FLatten Transformer: Vision Transformer with Focused Linear Attention
Supplementary Material

A. Proof of Proposition 1

As mentioned in the main paper, with the aim to restore the sharp distribution in linear attention, we present our **Focused Function** f_p:

$$\operatorname{Sim}\left(Q_i, K_j\right)=\phi_p\left(Q_i\right) \phi_p\left(K_j\right)^T, \tag{1}$$

$$\text{where } \phi_p(x)=f_p(\operatorname{ReLU}(x)),\ f_p(x)=\frac{\|x\|}{\left\|x^{**p}\right\|} x^{**p}, \tag{2}$$

and x^{**p} represents the power p of x bit by bit. We follow previous linear attention modules to use the ReLU function first to ensure the non-negativity of input. Therefore, when proving the effects of f_p, we only consider $x, y \geq 0$.

Proposition 1 (Feature direction adjustment with f_p) *Let $x=(x_1, \cdots, x_n), y=(y_1, \cdots, y_n) \in \mathbb{R}^n, x_i, y_j \geq 0$. Assume $0<\langle x, y\rangle<\|x\|\|y\|$ and x, y have the **single** largest value x_m, y_n respectively.*
For a pair of feature $\{x, y\}$ with $m=n$:

$$\exists\, p>1,\ s.t.\ \left\langle\phi_p(x), \phi_p(y)\right\rangle>\langle x, y\rangle . \tag{3}$$

For a pair of feature $\{x, y\}$ with $m \neq n$:

$$\exists\, p>1,\ s.t.\ \left\langle\phi_p(x), \phi_p(y)\right\rangle<\langle x, y\rangle . \tag{4}$$

Proof.

$$\begin{aligned}
&\phi_p(x)=f_p(\operatorname{ReLU}(x))=f_p(x), \\
&\phi_p(y)=f_p(\operatorname{ReLU}(y))=f_p(y).
\end{aligned} \tag{5}$$

$$\left\|f_p(x)\right\|=\frac{\|x\|}{\left\|x^{**p}\right\|}\left\|x^{**p}\right\|=\|x\|,\ \left\|f_p(y)\right\|=\|y\| . \tag{6}$$

Therefore, we have:

$$\begin{aligned}
\left\langle\phi_p(x), \phi_p(y)\right\rangle & =\left\langle f_p(x), f_p(y)\right\rangle \\
& =\left\|f_p(x)\right\|\left\|f_p(y)\right\|\langle u, v\rangle \\
& =\|x\|\|y\|\langle u, v\rangle,
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
\langle u, v\rangle & =\left\langle\frac{f_p(x)}{\left\|f_p(x)\right\|}, \frac{f_p(y)}{\left\|f_p(y)\right\|}\right\rangle \\
& =\frac{\sum_{i=1}^n x_i^p y_i^p}{\sqrt{\left(\sum_{i=1}^n x_i^{2p}\right)\left(\sum_{i=1}^n y_i^{2p}\right)}} \\
& =\frac{\sum_{i=1}^n a_i^p b_i^p}{\sqrt{\left(\sum_{i=1}^n a_i^{2p}\right)\left(\sum_{i=1}^n b_i^{2p}\right)}},
\end{aligned} \tag{8}$$

and

$$\begin{gathered}
a_i=x_i / \max_{1 \leq i \leq n}\left(x_i\right), b_i=y_i / \max_{1 \leq i \leq n}\left(y_i\right), \\
a_i, b_i \in[0,1] .
\end{gathered} \tag{9}$$

Based on our assumption, we have:

$$\exists!m,\ s.t.\ a_m=1,\ \ \exists!n,\ s.t.\ b_n=1. \tag{10}$$

Therefore,

$$\lim_{p \rightarrow \infty} a_i^p=\begin{cases}1, & i=m \\ 0, & i \neq m\end{cases},\quad \lim_{p \rightarrow \infty} b_j^p=\begin{cases}1, & j=n \\ 0, & j \neq n\end{cases}. \tag{11}$$

Then we consider the following two cases:
(1) $m=n$:

$$\begin{aligned}
\lim_{p \rightarrow \infty}\langle u, v\rangle & =\lim_{p \rightarrow \infty} \frac{\sum_{i=1}^n a_i^p b_i^p}{\sqrt{\left(\sum_{i=1}^n a_i^{2p}\right)\left(\sum_{i=1}^n b_i^{2p}\right)}} \\
& =\frac{1 \times 1}{\sqrt{1 \times 1}}=1 .
\end{aligned} \tag{12}$$

Eq. (7), Eq. (12) $\Rightarrow$

$$\begin{aligned}
\lim_{p \rightarrow \infty}\left\langle\phi_p(x), \phi_p(y)\right\rangle & =\lim_{p \rightarrow \infty}\|x\|\|y\|\langle u, v\rangle \\
& =\|x\|\|y\|>\langle x, y\rangle .
\end{aligned} \tag{13}$$

Thus we have,

$$\exists p>1,\ s.t.\ \left\langle\phi_p(x), \phi_p(y)\right\rangle>\langle x, y\rangle . \tag{14}$$

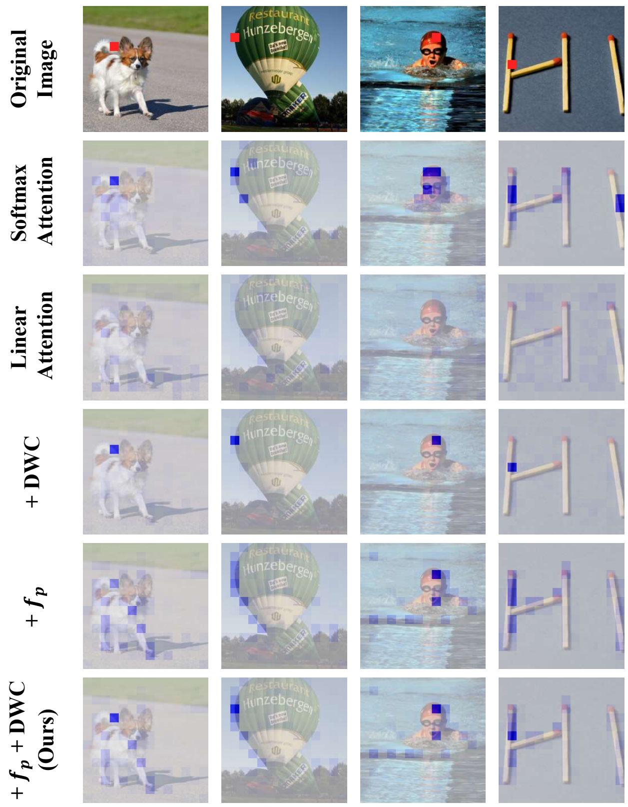

Figure 1. The distribution of attention weights from DeiT-tiny. Feature corresponding to the red block is used as query.

(2) $m \neq n$:

$$\lim_{p\to\infty} \langle u, v\rangle = \lim_{p\to\infty} \frac{\sum_{i=1}^{n} a_i^p b_i^p}{\sqrt{\left(\sum_{i=1}^{n} a_i^{2p}\right)\left(\sum_{i=1}^{n} b_i^{2p}\right)}} = \frac{1\times 0 + 0\times 1}{\sqrt{1\times 1}} = 0. \tag{15}$$

Eq. (7), Eq. (15) $\Rightarrow$

$$\lim_{p\to\infty} \langle \phi_p(x), \phi_p(y)\rangle = \lim_{p\to\infty} \|x\|\,\|y\|\,\langle u, v\rangle = 0 < \langle x, y\rangle\,. \tag{16}$$

Thus we have,

$$\exists p > 1,\ s.t.\ \langle \phi_p(x), \phi_p(y)\rangle < \langle x, y\rangle\,. \tag{17}$$

$\square$

Therefore, with a proper p, our focused function $f_p(\cdot)$ practically achieves a more distinguished difference between similar query-key pairs (Eq. (3)) and dissimilar query-key pairs (Eq. (4)). Actually, f_p divides the features into several groups according to their nearest axes, improving the similarity within each group while reducing the similarity between the groups, thus restoring the sharp attention distribution as the original Softmax function.

Method	Reso	#Params	Flops	Top-1
DeiT-T [3]	224^2	5.7M	1.2G	72.2
FLatten-DeiT-T	224^2	6.1M	1.1G	**74.1** (+1.9)
PVT-T [4]	224^2	13.2M	1.9G	75.1
FLatten-PVT-T	224^2	12.2M	2.0G	**77.8** (+2.7)
PVT-S	224^2	24.5M	3.8G	79.8
FLatten-PVT-S	224^2	21.7M	4.0G	**81.7** (+1.9)
PVT-M	224^2	44.2M	6.7G	81.2
FLatten-PVT-M	224^2	37.2M	7.0G	**83.0** (+1.8)
PVT-L	224^2	61.4M	9.8G	81.7
FLatten-PVT-L	224^2	50.6M	10.4G	**83.4** (+1.7)
PVTv2-B0 [5]	224^2	3.4M	0.6G	70.5
FLatten-PVTv2-B0	224^2	3.6M	0.6G	**71.1** (+0.6)
PVTv2-B1	224^2	13.1M	2.1G	78.7
FLatten-PVTv2-B1	224^2	12.9M	2.2G	**79.5** (+0.7)
PVTv2-B2	224^2	25.4M	4.0G	82.0
FLatten-PVTv2-B2	224^2	22.6M	4.3G	**82.5** (+0.5)
PVTv2-B3	224^2	45.2M	6.9G	83.2
FLatten-PVTv2-B3	224^2	38.3M	7.3G	**83.7** (+0.5)
PVTv2-B4	224^2	62.6M	10.1G	83.6
FLatten-PVTv2-B4	224^2	51.8M	10.7G	**84.0** (+0.4)
Swin-T [2]	224^2	29M	4.5G	81.3
FLatten-Swin-T	224^2	29M	4.5G	**82.1** (+0.8)
Swin-S	224^2	50M	8.7G	83.0
FLatten-Swin-S	224^2	51M	8.7G	**83.5** (+0.5)
Swin-B	224^2	88M	15.4G	83.5
FLatten-Swin-B	224^2	89M	15.4G	**83.8** (+0.3)
Swin-B	384^2	88M	47.0G	84.5
FLatten-Swin-B	384^2	91M	46.5G	**85.0** (+0.5)
CSwin-T [1]	224^2	23M	4.3G	82.7
FLatten-CSwin-T	224^2	21M	4.3G	**83.1** (+0.4)
CSwin-S	224^2	35M	6.9G	83.6
FLatten-CSwin-S	224^2	35M	6.9G	**83.8** (+0.2)
CSwin-B	224^2	78M	15.0G	84.2
FLatten-CSwin-B	224^2	75M	15.0G	**84.5** (+0.3)
CSwin-B	384^2	78M	47.0G	85.4
FLatten-CSwin-B	384^2	78M	46.4G	**85.5** (+0.1)

Table 1. Comparisons of focused linear attention with other vision transformer backbones on the ImageNet-1K classification task.

B. More Visualizations

We visualize more examples of attention weights in Fig. 1. To better show the contribution of our focused function and DWC, we start from the vanilla linear attention and introduce f_p and DWC separately. As demonstrated in the last three rows, DWC improves local focus ability but cannot focus on any position, while f_p practically enhances model's focus ability, helping model focus on more informative regions. Combining f_p and DWC, our focused linear attention module restores the sharp distribution as the original Softmax attention.

C. Full Classification Results

Due to the page limit, we only present representative ImageNet classification results in Figure 6 of main paper. Here, we give all the classification results when applying our focused linear attention module on various sizes of the five baseline models in Tab.1.

D. Model Architectures

We summarize the architectures of five Transformer models adopted in the main paper, including DeiT [3], PVT [4], PVTv2 [5], Swin Transformer [2], CSwin Transformer [1] in Tab.2-8. In practice, we substitute the original self-attention blocks at all stages of the DeiT, PVT and PVTv2 with the focused linear attention block, but only adopt our module at early stages of Swin and CSwin. The model structure (width and depth) are kept unchanged, except for CSwin-T and CSwin-B, where we increase the depth of the first and second stages and correspondingly reduce the depth of the third stage to better reflect our module's advantage of enlarged receptive field.

References

[1] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. 2, 3

[2] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3

[3] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2, 3

[4] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 2, 3

[5] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2, 3

stage	output	FLatten-DeiT-T	
		FLatten	DeiT Block
res1	14×14	$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 192 \\ \text{head } 3 \end{bmatrix} \times 12$	None

Table 2. Architectures of FLatten-DeiT models.

stage	output	FLatten-PVT-M		FLatten-PVT-L	
		FLatten	PVT Block	**FLatten**	PVT Block
res1	56×56	Conv1×1, stride=4, 64, LN			
		$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 3$	None
res2	28×28	Conv1×1, stride=2, 128, LN			
		$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 3$	None
res3	14×14	Conv1×1, stride=2, 320, LN			
		$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 6$	None
res4	7×7	Conv1×1, stride=2, 512, LN			
		$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 3$	None

Table 3. Architectures of FLatten-PVT models (Part1).

stage	output	FLatten-PVT-M		FLatten-PVT-L	
		FLatten	PVT Block	**FLatten**	PVT Block
res1	56×56	Conv1×1, stride=4, 64, LN			
		$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 3$	None	$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1 \end{bmatrix} \times 3$	None
res2	28×28	Conv1×1, stride=2, 128, LN			
		$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 3$	None	$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2 \end{bmatrix} \times 8$	None
res3	14×14	Conv1×1, stride=2, 320, LN			
		$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 18$	None	$\begin{bmatrix} \text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5 \end{bmatrix} \times 27$	None
res4	7×7	Conv1×1, stride=2, 512, LN			
		$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 3$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8 \end{bmatrix} \times 3$	None

Table 4. Architectures of FLatten-PVT models (Part2).

stage	output	FLatten-PVTv2-B0		FLatten-PVTv2-B1		FLatten-PVTv2-B2	
		FLatten	PVTv2 Block	**FLatten**	PVTv2 Block	**FLatten**	PVTv2 Block
res1	56×56	Conv4×4, stride=4, 32, LN		Conv4×4, stride=4, 64, LN			
		$\begin{bmatrix}\text{win } 56\times56 \\ \text{dim } 32 \\ \text{head } 1\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1\end{bmatrix}\times 3$	None
res2	28×28	Conv1×1, stride=2, 64, LN		Conv1×1, stride=2, 128, LN			
		$\begin{bmatrix}\text{win } 28\times28 \\ \text{dim } 64 \\ \text{head } 2\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2\end{bmatrix}\times 3$	None
res3	14×14	Conv2×2, stride=2, 160, LN		Conv2×2, stride=2, 320, LN			
		$\begin{bmatrix}\text{win } 14\times14 \\ \text{dim } 160 \\ \text{head } 5\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5\end{bmatrix}\times 6$	None
res4	7×7	Conv2×2, stride=2, 256, LN		Conv2×2, stride=2, 512, LN			
		$\begin{bmatrix}\text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8\end{bmatrix}\times 2$	None	$\begin{bmatrix}\text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8\end{bmatrix}\times 3$	None

Table 5. Architectures of FLatten-PVTv2 models (Part1).

stage	output	FLatten-PVTv2-B3		FLatten-PVTv2-B4	
		FLatten	PVTv2 Block	**FLatten**	PVTv2 Block
res1	56×56	Conv4×4, stride=4, 64, LN			
		$\begin{bmatrix}\text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1\end{bmatrix}\times 3$	None	$\begin{bmatrix}\text{win } 56\times56 \\ \text{dim } 64 \\ \text{head } 1\end{bmatrix}\times 3$	None
res2	28×28	Conv2×2, stride=2, 128, LN			
		$\begin{bmatrix}\text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2\end{bmatrix}\times 3$	None	$\begin{bmatrix}\text{win } 28\times28 \\ \text{dim } 128 \\ \text{head } 2\end{bmatrix}\times 8$	None
res3	14×14	Conv2×2, stride=2, 320, LN			
		$\begin{bmatrix}\text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5\end{bmatrix}\times 18$	None	$\begin{bmatrix}\text{win } 14\times14 \\ \text{dim } 320 \\ \text{head } 5\end{bmatrix}\times 27$	None
res4	7×7	Conv1×1, stride=2, 512, LN			
		$\begin{bmatrix}\text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8\end{bmatrix}\times 3$	None	$\begin{bmatrix}\text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 8\end{bmatrix}\times 3$	None

Table 6. Architectures of FLatten-PVTv2 models (Part2).

stage	output	FLatten-Swin-T		FLatten-Swin-S		FLatten-Swin-B	
		FLatten	Swin Block	**FLatten**	Swin Block	**FLatten**	Swin Block
res1	56×56	concat 4×4, 96, LN		concat 4×4, 96, LN		concat 4×4, 128, LN	
		$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 96 \\ \text{head } 3 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 96 \\ \text{head } 3 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 56\times56 \\ \text{dim } 128 \\ \text{head } 3 \end{bmatrix} \times 2$	None
res2	28×28	concat 4×4, 192, LN		concat 4×4, 192, LN		concat 4×4, 256, LN	
		$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 192 \\ \text{head } 6 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 192 \\ \text{head } 6 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 28\times28 \\ \text{dim } 256 \\ \text{head } 6 \end{bmatrix} \times 2$	None
res3	14×14	concat 4×4, 384, LN		concat 4×4, 384, LN		concat 4×4, 512, LN	
		None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 384 \\ \text{head } 12 \end{bmatrix} \times 6$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 384 \\ \text{head } 12 \end{bmatrix} \times 18$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 12 \end{bmatrix} \times 18$
res4	7×7	concat 4×4, 768, LN		concat 4×4, 768, LN		concat 4×4, 1024, LN	
		None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 768 \\ \text{head } 24 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 768 \\ \text{head } 24 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 1024 \\ \text{head } 24 \end{bmatrix} \times 2$

Table 7. Architectures of FLatten-Swin models.

stage	output	FLatten-CSwin-T		FLatten-CSwin-S		FLatten-CSwin-B	
		FLatten	CSwin Block	**FLatten**	CSwin Block	**FLatten**	CSwin Block
res1	56×56	Conv7×7, stride=4, 64, LN				Conv7×7, stride=4, 96, LN	
		$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 64 \\ \text{head } 2 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 64 \\ \text{head } 2 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 96 \\ \text{head } 4 \end{bmatrix} \times 3$	None
res2	28×28	Conv7×7, stride=4, 128, LN				Conv7×7, stride=4, 192, LN	
		$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 128 \\ \text{head } 4 \end{bmatrix} \times 4$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 128 \\ \text{head } 4 \end{bmatrix} \times 4$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 192 \\ \text{head } 8 \end{bmatrix} \times 6$	None
res3	14×14	Conv7×7, stride=4, 256, LN				Conv7×7, stride=384, LN	
		None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 256 \\ \text{head } 8 \end{bmatrix} \times 18$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 256 \\ \text{head } 8 \end{bmatrix} \times 32$	None	$\begin{bmatrix} \text{win } 3\times3 \\ \text{dim } 384 \\ \text{head } 16 \end{bmatrix} \times 29$
res4	7×7	Conv7×7, stride=4, 512, LN				Conv7×7, stride=4, 768, LN	
		None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 16 \end{bmatrix} \times 1$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 512 \\ \text{head } 16 \end{bmatrix} \times 2$	None	$\begin{bmatrix} \text{win } 7\times7 \\ \text{dim } 768 \\ \text{head } 32 \end{bmatrix} \times 2$

Table 8. Architectures of FLatten-CSwin models.