Supplementary Material –

Mingyu Ding[1], Bin Xiao[2†], Noel Codella[2], Ping Luo[1†], Jingdong Wang[3], Lu Yuan[2]
[1]The University of Hong Kong [2]Microsoft Cloud + AI [3]Baidu
mingyuding@hku.hk {bixi,ncodella,luyuan}@microsoft.com
pluo@cs.hku.hk wangjingdong@outlook.com

Table 1. 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	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 24 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 24 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$	$\begin{bmatrix} \text{win. sz. } 7 \times 7, P_w = 49 \\ N_h^4 = N_g^4 = 32 \\ C_h^4 = C_g^4 = 32 \end{bmatrix} \times 1$

A Details of Model Configuration

In this work, we simply follow the design strategy suggested by previous works [3,8]. We divide the entire architecture into four stages, where a patch embedding layer is inserted at the beginning of each stage. Here, our patch embedding layer is implemented by stride convolution. The convolutional kernels and stride values of our four patch embedding layers are $\{7, 2, 2, 2\}$ and $\{4, 2, 2, 2\}$, respectively. Note the large kernel in the first layer introduces almost no additional calculations as the number of input channels is only 3. For the rest kernel values, we use 2 to perform non-overlapping patch merging. We stack our dual attention blocks in each stage with the resolution and feature dimension kept the same. These stages jointly produce a hierarchical representation, with the same feature map resolutions as those of typical convolutional networks, e.g., VGG [4] and ResNet [1]. As a result, the proposed architecture can conveniently replace the backbone networks in existing methods for various vision tasks.

Table 2. Comparison on ImageNet-1K by replacing half of DeiT [5] attention blocks with our channel attention block. Performance improvement over DeiT is highlighted in blue font.

Model	#Params (M)	FLOPs (G)	Top-1 (%)
DeiT-Tiny [5]	5.7	1.2	72.2
DeiT-Tiny [5] + Channel Attention	5.7	1.2	**74.9** (+2.7)
DeiT-Small [5]	22.1	4.5	79.8
DeiT-Small [5] + Channel Attention	22.1	4.5	**81.2** (+1.4)
DeiT-Base [5]	86.7	17.4	81.8
DeiT-Base [5] + Channel Attention	86.7	17.4	**82.3** (+0.5)

Specifically, take an image with $H \times W$, a C^1-dimensional feature with a resolution of $\frac{H}{4} \times \frac{W}{4}$ is obtained after the first patch embedding layer. And its resolution is further reduced into $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$ with the feature dimension increasing to C^2, C^3, and C^4 after the other three patch embedding layer, respectively. For simplicity, we set the window size of 7×7 thus $P_w = 49$ for all models. We also set the number of channels per group $C_g = 32$ and the number of channels per head $C_h = 32$ for all blocks of our three models. For DaViT-tiny and DaViT-small, we set the number of heads/groups $N_h = N_g = \{3, 6, 12, 24\}$ for four stages, respectively; and we set the number of heads/groups $N_h = N_g = \{4, 8, 16, 32\}$ for four stages in DaViT-base. Also, we set the number of dual attention blocks $\{1, 1, 3, 1\}$ for our tiny model and $\{1, 1, 9, 1\}$ for the small and base models.

For the models without FFN, we simply change the number of dual attention blocks to keep the total computation costs similar, though we believe there should be a better configuration specifically for them. We set the number of dual attention blocks $\{2, 2, 11, 2\}$ for our tiny model (without FFN) and $\{2, 2, 28, 2\}$ for the small and base models (without FFN).

When more training data is involved, we further scale up DaViT to large, huge, and giant sizes to validate the scaling ability of the proposed architecture for image classification. We set $C^1 = \{192, 256, 384\}$, $N_h^1 = N_g^1 = \{6, 8, 12\}$, and the number of dual attention blocks of the third stage as $\{9, 9, 12\}$ respectively for large, huge, and giant models. All model training is accelerated by NVIDIA Apex Automatic Mixed Precision (AMP).

B Channel Attention on Vanilla ViT

We apply our channel group attention on the vanilla DeiT [5] to show the generalizability and effectiveness of our dual attention mechanism. We alternatively arrange the vanilla patch-level self-attention and our channel group self-attention. We set the number of groups and channels of our channel-wise group attention the same as the number of heads and channels of self-attention in DeiT, to make the number of parameters and FLOPs comparable with the vanilla DeiT.

Table 3. Comparison of Transformers without FFNs on ImageNet-1K. FFN is removed and more attention layers are added to match the computational cost.

Model (w/o FFN)	#Params	FLOPs	Top-1 (%)
Window Attention-Tiny (w/o FFN)	25.8	4.6	79.1
Channel Attention-Tiny (w/o FFN)	25.8	4.5	79.3
Dual Attention-Tiny (w/o FFN)	25.8	4.5	80.8
Dual Attention-Small (w/o FFN)	46.3	8.7	81.9
Dual Attention-Base (w/o FFN)	81.6	15.2	82.5

Table 4. 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 [3] and Focal [8]) to 2 (4) and further 1 (2).

Depths	Model	#Params. (M)	FLOPs (G)	Top-1 (%)
2-2-2-2	Swin [3]	21.2	3.1	78.7
	Focal [8]	21.7	3.4	79.9
1-1-1-1	DaViT (ours)	21.2	3.1	**80.2**
2-2-4-2	Swin [3]	24.7	3.8	80.2
	Focal [8]	25.4	4.1	81.4
1-1-2-1	DaViT (ours)	24.7	3.8	**81.8**
2-2-6-2	Swin [3]	28.3	4.5	81.2
	Focal [8]	29.1	4.9	82.2
1-1-3-1	DaViT (ours)	28.3	4.5	**82.8**

From Table 2 we observe substantial gains of our dual attention across all model sizes, *e.g.*, 2.7% over the tiny model and 0.5% even compared to the base model. The result shows that our channel-wise attention can be combined with window attention and patch-level global attention, improving the performance of both spatial-wise self-attentions.

C Transformer without FFNs

FFN has been a default component of Transformers with little research on it. However, it dominates the number of FLOPs and model parameters of our DaViT. Considering our dual attention has both channel-wise and spatial-wise interactions, we conduct an initial exploration to show the potential of the pure-attention structure without FFNs. We remove FFNs and add more dual attention blocks to match the computational costs.

From Table 3 we see that: pure dual attention without FFNs achieves 1.5% and 1.7% better Top-1 accuracy than pure window attention and channel attention, respectively, showing the effectiveness of dual attention that has both channel-wise and spatial-wise interactions. The model without FFN shows comparable and even better performance with models like PVT [7] and DeiT [5], but still inferior to recent SoTAs like Swin [3], Focal [8], and our full DaViT.

Table 5. Throughputs of Swin and DaViT.

Model	Throughput (samples/s)
Swin-T	1024
DaViT-T	1059
Swin-S	655
DaViT-S	685
Swin-B	496
DaViT-B	523

Table 6. Comparisons to the channel-wise attention used in CNNs, by replacing our channel group attention with SENet [2] and ECANet [6], respectively.

Method	Top-1 (%)
SE Block [2]	81.2
ECA Block [6]	81.2
Channel Self-Attention	82.8

D Model Capacity against Model Depth

Our DaViT contains both spatial-wise and channel-wise attention, and both local and global interactions in each dual attention block. We conduct experiments to show whether it needs less number of layers to obtain similar modeling capacity as previous works [3,8]. We gradually reduce the number of transformer layers at the third stage from original 3 (6 in Swin [3] and Focal [8]) to 2 (4 in Swin [3] and Focal [8]) and further 1 (2 in Swin [3] and Focal [8]). From Table 4, we find our model outperforms existing SoTA models consistently with the same depth. Moreover, using two fewer layers, our model achieves comparable and even better performance to Swin Transformer. Also, with fewer computational costs, our model outperforms Focal Transformer by a large margin.

E Throughput Analysis

In addition to the main criteria of computational cost in the main paper, *i.e.*, FLOPs and #parameters, we report the real-time inference speed/throughput against Swin Transformer [3] to show the efficiency of our work. Compared to the strong baseline Swin, our model does have advantages in real-time throughput as the cleaner structure and high efficiency of the group channel attention. For example, we remove the shifted window partition, which depends on `torch.roll()` to perform cyclic shift and its reverse on features. This operation is not fully optimized by popular inference frameworks. Table 5 demonstrates the comparison between Swin and DaViT. Nvidia V100 GPU is utilized for the benchmark, and the image resolution is 224×224. It shows that DaViT consistently outperforms Swin across different model sizes in efficiency.

F Comparisons with SE and ECA Blocks

To make quantitative comparisons with traditional channel-wise attentions blocks, we did experiments by replacing the channel self-attention in our tiny model with SE block [2] and ECA block [6]. The results in Table 6 show that our channel group self-attention is more powerful by performing dynamic feature fusion

Table 7. Top-1 on ImageNet. 'Window-only' and 'Window + Global' denotes local attention and straightforward global baseline, respectively. 'Window + Channel' is our DaViT. For fair comparison, the three models have the same layout, patch/window size. Our DaViT has similar efficiency to local transformers (Window-only), while maintaining the same high performance as global methods (Window+Global).

Attention Type (Tiny)	Params	FLOPs	Top-1 Acc.
Window-only	28.3M	**4.5G**	81.2%
Window+Global	28.3M	6.7G	**82.8%**
Window+Channel (Ours)	28.3M	**4.5G**	**82.8%**

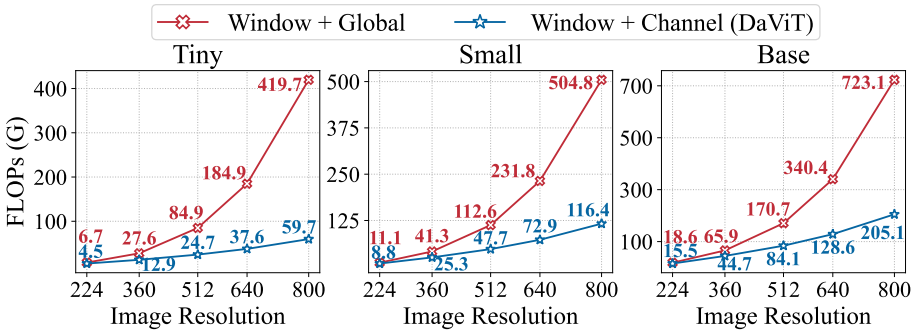

Fig. 1. FLOPs comparison between Window+Global (conventional global attention) and Window+Channel (our channel group attention) as input resolution increases across three model sizes.

across global tokens (different global views of the entire image) in transformers. It shows superior performance (1.6%) than both of the two variants using SE [2] and ECA [6] blocks, respectively.

G Baselines and Relationships with DaViT

Attention in DaViT does capture both global and local fine-grained interactions in a computationally efficient manner, *i.e.*, linear complexity $O(2PC(P_w + C_g))$ with respect to spatial dimension P and channel dimension C. Below, we detail the relationship between DaViT and its baselines.

1) Conventional global attention is of quadratic complexity with respect to the number of tokens, while local attention has linear complexity by sacrificing global interaction, such as window attention in Swin Transformer [40]. We call this attention type **Window-only**, which has inferior performance but high efficiency. 2) We propose DaViT, which captures global interactions while preserving fine-grained local details. We introduce global interactions by channel group attention with linear complexity (same as local window attention). Dual attention in DaViT is represented by **Window+Channel** with both high per-

formance and high efficiency. 3) To show the efficiency of DaViT, we replace all channel group attention with conventional global attention, named **Window+Global**. Global interactions are obtained at a quadratic computational cost, resulting in high performance but low efficiency.

Tab. 7 shows the accuracy and FLOPs of the three models. Our Window + Channel enjoys high performance of global interactions without quadratic complexity, *e.g.*, 1.6% improvement (82.8 vs 81.2) over Window-only with 33% lower FLOPs (4.5 vs 6.7) than Window+Global. The gap on FLOPs widens further as the input resolution increases in Fig. 1. We can see that the complexity of our model increases linearly, while the trend for the baseline is quadratic.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1
2. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141 (2018) 4, 5
3. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021) 1, 3, 4
4. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 1
5. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357. PMLR (2021) 2, 3
6. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. In: CVPR (2020) 4, 5
7. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV (2021) 3
8. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641 (2021) 1, 3, 4