Fcaformer: Forward Cross Attention in Hybrid Vision Transformer

Haokui Zhang[1,2], Wenze Hu[1], Xiaoyu Wang[3]
[1]Intellifusion [2]Yan‘an University
[3]The Hong Kong University of Science and Technology (Guangzhou)

Abstract

Currently, one main research line in designing a more efficient vision transformer is reducing the computational cost of self attention modules by adopting sparse attention or using local attention windows. In contrast, we propose a different approach that aims to improve the performance of transformer-based architectures by densifying the attention pattern. Specifically, we proposed forward cross attention for hybrid vision transformer (FcaFormer), where tokens from previous blocks in the same stage are secondary used. To achieve this, the FcaFormer leverages two innovative components: learnable scale factors (LSFs) and a token merge and enhancement module (TME). The LSFs enable efficient processing of cross tokens, while the TME generates representative cross tokens. By integrating these components, the proposed FcaFormer enhances the interactions of tokens across blocks with potentially different semantics, and encourages more information flows to the lower levels. Based on the forward cross attention (Fca), we have designed a series of FcaFormer models that achieve the best trade-off between model size, computational cost, memory cost, and accuracy. For example, without the need for knowledge distillation to strengthen training, our FcaFormer achieves 83.1% top-1 accuracy on Imagenet with only 16.3 million parameters and about 3.6 billion MACs. This saves almost half of the parameters and a few computational costs while achieving 0.7% higher accuracy compared to distilled EfficientFormer. Code is available at https://github.com/hkzhang-git/FcaFormer

1. Introduction

With the rapid adoption of transformer structures in the computer vision community, several types of attention patterns have been proposed to enhance the performance or speed of transformer models. For instance, ViT [5] employs the vanilla global multi-head self-attention, Swin Transformers [20] uses local windowed attention, MaxViT [32] incorporates grid attention across interleaved tokens, and Dynamic ViT [28] utilizes attention on progressively pruned tokens. These approaches aim to sparsify the attention patterns of the original ViT to achieve a better trade-off between speed and accuracy.

In contrast, we propose a new model block as well as a family of models called FcaFormer, which improves the performance of vision transformers by further densifying the attention patterns at a limited extra cost. Specifically, we propose to connect the input of the standard multi-head attention (MHA) module with extra tokens transformed from previous blocks in the same stage, while still restricting the attention module to output the original amount of tokens. To further reduce the computational cost, we merge the tokens from previous blocks by using depthwise convolutions with large strides. These tokens are further calibrated by scaling them with learned parameters, before being taken into the attention units in subsequent blocks.

The new forward cross attention connection has several advantages: 1) it helps transformers further exploit the interactions of tokens across different levels; 2) it reuses the previously generated tokens so that some of the information no longer needs to be preserved by the subsequent transformer operations, leading to potentially smaller models with similar accuracy; 3) similar to the residual connections in ResNet, this extra cross layer connection encourages more information flows to the lower levels of the network, which further accelerate the convergence.

The newly densified connections come with a limited increase in computational cost. As explained in Section 3.3, this cost increase is linear rather than quadratic, since we keep the number of output tokens the same as in standard ViTs. Furthermore, most of the computation cost in most hybrid vision transformer architectures is in the feed forward network (FFN) rather than the MHA part of transformer blocks. Thus, the linear growth of computational complexity from densified connections does not significantly affect the overall computation cost. Finally, to further reduce the number of extra inputs, we use depthwise convolutions with large kernels and long strides to aggregate tokens from previous blocks.

We have incorporated the proposed Fca design into two typical classes of transformer models: the plain ViT model used in DeiT, and the hybrid ConvNet and transformer structures frequently seen in recent works [24, 9, 17]. Our experiments demonstrate that the Fca block can seamlessly replace the corresponding transformer blocks in these architectures, leading to significantly improved performance compared to their corresponding baselines. Specifically, FcaFormer-L1 achieves a top-1 accuracy of 80.3% with approximately 6.2 million parameters and about 1.4 billion MACs. This is achieved while saving almost half the number of parameters, and achieving 1.1% higher accuracy compared to the recently proposed EfficientFormer. Table 2 displays the comparison results.

The contribution of this paper is summarized as follows.

- Opposite to many recent works that use sparse attentions to improve transformer models, we propose to design more efficient models by densifying the attention connection patterns, which open up a new and worthwhile research avenue for consideration.
- We propose the FcaFormer block, which leverages existing tokens and enhance interactions across different levels. To achieve this, we introduce two new components: learnable scale factors (LSFs) and a token merge and enhancement module (TME). The LSFs allow us to effectively process cross tokens, while the TME generates representative cross tokens. Together, these components improve the performance of the FcaFormer models.
- Based on the proposed FcaFormer block, we constructed several new models which have demonstrated better performance than various other recently proposed models.

2. Related Works

2.1. Pure vision transformers

Dosovitskiy *et al.* introduced transformer model into vision tasks and proposed the ViT [5]. It cropped an image into 16×16 patches as an input token sequence to the transformer and used positional encoding to model spatial relations among tokens. DeiT [31] lowered the difficulty of ViT model training by knowledge distillation, and achieved competitive accuracy with less pretraining data. To Further improve the model architecture, researchers attempted to optimize the ViT toward improving its computational efficiency. Among them, the Swin transformer [20] computes self attention among shifted local windows. The MaxViT [32] uses block attention and grid attention alternatively to keep spatially global information exchange while significantly reduce the number of tokens involved in the self attention computation. The DynamicViT [28] prunes redundant tokens progressively and dynamically depending on the input features. Mohsen *et al.* [7] proposed a differentiable parameter-free adaptive token sampler and plugged it into ViTs to sample part tokens for attention computation.

Except for the DeiT, all methods above keep pure vision transformer architectures and seek to achieve better accuracy speed trade-off by using sparse attention via reducing the number of tokens in attention patterns. In contrast, we propose to achieve this goal by densifying the attention pattern, reusing existing tokens from previous blocks. Such interactions promote attentions across features of different semantic levels, which is very common in ConvNets and many methods before the wide adoption of deep learning.

2.2. Hybrid ConvNet and vision transformers

Rather than simplifying ViTs, another popular line of research is to combine elements of ViTs and ConvNets to form new backbones. Two early attempts can be found in [9, 39], where ConvNet blocks are employed to extract low level information in early stages and ViT blocks are adopted in deep stages. Such a hybrid structure improves optimization stability and model performance. Similarly, BoTNet [30] replaces the standard convolution with multi-head attention in the last few blocks of ResNet. In [24], the deeper stages of MobileNetv2 [29] are replaced with their proposed MobileViT block. There are other hybrid models which mix convolution operation with self attention and channel mixer operations. For example, ConViT[6] incorporates soft convolutional inductive biases via a gated positional self-attention. CMT [10] and Next-ViT [15] insert both convolution operataion and self attention module into a single block. PVT v1 [36], PVT v2 [37], LIT [27] and LIT v2 [26] insert convolutional operations into each stage of ViT models to reduce the number of tokens, and build hybrid multi-stage structures.

Generally speaking, hybrid models achieve better tradeoff between model cost and accuracy compared with pure vision transformer models. Therefore, we mainly focus on applying forward cross attention design on hybrid models to evaluate its effects. Our work is complementary to these hybrid design attempts, and can be used to replace the transformer part for most of these models.

2.3. Skip connections in ConvNets

In retrospect, our work is also related to several key improvements in ConvNets design that introduces extra connections to improve the information flows.

ResNet [13] employs shortcut connections to overcome the degradation problem, where accuracy gets saturated and then degrades rapidly with the increasing convolutional network depth. DenseNet [14] connects each layer to every other layer in a feedforward fashion. As with ResNet that builds the whole network by stacking several residual units, DenseNet consists of multiple dense blocks. Wang *et*

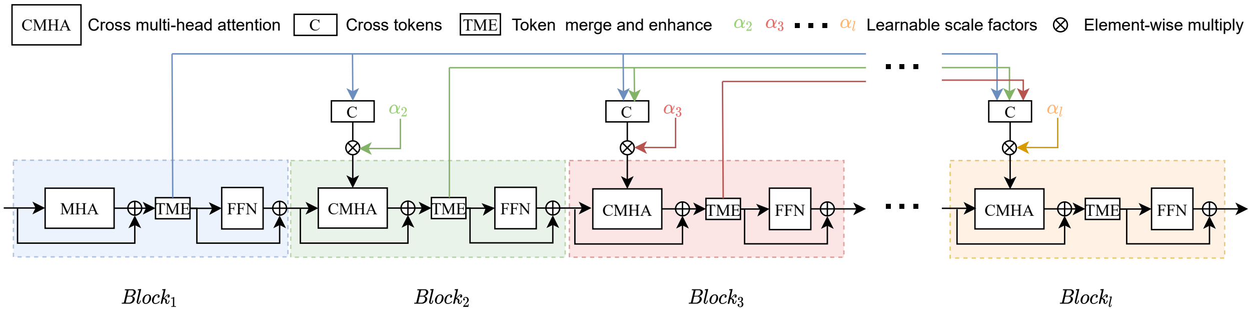

Figure 1. A FcaFormer stage which consists of L FcaFormer blocks. Compared with standard ViT blocks, we add token merge and enhancement (TME) part, which uses long stride large kernel convolutions to merge tokens spatially as cross tokens, and small kernel convolutions to further enhance tokens for channel mixing (FFN). The cross tokens are then used in later blocks as extra tokens for multi-head attention, after being calibrated by multiplying them with learned scaling factors (LSFs, α).

al. [35] further improve DenseNet by using two-way dense layers to obtain different receptive fields. With limited extra computational cost, these model design choices solved bottlenecks existed in ConvNets and are still widely used in both academia and industry. Skip connection is also used in transformer models. For instance, Denseformer [22] reuses the first layer CLS token into subsequent layers.

Our proposed forward cross attention is similar to the works above in that it reuses existing intermediate results and introduces extra connections to the overall network structure. As experiments show, our work introduces similar benefits to transformer models such as better model performance and faster model convergence.

2.4. Cross attention transformers

In transformers, cross attention is usually used to mix two different embedding sequences. In [34, 8], the output of encoder is fed to decoder via cross attention. CrossViT [1] mixes small-patch and large-patch tokens with cross attention to extract multi-scale feature. Our proposed FcaFormer shares the basic idea of integrating information with cross attention. However, unlike previous models, our FcaFormer integrates tokens from different semantic levels using TME and LSFs to overcome their distinct characteristics.

3. The proposed FcaFormer

3.1. FcaFormer block and FcaFormer stage

Fig.1 shows the major parts and connection relations of a FcaFormer stage which consists of L FcaFormer blocks. Each FcaFormer block is composed of three major parts, which are cross multi-head attention (CMHA), token merge and enhancement (TME), and feed forward network (FFN) respectively.

CMHA and LSFs. Different from the standard ViT block, our proposed FcaFormer block receives two sets of tokens as input. The block in turn generates two sets of tokens as well, which are denoted as x^l and $\bar{x}^l$ respectively. Taking the block at depth l as an example, the inputs are the regular tokens from its previous block x^{l-1} and a set of block cross tokens $(\bar{x}^{l-2}, \bar{x}^{l-3}...\bar{x}^1)$ from earlier blocks in the stage. *The CMHA takes both x^{l-1} and $(\bar{x}^{l-2}, \bar{x}^{l-3}...\bar{x}^1)$ as input, but only generates n tokens $\tilde{y}^k$ as its output.*

It is worth noting that the cross tokens are scaled by learned calibration coefficients (learnable scale factors LSFs) α before they are used in the CMHA. We found that the statistics of tokens from different semantic levels are very different. Without this calibration operation, cross tokens are hard to integrate to regular tokens in current blocks to work as we expected. Details are shown and analyzed in our ablation study part.

Specifically, given the inputs above, the query, key and value for multi-head attention modules are constructed as:

$$Q = W^Q x^{l-1} \quad (1)$$

$$K = W^K \left[x^{l-1}, (\mathbf{1}\alpha^l)^T \otimes (\bar{x}^{l-2}, \cdots, \bar{x}^1)\right] \quad (2)$$

$$V = W^V \left[x^{l-1}, (\mathbf{1}\alpha^l)^T \otimes (\bar{x}^{l-2}, \cdots, \bar{x}^1)\right] \quad (3)$$

where W^K, W^V, W^Q are the three weight matrices transforming input into key, value and query tokens respectively. Elements in $\alpha \in \mathbb{R}^{(l-2)\bar{n}\times 1}$ are the learnable scale factors for cross tokens and $\otimes$ denotes the element-wise multiplication where each token in $\bar{x}$ is multiplied by a corresponding scaling scalar in α. $\bar{n}$ represents the number of cross tokens for each layer.

The computed Q, K, V are then connected to the standard dot product attention operators as in the original transformer [34], which computes the globally mixed intermediate tokens y^l.

$$y^l = x^{l-1} + W^P \left[\text{softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right) V\right] \quad (4)$$

The B in Eqn.4 is a learnable matrix consisting of two parts, which are relative position bias for x and relative depth bias for $\bar{x}$. The query has only n tokens, so the output y^l is of size $n \times d$.

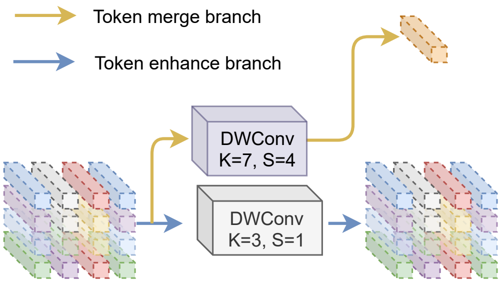

Figure 2. The detailed structure of the token merge and enhancement (TME) module.

TME. The intermediate tokens y^l are then passed to the added token merge and enhancement (TME) part, which computes the cross tokens $\bar{x}^l$ and locally enhanced tokens z^l using two separate depthwise convolutions. As shown in Fig 2, our TME has two branches. In the token merge branch, we use a large kernel (7×7) large stride $(s = 4)$ depthwise convolution to generate $\bar{x}^l$, resulting in a small number of cross tokens to be used by subsequent blocks. In token enhancement branch, a standard depth wise convolution $(3 \times 3, s = 1)$ is used to locally mix the tokens so as to enhance the 2d spatial relations in the token sequence.

FFN. The locally enhanced tokens z^l are then used in the standard FFN part of ViT to compute the output token x^l, which together with $\bar{x}^l$ are the entire output of a FcaFormer block at depth l.

In summary, our FcaFormer are different from vanilla ViT in following ways:

- **Asymmetric input and output**. In our FcaFormer, CMHA part takes both regular tokens and cross tokens as input but keep the output sequence length fixed to n as in regular transformers. This is the key point that why our proposed FcaFormer only introduce limited extra computational cost. More details are explained in section 3.3.
- **Cross token scale**. LSFs are used to facilitate the integration of cross tokens into regular tokens. Without this calibration process, cross tokens may not function as expected. As shown in Fig 5.
- **Relative depth embedding**. In the CMHA part, we choose to use relative positional encoding instead of absolute ones. Apart from its generally better performance, the choice also gives model the flexibility to encode relative depth of the cross tokens, which simplifies the model design.

3.2. FcaFormer Models

The FcaFormer block is a generic block that can be grouped as FcaFormer stages to construct models. To demonstrate the universal effectiveness of our new attention pattern, we build two types of FcaFormer based models, each falls into one of the major categories of transformer related computer vision models as described in Sec. 2. The overall structure of these models for classification task are illustrated in Fig.3.

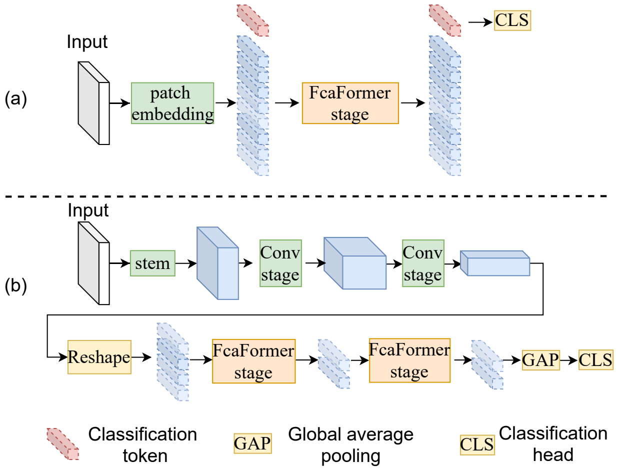

Figure 3. The overall structure of FcaFormer Mobels for image classification tasks. (a) Plain FcaFormer model. This model is directly modified from the DeiT structure. (b) Hybrid FcaFormer model. The ConvNet stages are composed of ConvNext blocks. The detailed model scaling hyperparameters are specified in Sec.3.2

Plain FcaFormer. As shown in Fig 3 (a), following the style of vanilla ViT, we construct our plain FcaFormer model (denoted as FcaFormer-L0). The model essentially uses one block of FcaFormer after the patch embedding layer that crops 16×16 patches and converts them into a token sequence. The corresponding task related head is kept unchanged compared with the original ViT model.

Hybrid FcaFormer. As illustrated in Fig 3 (b). Following the trend of combining ConvNet structures and transformer structures to build hybrid models, we also propose to build hybrid FcaFormer. Following LeViT [9] and ViT-C[39], we adopt the most straightforward way to build our Hybrid FcaFormers, where there are two conventional ConvNet stages followed by two FcaFormer stages. The ConvNet stage is composed of ConvNext blocks [21] plus a downsampling layer that uses pointwise and depthwise convolutions to reduce the feature map resolution by 4× and increases the feature map channels.

Based on the above overall structure, we build models of different sizes to compare with other works. The key scaling hyper-parameters is summarized below, where D denotes the channel size:

- FcaFormer-L0: D=192, L=12
- FcaFormer-L1: D=(64,128,192,320), L=(2,2,6,2)
- FcaFormer-L2: D=(96,192,320,480), L=(2,2,7,2)

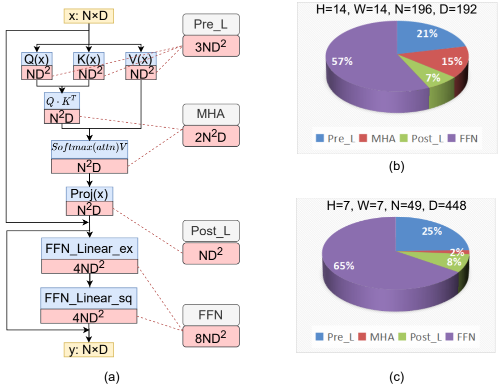

Figure 4. Computational cost analysis. (a) Detailed analysis on the number of parameters and MACs for a transformer block; (b)-(c) Computation decompositions in blocks of stage 3 of DeiT-Tiny and stage 4 of EfficientFormer-L1 respectively.

3.3. Computational Complexity of FcaFormer

In terms of computational complexity, transformers are notorious for its $\mathcal{O}(n^2)$ scaling property, which is why so many prior works try to reduce the inputs to the MHA model. However, the extra computational cost of densifying the attention pattern is rather limited in our FcaFormer models, because of the following reasons:

1) The added extra computation is restricted to the MHA part, since we keep the output sequence length of MHA fixed to n as in regular transformers. This leads to linear scaling cost in terms of the number of additional input tokens, which itself is a small ($(l-1) * n/s^2$) compared with n. The computational cost of FFN part is kept unchanged compared with original ViT blocks.

2) In vision transformer model families, the FFN part actually constitutes the majority of computations. In [19], Liu *et al.* also emphasized that larger models incur higher computational costs, primarily due to FFNs. Here, we delve into a more detailed analysis. This is paradoxical to the fact that MHA scales at $\mathcal{O}(n^2)$ while FFN scales at $\mathcal{O}(n)$, largely because that the neglected constant token size d in big O analysis is rather large and scales poorly. Fig. 4 details the exact adds and multiplications (MACs) of a standard ViT block in terms of the number of tokens n and the dimension of tokens d. Taking the transformer block DeiT-Tiny as an example, because $n = 196$ and $d = 192$ are at the same scale, the $2n^2d$ MACs in MHA is much less compared with that $(8nd^2)$ of FFN (15% vs 56% respectively). For models in pyramid shapes such as swin transformers where $n < d$, the majority skews even more towards FFN because of the local windowing effect. Compared with the reference models, our FcaFormer variants only introduce about 13% extra computational cost.

Models	# params. (M)	FLOPs (G)	Top1 acc
DeiT-T [31]	5.5	1.23	72.2
FcaFormer-L0	5.9.	1.49	74.3
Swin-1G‡	6.3	1.5	78.4
FcaFormer-L1	6.2	1.4	80.3
ConvNext-Tiny [21]	29	4.5	82.1
Swin-Tiny [20]	29	4.5	81.3
FcaFormer-L2	16.3	3.6	83.1

Table 1. Comparison with baselines. The plain version FcaFormer is based on DeiT-T. The hybrid ViT FcaFormer models are based on ConvNext and Swin Transformers. ‡means our implementation

It looks like that the proposed FcaFormer is heavier than its reference model as it introduces extra computations. However, the counter-intuitive fact is that *FcaFormer can be more light-weight than its reference model.* As is mentioned in introduction, FcaFormer reuses tokens from previous blocks, enhances interaction of tokens across blocks and improve information flow. Compared with the vanilla ViT, FcaFormer has better parameter efficiency. Therefore, FcaFormer can get comparable or even better performance with fewer layers and channels. Our experiments further verifies this advantage.

In addition to the detailed analysis above, we want to point out that the prior efforts to sparsify the attention patterns spatially in Sec.2.1 and our proposal to densify the attention patterns across semantic levels are complementary to each other. While not the key emphasis of this paper, it is definitely worth exploring if the two types of designs can be combined to yield even more efficient and performant models.

4. Experiments

In this section, we conduct image classification experiments on Imagenet-1K [3], semantic segmentation experiments on ADE20K [44], and object detection experiments on MS-COCO [18] to evaluate our proposed models. We first compare the proposed FcaFormers with our baselines and the previous SOTA methods. Then, we conduct detailed study to show the effectiveness of our design choices.

4.1. ImageNet classification

Experiment settings. The FcaFormer-L0 is implemented based on code of DeiT [1]. The hybrid FcaFormer models are implemented based on code of ConvNext [2] and Swin [3]. We follow the training recipes in DeiT [31] to train our FcaFormer-L0, except that we did not use the knowledge distillation. To train the hybrid FcaFormers, we use the same training hyper parameters and augmentations as used in ConvNext except that the batch size is restricted to 1024 and the initial learning rate is reduced to 2e-3. This change is because we don't have enough GPUs to support the default batch size of 4096 in ConvNext.

[1] https://github.com/facebookresearch/deit
[2] https://github.com/facebookresearch/ConvNeXt
[3] https://github.com/microsoft/Swin-Transformer

Models	KD	Type	Param.	MACs	Top1
DeiT-Tiny [31]	Y	ViT	6	-	74.5
DeiT-Tiny [31]	-	ViT	6	-	72.2
FcaFormer-L0	-	ViT	6	-	74.3
LeViT-128 [9]	Y	Hybrid	9	0.4	78.6
EfficientFormer-L1 [17]	Y	Hybrid	12	1.3	79.2
ParCNet [43]	-	Conv	5	1.7	78.6
TNT-Ti [12]	-	ViT	6	1.4	73.9
Swin-1G† [20]	-	ViT	7	1.0	77.3
Swin-2G† [20]	-	ViT	13	2.0	79.2
EfficientFormer-L1‡	-	Hybrid	12	1.3	76.1
MobileViT-V1 [24]	-	Hybrid	6	2.0	78.4
EdgeViT-XS [25]	-	Hybrid	7	1.1	77.5
MobileViTV2[29]	-	Hybrid	5	1.8	78.1
CoaT-Tiny [40]	-	Hybrid	6	4.4	78.3
PVT-V2-B1 [37]	-	Hybrid	13	2.1	78.7
Mobile-Former [2]	-	Hybrid	14	0.5	79.3
Edgenext [23]	-	Hybrid	6	1.3	79.4
MobileOne-S4 [33]	-	Hybrid	15	3.0	79.4
FcaFormer-L1	-	Hybrid	6	1.4	**80.3**
DeiT-S [31]	Y	ViT	22	4.6	81.2
LeViT-256 [9]	Y	Hybrid	19	1.1	81.6
EfficientFormer-L3 [31]	Y	Hybrid	31	3.9	82.4
ResNet50 [13]	-	Conv	25	4.1	78.8
ResNet50‡	-	Conv	25	4.1	79.1
PoolFormer-S24 [41]	-	Conv	21	3.4	80.3
PoolFormer-S36 [41]	-	Conv	31	5.0	81.4
ConvNext-Tiny [21]	-	Conv	29	4.5	82.1
VAN-B2 [11]	-	Conv	27	5.0	82.8
DeiT-S [31]	-	ViT	22	4.6	79.9
Swin-T [20]	-	ViT	29	4.5	81.3
T2T-ViT-14 [42]	-	ViT	22	4.8	81.5
T2T-ViT-19 [42]	-	ViT	39	8.5	81.9
MViTv2-T [16]	-	ViT	24	4.7	82.3
CSWin-T [4]	-	ViT	23	4.3	82.7
MobileViTV2 [29]	-	Hybrid	19	7.5	81.2
LITV2 [26]	-	Hybrid	28	3.7	82.0
Next-ViT-S [15]	-	Hybrid	32	5.8	82.5
FcaFormer-L2	-	Hybrid	16	3.6	**83.1**

Table 2. Comparison with the SOTA methods on ImageNet-1K validation set. KD means knowledge distillation is used. ‡ indicates implemented by us, where models are trained following the training setting used in ConvNext. Accuracy and FLOPs are calculated on input image size 224× 224. † borrowed from [2]

Comparison with baselines. The results are summarized in Table 1, from which we can see that both types of our proposed FcaFormer models outperform their reference models by a significant margin. Compared with DeiT-T, FcaFormer-L0 does have 0.4 M more parameters, but it achieves 2.1% higher accuracy. Our hybrid FcaFormers achieve better performance in all three metrics: accuracy, model size and computational cost. FcaFormer-L1 achieves 1.9% higher top-1 accuracy compared with similarly sized Swin-1G. FcaFormer-L1 also has good scalability. The scaled up version FcaFormer-L2 outperforms the ConvNext-Tiny by 1.0% with 20% less computational cost and 44% fewer parameters. FcaFormer-L2 also surpasses Swin-T by 1.8% while using fewer parameters and less computation.

Method	Backbone	mIOU	# params.	MACs
DANet	ResNet-101	45.2	69	1119
DpLab.v3+	ResNet-101	44.1	63	1021
ACNet	ResNet-101	45.9	-	-
DNL	ResNet-101	46.0	69	1249
OCRNet	ResNet-101	45.3	56	923
UperNet	ResNet-101	44.9	86	1029
UperNet	DeiT III (ViT-S)	46.8	42	588
UperNet	Swin-T	45.8	60	945
UperNet	ConvNext-T	46.7	60	939
UperNet	FcaFormer-L2	47.6	46	730

Table 3. Semantic segmentation on the ADE20K dataset. We use UperNet as our segmentation method and compare our performance using the same method with other popular backbones.

In summary, our proposed forward cross attention design universally improves performances of both plain version and hybrid ViT models without increasing too much extra computation cost or even saving parameters and FLOPs.

Comparison with other models. In Table 2, we make a comparison with other models proposed in recent two years. Compared to the latest state-of-the-arts, including ConvNets, ViTs and Hybrid models, our proposed FcaFormer models achieve the best accuracy under the condition of having similar model sizes. In addition, the proposed models beat models strengthened by knowledge distillation in classification accuracy.

Specifically, FcaFormer-L1 outperforms MobileOne and EdgeNext by 0.9% in classification accuracy, while keeping comparable or smaller model size and fewer computational cost. Among models having about 25 million parameters, FcaFormer-L2 achieves the highest classification accuracy with the fewest parameters. Compared with Next-ViT-S which achieves the second highest accuracy, FcaFormer-L2 saves about half parameters and 38% computational cost, while gaining 0.6 percentage points higher accuracy. Also, note that both EfficientFormer-L3 and LeViT-256 are trained with knowledge distillation, which has been verified to improve accuracy significantly [31]. Even so, FcaFormer-L2 still has better accuracy and smaller model size compared with these two models.

4.2. Semantic segmentation

Training setting. To evaluate the proposed FcaFormers on downstream tasks. We apply them on the ADE20K semantic segmentation task. Following Swin and ConvNext, we adopt UperNet [38] as our base framework and imple-

Backbone	#Params.	MACs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask-RCNN 3× schedule								
SWin-T	48	267	46.0	68.1	50.3	41.6	65.1	44.9
ConvNext-T	48	262	46.2	67.9	50.8	41.7	65.0	44.9
FcaFormer-L2	37	249	47.0	68.9	51.8	42.1	65.7	45.4
Cascade Mask-RCNN 3× schedule								
ResNet-50	82	739	46.3	64.3	50.5	40.1	61.7	43.4
DeiT-S	80	889	48.0	67.2	51.7	41.4	64.2	44.3
X101-32	101	819	48.1	66.5	52.4	41.6	63.9	45.2
X101-64	140	972	48.3	66.4	52.3	41.7	64.0	45.1
Swin-T	86	745	50.4	69.2	54.7	43.7	66.6	47.3
ConvNext-T	86	741	50.4	69.1	54.8	43.7	66.5	47.3
FcaFormer-L2	74	728	51.0	69.4	55.5	43.9	67.0	47.4
Swin-S	107	838	51.9	70.7	56.3	45.0	68.2	48.8
ConvNext-S	108	827	51.9	70.8	56.5	45.0	68.4	49.1
FcaFormer-L3	86	792	52.4	71.1	57.2	44.9	68.4	49.1
Swin-B	145	982	51.9	70.5	56.4	45.0	68.1	48.9
ConvNext-B	146	964	52.7	71.3	57.2	45.6	68.9	49.5
FcaFormer-L4	124	970	53.7	72.4	58.2	46.3	69.7	50.4

Table 4. Object detection on the COCO dataset.

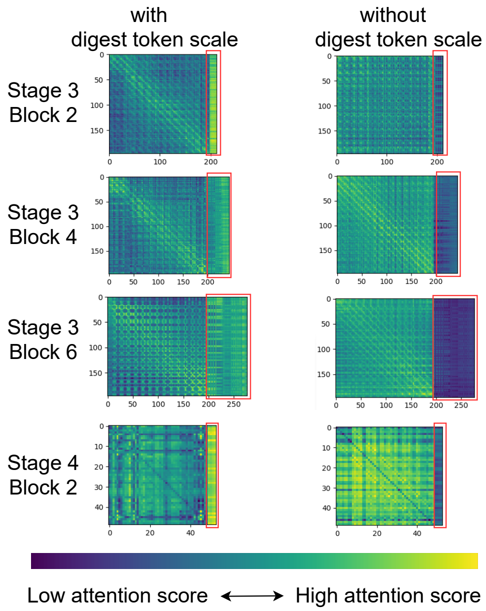

Figure 5. Effects of LSFs. Attention scores for cross tokens are marked with red boxes. From rows 1-4, attention maps are from blocks 2,4 and 6 of stage3, and block 2 of stage 4 respectively.

ment segmentation experiments on mmseg [4]. FcaFormer-L2 is trained for 160K iterations with a batch size of 16. Model pretrained on ImageNet-1K is used to initialize segmentation model. More details are presented in supplementary materials.

Comparison with baselines. The results are summarized in Tab.3, compared with Swin-Tiny, FcaFormer-L2 achieves 1.8% higher mIoU. Compared with ConvNext-Tiny, FcaFormer-L2 achieves 0.9% higher mIoU. Compared with Swin-T and ConvNext-T, the last two stages of FcaFormer-L2 are narrower (384 vs 320, 768 vs 480), which saves parameters and computational cost. Therefore, our FcaFormer-L2 achieves higher mIOU, while saving 23% parameters and 22-23% computation cost.

4.3. Object detection

Training setting. Object detection experiments are conducted on COCO 2017 and implemented on mmdet [5]. Following [20] and [21], we fintune MASK-RCNN and Cascade Mask R-CNN on the COCO dataset with FcaFormer pretrained on ImageNet-1k. We use multi-scale training, AdamW optimizer, and a 3× schedule. More details are presented in supplementary materials.

Comparison with baselines. The results are summarized in Tab.4. Experimental results on objection detection show a similar trend with that on semantic segmentation. Compared with Swin-Tiny and ConvNext-Tiny, FcaFormer-L2 achieves higher values on all six metrics, while having fewer parameters and less computational cost.

4.4. Ablation study

In this section, we conduct ablation analysis on components proposed in our FcaFormer.

Effects of the learnable scale factors (LSFs). Fig 5 shows the attention score maps of several FcaFormer blocks trained with and without LSFs. It can be seen that without LSFs, the attention between cross tokens and regular tokens are generally weak, and become weaker as the depth goes deepr (down on the figure). This verifies our hypothesis that characteristics of tokens in different levels can be very different. Comparing the two columns of attention maps, it clearly shows that the LSFs helps increasing the correlation between regular tokens and the cross tokens, thus encourages the reuse of previously generated tokens.

From baseline to FcaFormer. The high performance of FcaFormer models is based on two key factors: combining the strengths of ViTs and ConvNets, and utilizing a dense but not heavy forward cross attention design. Table 5 shows how we integrate these two key points in designing our models. To evaluate the performance of the proposed FcaFormer on real-world applications, we deployed each model on the edge device Rockchip 3288, which is widely used in various embedded applications. We collected latency and memory usage information for each model.

The first row in Table 5 provides a baseline micro model to speed up experiments. The second row uses the simplest way to build a hybrid model, which inherits some advantages of ConvNets and ViTs. Compared to the baseline, the

[4] https://github.com/open-mmlab/mmsegmentation

[5] https://github.com/open-mmlab/mmdetection

Rows	Models	Model differences	# Param. (M)	MACs (B)	Latency (ms)	Memory (M)	Top1 acc (%)
1	Swin-1G	baseline	6.25	1.49	340	46.64	78.4
2	ConvSwin	+early convolution	6.11	1.26	269	38.34	78.8 (+0.4)
3	ConvViT	+global attention	6.11	1.32	300	42.01	79.4 (+1.0)
4	FcaFormer	+naive forward cross attention.	6.11	1.37	311	42.07	79.4 (+1.0)
5	FcaFormer	+learnable scale factors.	6.11	1.37	311	42.07	79.9 (+1.5)
6	FcaFormer	+TME.	6.19	1.37	312	42.31	80.3 (+1.9)
7	FcaFormer	scale up to L2	16.3	3.6	728	95	83.1

Table 5. Ablation study on Imagenet-1K. Steps 1-7 depict the process we followed to develop our FcaFormer models from the Swin-1G. We evaluated the latency and memory usage of the models on ARM Quad Core Cortex-A17. To conduct our experiments, we utilized the RK3288 platform, which is commonly employed in real-world applications such as smart TV and AI entrance guard systems.

Models	# params.	MACs	Latency	Memory	Acc
	(M)	(B)	ARM(ms)	(M)	(%)
ConvNext-Tiny	29	4.5	875	129	82.1
ConvNext-Small	50	8.7	1618	211	83.1
ConvNext-Base	89	15.4	2708	364	83.8
ConvNext-Large	198	34.4	5604	764	84.3
Swin-Tiny	29	4.5	855	139	81.3
Swin-Small	50	8.7	1576	222	83.0
Swin-Base	88	15.4	2624	378	83.5
FcaFormer-L1(Micro)	6.2	1.4	312	42	80.3
FcaFormer-L2(Tiny)	16	3.6	728	95	83.1
FcaFormer-L3(Small)	28	6.7	1344	148	84.2
FcaFormer-L4(Base)	66	14.5	2624	328	84.9

Table 6. Batch size=1, image size=224, four threads. **ARM:**Quad Core Cortex-A17.

early convolution structure achieves higher accuracy while requiring less computation and memory usage. In the third row, we replaced window attention with global attention based on computational complexity analysis in section 3.3. This improved the accuracy by 0.6% while introducing very limited extra cost. However, latency increased by 11%, and memory usage increased by 7.7 M.

Rows 4 and 5 attempt to introduce forward cross attention. In row 4, cross tokens are not calibrated, which only introduces extra cost and has no benefit to accuracy. In row 5, we adopted LSFs to adjust cross tokens, which further improved the accuracy by 0.5%. So far, FcaFormer has outperformed the baseline by 1.5% while utilizing fewer parameters and incurring lower computational costs compared to the baseline.

In row 6, we introduced the TME module to generate representative cross tokens and enhance regular tokens, which improved the final accuracy to 80.3%, 1.9% higher than the baseline. Finally, we scaled up the micro size model and got FcaFormer-L2. This new model achieved top-1 accuracy of 83.1%, meeting the requirements for some practical applications with low latency, taking less than 1 second and minimal memory usage, below 100M.

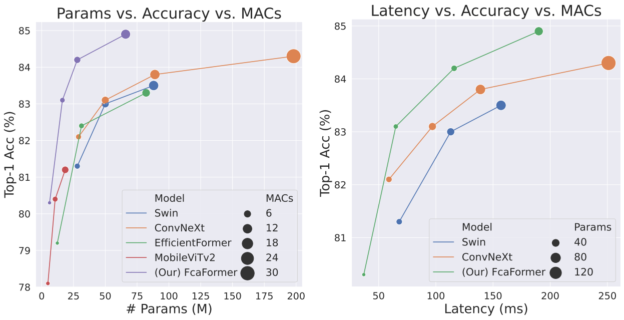

Figure 6. Comparison experiments. Left: Params vs. acc vs. MACs. Right: Latency vs. acc vs. MACs. The latency is measured on a single **NVIDIA RTX 3090** GPU with batchsize=64.

4.5. Test on different devices

We deployed our proposed models on two different devices, the widely used edge device RK3288 and GPU device RTX3090, to test their inference efficiency. We also built larger models, FcaFormer-L3 (D=(96,192,320,512), L=(3,6,12,3)), and FcaFormer-L4 (D=(128,256,512,768), L=(3,6,12,3)), to validate the scalability of FcaFormer. We repeated two sets of experiments for 100 and 1000 times, respectively. The average cost is listed in Table 6 and shown in Fig 6. Our models achieved higher accuracy compared with ConvNexts while having far fewer parameters, less memory usage, and lower latency. Furthermore, our network demonstrated good scalability. From tiny model to base model, FcaFormers consistently maintained a clear advantage compared to both Swin and ConvNext models.

5. Discussions

In this paper, we introduce a new type of attention pattern for hybrid vision models. This attention pattern leverages previously generated tokens to create forward cross-attentions that span different semantic levels. Our experiments show that this approach is effective across different model scales and vision tasks. For future work, we propose combining this design with prior research on sparsifying spatial attention patterns, which could lead to even more efficient model backbones for a range of applications

References

[1] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3

[2] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022. 6

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[4] 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. 6

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[6] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021. 2

[7] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei, and Sommerlade1 Hamed Pirsiavash2 Juergen Gall. Adaptive token sampling for efficient vision transformers. ECCV, 2022. 2

[8] Mozhdeh Gheini, Xiang Ren, and Jonathan May. On the strengths of cross-attention in pretrained transformers for machine translation. 2021. 3

[9] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021. 2, 4, 6

[10] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 2

[11] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022. 6

[12] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[15] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 2, 6

[16] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 6

[17] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022. 2, 6

[18] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick. Microsoft coco: Common objects in context. *Springer International Publishing*, 2014. 5

[19] Jing Liu, Zizheng Pan, Haoyu He, Jianfei Cai, and Bohan Zhuang. Ecoformer: Energy-saving attention with linear complexity. *Advances in Neural Information Processing Systems*, 35:10295–10308, 2022. 5

[20] 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. 1, 2, 5, 6, 7

[21] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 4, 5, 6, 7

[22] Haoyan Ma, Xiang Li, Xia Yuan, and Chunxia Zhao. Denseformer: A dense transformer framework for person re-identification. *IET Computer Vision*, 2022. 3

[23] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 3–20. Springer, 2023. 6

[24] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 2, 6

[25] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel,*

October 23–27, 2022, Proceedings, Part XI, pages 294–311. Springer, 2022. 6

[26] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. *arXiv preprint arXiv:2205.13213*, 2022. 2, 6

[27] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2035–2043, 2022. 2

[28] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2, 6

[30] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021. 2

[31] 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, 5, 6

[32] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022. 1, 2

[33] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 6

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[35] Robert J Wang, Xiang Li, and Charles X Ling. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018. 3

[36] 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

[37] 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, 6

[38] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 6

[39] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021. 2, 4

[40] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021. 6

[41] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 6

[42] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 6

[43] Haokui Zhang, Wenze Hu, and Xiaoyu Wang. Parc-net: Position aware circular convolution with merits from convnets and transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 613–630. Springer, 2022. 6

[44] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5