A Close Look at Spatial Modeling: From Attention to Convolution

Xu Ma[†], Huan Wang[†], Can Qin[†], Kunpeng Li[‡], Xingchen Zhao[†], Jie Fu[♯], Yun Fu[†]
[†]Northeastern University [‡]Meta Reality Labs [♯]Mila

Abstract

Vision Transformers have shown great promise recently for many vision tasks due to the insightful architecture design and attention mechanism. By revisiting the self-attention responses in Transformers, we empirically observe two interesting issues. First, Vision Transformers present a query-irrelevant behavior at deep layers, where the attention maps exhibit nearly consistent contexts in global scope, regardless of the query patch position (also head-irrelevant). Second, the attention maps are intrinsically sparse, few tokens dominate the attention weights; introducing the knowledge from ConvNets would largely smooth the attention and enhance the performance. Motivated by above observations, we generalize self-attention formulation to abstract a query-irrelevant global context directly and further integrate the global context into convolutions. The resulting model, a Fully Convolutional Vision Transformer (i.e., FCViT), purely consists of convolutional layers and firmly inherits the merits of both attention mechanism and convolutions, including dynamic property, weight sharing, and short- and long-range feature modeling, etc. *Experimental results demonstrate the effectiveness of FCViT. With less than 14M parameters, our FCViT-S12 outperforms related work ResT-Lite by 3.7% top-1 accuracy on ImageNet-1K. When scaling FCViT to larger models, we still perform better than previous state-of-the-art ConvNeXt with even fewer parameters. FCViT-based models also demonstrate promising transferability to downstream tasks, like object detection, instance segmentation, and semantic segmentation. Codes and models are made available at: https://github.com/ma-xu/FCViT.*

1. Introduction

In the past few years, Vision Transformers [10, 36] has dominated various visual tasks in the computer vision community. Although the costs (*i.e.*, parameters, and computations) are generally high, Vision Transformers are more likely to model better spatial relations and scale better with large models and datasets compared with the conventional convolutions [23]. A common belief is that these gratifying virtues are credited to the self-attention mechanism. Nevertheless, this plausible conjecture has been challenged recently. Studies [8, 28, 42] show that a ConvNet trained with strong training recipes can also achieve competitive or even higher performance, indicating that *a deep investigation of the spatial modeling methods is worth further exploring*.

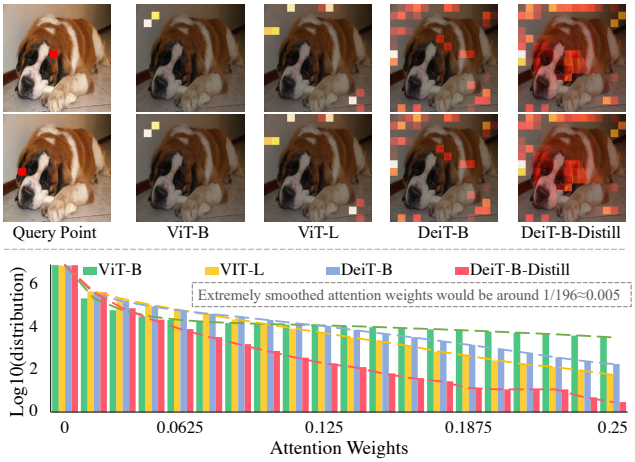


Figure 1. Illustration of sparse and query-irrelevant issues in deep attention layers. We examine several ViT variants (patch size = 16 and image patches number =196 for all). **Top:** examples of two different query points and related attention maps. **Bottom:** statistical analysis of the distribution of attention weights over ImageNet-1K validation set, normalized by Log10. If most patches contribute to attention, the attention weights would be concentrated at a very small value (∼0.005) and marginally large values; otherwise, few patches dominate the attention. **Observations:** 1) ViT variants demonstrated sparse attention, while knowledge from convolution (*e.g.*, DeiT-B-Distill) can largely smooth the attention weights, supported by both histogram and examples; 2) ViT variants exhibited a query-irrelevant (also head-irrelevant) behavior, supported by top examples. **Solution:** we address above issues by directly extracting a global context and introduce it into convolution, refer to §3.

To verify what has been learned by self-attention in Vision Transformers, we take a close look at the attention maps of the deep layers in several representative Vision Transformers, as shown in Fig. 2. Astonishingly, all of these variants present a query-irrelevant behavior that reveals nearly consistent contexts in the global scope. This observation is a departure from the design philosophy of self-attention mechanism,

Figure 2. Attention map visualizations of Vision Transformers [10, 36]. For each pair, we show the *query point* and its corresponding *attention map* (of last block and last head). All pre-trained models are downloaded from TIMM [41]. The right color bar identifies the value of normalized attention maps. Surprisingly, **the attention maps are almost the same, regardless of the query points**. Moreover, **the attention maps are intrinsically sparse if no knowledge from ConvNets is introduced**. Results from different models indicate that **these phenomena are common in Vision Transformers**. See supplementary for more examples and § 1 for a more detailed analysis.

indicating that a global context may be concealed behind the attention mechanism. Meanwhile, we notice that the attention weight are considerably sparse, as shown in ViT and DeiT-B. By distilling the knowledge from ConvNets to Vision Transformer like DeiT-B-Distill, the attention map gets considerably smoother and concentrates more on objects. This phenomenon suggests that combining convolution and self-attention may lead to gratifying results. Notice that the above two observations are not just limited to particular images, a statistical analysis on ImageNet-1K dataset and fair comparisons shown in Fig. 1 also confirm the pervasiveness.

Motivated by aforementioned findings, we propose **F**ully **C**onvolutional **Vi**sion **T**ransformer (*i.e.*, FCViT) in this paper. Starting from the observation in Fig. 2, we progressively loosen the formulation of self-attention and abstract a global context that describes the global-range visual concepts. The global context is further dynamically introduced to local convolutional operations, making the efficient fusion of short- and long-range dependencies feasible. Intrinsically, FCViT is a pure ConvNet but a Transformer-alike model that inherits the advantages of both Vision Transformer and ConvNet. Extensive experiments show that our FCViT consistently and significantly outperforms other methods with comparable costs. Remarkably, our FCViT achieves 80.9% top-1 accuracy on ImageNet-1K [7] using only 14M parameters, outperforming ResT-Lite [51] and PoolFormer-S12 by **3.7%**. Compared with state-of-the-art models like ConvNeXt [28], we still present better results with even fewer parameters. FCViT also exhibits an excellent generalization ability. When transferred to downstream vision tasks like object detection and semantic segmentation, our FCViT consistently exhibits promising performance.

2. Related Work

Vision Transformers Dominate. Originating from natural language processing, Transformer [38] targets to dynamically build mutual relationships for each token pair in a global scope. Motivated by the successes in language, tentative efforts have been made toward migrating Transformer to the vision community. The pioneering work ViT [10] has emerged as a promising approach for directly processing images using Transformer. Given an input image, ViT first tokenizes the input to non-overlapped patches and extracts token features via a stack of isotropic Transformer blocks. With an adequate training scheme, DeiT [36] circumvents the problem of requiring large datasets. Since then, various Vision Transformer variants have been springing up [27, 49]. Besides the aforementioned works, recent efforts [24, 39, 45] mainly leverage the inductive biases from ConvNets to improve the performance. In this work, we push this trend further by connecting convolution and attention in FCViT. FCViT is a departure from standard Transformers, considering the fully convolutional operations; nevertheless, it unambiguously inherits the framework of the transformer architecture [48] and enjoys the global receptive field.

ConvNets Strike Back. The necessity of multi-head self-attention in Transformers has been challenged, from language [9, 21] to vision community [33, 48, 50]. To put it another way, these critical investigations motivate lots of researchers to gush into the revitalization of ConvNets. A systematic study is ConvNeXt [28], which reexamines the design spaces of a ConvNet by gradually modifying ResNet [18] toward the standard vision Transformer [10] (*i.e.*, ViT). Astonishingly, ConvNeXt demonstrates that the resulting ConvNets compete favorably with Transformers in terms of both accuracy and scalability. Similar phenomena can also be observed in other ConvNets, like [1, 8]. Redesigning the architectures *a la* the design philosophy of Transformers, ConvNets once again showed dominance in various tasks. This paper pushes the envelope further by inherently integrating virtues of ViTs with a fully convolutional network, which is where FCViT singularity lies.

Convolution Meets Attention. Another line of work on the topic of visual backbones learns to marriage the merits of both Transformers and ConvNets. That is, considering self-attention and convolution simultaneously in one token-mixer (mixing the spatial information). Dai *et al.* [6] find that depthwise convolution and self-attention can be naturally unified in a token-mixer module. By doing so, CoAtNet [6] effectively achieve better generalization and capacity. Similarly, CvT [44] introduces convolutional token embedding and convolutional projection into Vision Transformers and empirically presents better performances. Works in [3, 12] bridge convolution module and attention module in a hybrid fashion. *Different from the aforementioned methods that connect convolution and self-attention in an explicit manner, without intrinsic dedicated designs involved, our FCViT subtly bridges global context and local structure in one unified operation.* Besides, tentative efforts have been made toward exploring the relationships between convolution and attention. In this vein, our analysis is similar to those taken by [5, 16] and allows us to further improve the performance.

3. From Attention to Convolution

By revisiting self-attention and convolution, we first bridge self-attention and convolution in one unified operation. A close look at the attention maps in ViTs also demonstrates the necessity of integrating convolution and attention. Motivated by this, a simple yet effective Transformer-alike architecture is proposed to verify our findings. Fig. 3 intuitively shows one building block of FCViT.

3.1. A Close Look at Self-Attention and Convolution

3.1.1 Self-Attention and Convolution

As the main contribution in Transformers, self-attention effectively captures the long-distance dependencies and dynamically aggregates input features according to the query patch. Formally, given an input feature map $\mathbf{X} \in \mathbb{R}^{d\times n}$, where d is the embedding dimension, and n indicates the patch number, the self-attention mechanism adaptively aggregates global information for each patch by

$$\begin{aligned} y_i &= \sum_{j=1}^{n} w\left(q_i, k_j\right) v_j, \\ \text{s.t., } w\left(q_i, k_j\right) &= \operatorname{softmax}\left(q_i^\top k_j\right) = \frac{\exp\left(q_i^\top k_j\right)}{\sum_{l=1}^{n} \exp\left(q_i^\top k_l\right)}, \end{aligned} \tag{1}$$

where $q_i = \mathbf{W}_q x_i$, $k_i = \mathbf{W}_k x_i$, and $v_i = \mathbf{W}_v x_i$ are different embeddings from $\mathbf{X}$; i, j, and l index a patch. For brevity, we ignore the positional encoding and dimensional scalar $\sqrt{d}$ in Eq. 1.

Differently, the convolution formula can be written as:

$$y_i = \sum_{j\in\Omega} w_j x_j, \tag{2}$$

where Ω indicates a local receptive field (*e.g.*, a 3×3 kernel).

Comparing the formulations, two distinct differences between attention and convolution would be:

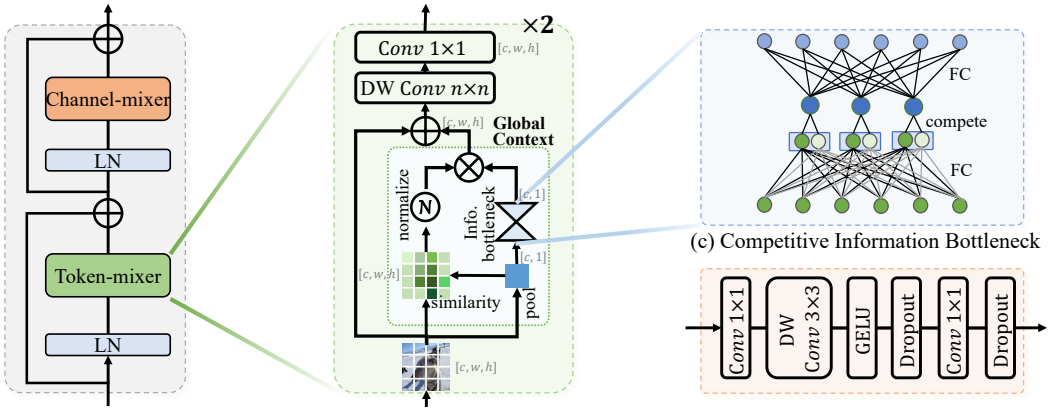
• *The dynamic of weights aggregation:* self-attention dynamically aggregates the values v_j using $w\left(q_i, k_j\right)$, which is based on the input features x_i. On the contrary, convolution utilizes fixed weights w_j to fuse features, and the weights are shared across patches.

• *The size of receptive field:* self-attention can model long-distance dependencies among patches directly, resulting in a global receptive field. Limited by the computations, convolution performs aggregation functions in a local region, like 3×3 and 7×7. A sliding window pattern (with a small step size) guarantees the information flow among regions. By stacking multiple layers, convolution can progressively enlarge the receptive field.

3.1.2 Attention Map Visualization

Some conventional wisdom leans to credit the success of Vision Transformers to the attention mechanism, considering the two differences discussed above. We acknowledge the advantages of a dynamic scheme and large receptive field; however, a close look into the details is still worth further exploring. To this end, we visualize the attention maps of various Vision Transformers, including ViT-B, ViT-L, DeiT-B, and DeiT-B-Distill. Results in Fig. 2 present some interesting phenomena:

- **Observation 1:** the attention maps consistently show a query-irrelevant (and even head-irrelevant) behavior. Visually, the attention maps $w\left(q_i, k_j\right)$ appear to be nearly identical for each testing model and image, regardless of the query patch q_i. This is a departure from the design philosophy of self-attention that each patch should exhibit a distinct attention map.



(a) FCViT Block (b) FCViT Token-mixer (c) Competitive Information Bottleneck (d) FCViT Channel-mixer

Figure 3. Illustration of an FCViT block. FCViT considers the block as a combination of token-mixer and channel-mixer. In the token-mixer, we dynamically integrate the global context with input tokens by the token-global similarity. A depth-wise convolution is employed to fuse local information. To improve the generalization of the global context, we introduce a competition-driven information bottleneck structure. Overall FCViT configurations are presented in the supplementary.

- **Observation 2:** the attention weights (see ViT-B, ViT-L, and DeiT-B) are relatively sparse, indicating that only several patches dominate the attention. By introducing the knowledge from convolution, the attention weights (see DeiT-B-Distill) are largely smoothed, and the performance is significantly improved as well (83.4% of DeiT-B-Distill *vs.* 81.8% of DeiT-B top-1 accuracy on ImageNet-1K validation set).

The aforementioned two observations suggest that: **1)** a query-irrelevant global context may be sufficient to work well for vision tasks; **2)** combining the knowledge from both self-attention and convolution can yield gratifying results.

3.1.3 From Self-Attention to Convolution

Inspired by the aforementioned observations, we revisit self-attention and connect it to convolution. Given the Eq. 1, we first remove the querying patches as implied by observation 1. The resulting simplified attention can be written as

$$y = \sum_{j=1}^{n} w\left(k_j\right) v_j = \sum_{j=1}^{n} \frac{\exp\left(k_j\right)}{\sum_{l=1}^{n} \exp\left(k_l\right)} v_j \\ = \mathcal{N}(\mathbf{K})\mathbf{V} = gc, \tag{3}$$

where $\mathcal{N}(\cdot)$ indicates a normalization function, like softmax. Since Eq. 3 is purely based on the global input features $\mathbf{X}$ and models the global information, we denote it as global context (*i.e.*, $gc \in \mathbb{R}^d$) for simplicity. Note that the objective of normalization function $\mathcal{N}(\cdot)$ is to generate a weight for v_i. Hence, it can be generalized to other forms besides softmax, like average or sigmoid function.

Furthermore, as indicated by observation 2, we are encouraged to combine the attention mechanism and convolution into a unified formulation. A simple implementation that bridges Eq. 3 and Eq. 2 is

$$y_i = \sum_{j \in \Omega} w_j (x_j + gc). \tag{4}$$

Conceptually, Eq. 4 introduces a global context into each patch and leverages a convolution to aggregate local information consequently. With Eq. 4, we efficiently introduce the global information into the local patches, avoiding the combination of two individual operations. Meanwhile, by employing the global context implemented in Eq 3, we significantly reduce the computational complexity of self-attention. Also, Eq. 4 can be easily implemented by a convolution layer, making our model concise yet effective. Next, we instantiate our FCViT based on the analysis above.

3.2. Fully Convolutional Vision Transformer

3.2.1 General Framework

Our FCViT follows the framework of MetaFormer [48], which adopts hierarchical architecture with 4 stages that progressively reduce the spatial size. Given an input image, we first utilize overlapping patch embedding to tokenize images by linear mapping. In each stage, a series of isotropic FCViT blocks are utilized to extract features. FCViT block involves two independent modules, the token-mixer and channel-mixer (as well as residual connections and Layer Normalization), as shown in Fig. 3. In the end, a classifier is employed to generate the classification logits. Varying the number of blocks and the channel number, we instantiate FCViT by FCViT-Tiny, FCViT-B12, and FCViT-B24, *etc*. Detailed configurations can be found in the supplementary. Next, we describe the detailed designs of our FCViT.

3.2.2 Enhanced Global Context in Token-Mixer

Following Eq. 4, our token-mixer can be implemented by

$$\mathbf{Y} = \text{conv}_1\left(\text{conv}_k\left(\mathbf{X} + gc\right)\right) = \text{conv}\left(\mathbf{X} + gc\right), \quad (5)$$

where conv_1 is a point-wise convolution and conv_k is a depth-wise convolution with kernel size of k. For convenience, we use conv to denote the combination of the two convolutions. We repeat the operation in Eq. 5 twice to achieve the best model size and accuracy trade-off. Specifically, we normalize the global context by average-pooling instead of softmax for simplicity. That is:

$$gc = \sum_{i=1}^{n} \frac{\mathbf{W}_v x_i}{n} = \mathbf{W}_v \sum_{i=1}^{n} \frac{x_i}{n}. \quad (6)$$

Dynamic Global Context. Directly fusing the global context $gc \in \mathbb{R}^d$ with input feature $\mathbf{X} \in \mathbb{R}^{d \times w \times h}$ may lead to limited improvements since gc is broadcasted equally for each patch in $\mathbf{X}$. In other words, Eq. 5 can be rewritten as

$$\begin{aligned} \mathbf{Y} &= \text{conv}\left(\mathbf{X} + \mathbf{W}_v \sum_{i=1}^{n} \frac{x_i}{n}\right) \\ &= \text{conv}\left(\mathbf{X}\right) + \mathbf{W}_{conv}^{\top} \mathbf{W}_v \sum_{i=1}^{n} \frac{x_i}{n}, \end{aligned} \quad (7)$$

where $\mathbf{W}_{conv}^{\top}$ is the weights in the convolution layer. *Eq. 7 explicitly demonstrates that directly fusing gc with $\mathbf{X}$ is equal to fusing gc outsides the convolution.* Hence, we expect a dynamic fusion scheme that fuses the global context with bias based on the input $\mathbf{X}$.

We circumvent this problem by promoting token-global similarity. For clarity, we denote $\sum_{i=1}^{n} \frac{x_i}{n}$ as $\overline{\mathbf{X}}$. The similarity score $\mathbf{S} \in \mathbb{R}^{w \times h}$ is given by $\mathbf{S} = \mathbf{X}\overline{\mathbf{X}}$, which calculates the relations between each patch x_i and the global average pooling $\overline{\mathbf{X}}$. We re-scale the similarity and update gc by

$$gc' = \left(\alpha \frac{\mathbf{S} - \mu_{\mathbf{S}}}{\sigma_{\mathbf{S}} + \epsilon} + \beta\right) gc = \mathbf{S}' gc, \quad (8)$$

where α and β are learnable scalars; $\mu_{\mathbf{S}}$ and $\sigma_{\mathbf{S}}$ are mean and standard deviation of $\mathbf{S}$; $\epsilon = 1e^{-5}$ is for numerical stability. We use $\mathbf{S}'$ and gc' to present normalized $\mathbf{S}$ and updated gc, respectively. By doing so, each patch dynamically integrates the global context according to the similarity.

Multi-Group Similarity. Inspired by multi-head attention, we further extend our token-global similarity to a multi-group fashion in pursuit of better diversity. We first divide gc, $\mathbf{X}$, and $\overline{\mathbf{X}}$ into g groups along the channel dimension, respectively. We then perform the operation of Eq. 8 for each group and merge the outputs via concatenation. That is, $gc' = \text{concat}\left([\mathbf{S}'_1 gc_1, \mathbf{S}'_2 gc_2, \cdots, \mathbf{S}'_g gc_g]\right)$. Different from the multi-head attention that maps all channels into multiple heads, we leverage the group operation to reduce computational overheads. Ablation study in Fig. 4 visually shows the differences between our multi-group similarity and multi-head attention.

Competitive Information Bottleneck. We next introduce a competitive information bottleneck to further improve the generalization ability of gc and better describe the global context. We first squeeze $\overline{\mathbf{X}}$ by $\mathbf{W}_{s1} \in \mathbb{R}^{\frac{d}{r} \times d}$ and $\mathbf{W}_{s2} \in \mathbb{R}^{\frac{d}{r} \times d}$, respectively. Then, the two squeezed vectors compete to generate a bottleneck representation by maxout [11]. Lastly, we recover the representation to original dimension by $\mathbf{W}_r \in \mathbb{R}^{d \times \frac{d}{r}}$. Our competitive information bottleneck can be written as $gc = \mathbf{W}_r \text{maxout}\left(\mathbf{W}_{s1}\overline{\mathbf{X}}, \mathbf{W}_{s2}\overline{\mathbf{X}}\right)$. By default, we set r to 8. Then, the multi-group dynamic gc can be applied consequently. While the design is similar to SENet [20], we are in pursuit of competitive information and parameters reduction; no attention design is involved. In spite of the fact that it is not essential to our FCViT, the competitive information bottleneck can significantly reduce parameters and consistently boost performance.

3.2.3 Conducive Implementation Details

While the token-mixer is the core of FCViT, some detailed designs are also instrumental. Different from the non-overlapping tokenization in ViT [10] and ConvNeXt [28], we consider the overlapped patch embedding as presented in [40], where each patch has small overlapping to ease the communication among patches. Also, a depth-wise convolution is introduced between the two point-wise convolution layers, as shown in Fig. 3.

4. Experiments

We validate FCViT on ImageNet-1K [7], MS COCO [26], and ADE20K [52] datasets. We first demonstrate the effectiveness of FCViT on ImageNet-1K classification task. Extensive ablation studies provide a close look at the internal operations. Then, we transfer pre-trained models to object detection, instance segmentation, and semantic segmentation tasks to examine the generalization ability of FCViT.

4.1. Image Classification on ImageNet-1K

Experimental Settings. We train FCViT models on the ImageNet-1K training set (with around 1.3M images) and evaluate upon the validation set. We follow the common training recipe in [6, 36, 41, 48]. All our models are trained for 310 epochs using AdamW [30] with a momentum of 0.9 and a weight decay of 0.05. The learning rate is initialized to 0.001 and adjusted by cosine schedular [29]. By default, the models are trained on 8 A100 GPUs with a mini-batch size

Method	Param. (M)	FLOPs (G)	Top-1 (%)	Speed (im/s)
PVTv2-B0 [40]	3.4	0.6	70.5	-
T2T-ViT-7 [49]	4.3	1.1	71.7	-
DeiT-Tiny/16 [36]	5.7	1.3	72.2	767.07
TNT-Ti [15]	6.1	1.4	73.9	-
FCViT-Tiny (ours)	4.6	0.8	**74.9**	759.79
PVT-Tiny [39]	13.2	1.9	75.1	-
ResT-Lite [51]	10.49	1.4	77.2	-
SOFT-Tiny [31]	13.0	1.9	79.3	-
Pool-S12 [48]	11.9	2.0	77.2	764.03
ResNet18 [18]	12	1.8	69.8	926.73
FCViT-B12 (ours)	14	2.5	**80.9**	771.56
DeiT-Small/16 [36]	22.1	4.6	79.8	762.19
PVT-Small [39]	24.5	3.8	79.8	724.52
Swin-T [27]	28.3	4.5	81.3	758.84
ResT-Base [51]	30.28	4.3	81.6	-
ResMLP-24 [35]	30.0	6.0	79.4	756.88
AS-MLP-T [25]	28	4.4	81.3	-
Pool-S24 [48]	21.4	3.6	80.3	763.77
CoAtNet-0 [6]	25	4.2	81.6	-
CvT-13 [44]	20	4.5	81.6	-
Conformer-Ti [32]	23.5	5.2	81.3	-
ResNet50 [18]	26	4.1	80.4	770.34
ConvNeXt-T [28]	28.6	4.5	82.1	747.27
PatchConv-S60 [37]	25.2	4.0	82.1	-
Focal-T(SRF) [45]	28.4	4.4	82.1	708.46
Focal-T(LRF) [45]	28.6	4.5	82.3	725.04
FCViT-B24 (ours)	25.7	4.7	**82.5**	706.97

Table 1. Comparison with SOTA backbones on ImageNet-1K benchmark. Throughput (images / s) is measured on a single A100 GPU with a batch size of 128. All models are trained and evaluated on 224×224 resolution. We use dots with different colors to present different types of token-mixer, attention-based, convolution-based, MLP-based, Att&Conv.-based, and other token-mixers. The best results are marked in **bold**. For more results of larger models, please see the supplementary.

of 128 (1024 in total). Similar to previous works [13, 36], we employ Exponential Moving Average (EMA) to improve the training. Results are reported in Table 1.

Performance Analysis. Empirically, our FCViT outperforms related work by a clear margin. FCViT-Tiny outperforms DeiT-Tiny by **2.7%** (74.9% *vs.* 72.2%) top-1 accuracy, using much fewer parameters and FLOPs. When increasing the model size, FCViT consistently achieves a leading performance. FCViT-B12 achieves 80.9% accuracy with 14M parameters, outperforming ResT-Lite and PoolFormer-S12 by **3.7%**. Comparing with the state-of-the-art model ConvNeXt, we still present an improvement of 0.4% (82.5% *vs.* 82.1%) accuracy, with fewer parameters but similar FLOPs.

4.2. Isotropic FCViT

We also investigate the compatibility of FCViT block with isotropic design (*e.g.*, DeiT [36], ResMLP [35]), in which no down-sampling block is introduced and the feature map resolution is constant across all depths. To build Isotropic FCViT, we "patchify" input images via a stride-p $p \times p$ convolution, with $p = 16$ by default, like ViT. Then, isotropic FCViT blocks are stacked to extract features. "256/12" indicates the input channel of each block is 256 and we use 12 FCViT blocks to build the network (same meaning for "384/16"). Isotropic FCViTs are trained with the same settings as before. From the resulting in Table 2, we observe Iso. FCViTs perform better than related isotropic architectures, indicating the effectiveness of our design.

Model	Param.(M)	FLOPs.(G)	Top-1(%)
DeiT-Ti [36]	5.7	1.3	72.2
Iso.-FCViT-256/12	8.2	1.4	**75.0**
DeiT-S [36]	22.1	4.6	79.8
ResMLP-S24 [35]	30	6.0	79.4
MLP-Mixer-B/16 [34]	59.9	12.6	76.4
ConvNeXt-S(iso.) [28]	22.0	4.3	79.7
ConvMixer-768/32	21.1	20.9	80.2
Iso.-FCViT-384/16	23.2	4.0	**80.3**

Table 2. Comparison between Isotropic FCViT and other isotropic architectures on ImageNet-1K benchmark.

4.3. Ablation Study

We next conduct extensive ablation studies to better understand and evaluate the utility of FCViT. Detailed analyses demonstrate the contribution of introducing the dynamic global context to local regions. Component ablations further disentangle the effectiveness of each module. In this subsection, we conduct all experiments based on FCViT-T, and we train all models with a mini-batch size of 256.

Size	3	5	7	8	9	11	13
Acc.	73.2	73.2	73.4	73.4	73.2	**73.5**	73.2

Kernel Size. We vary the kernel size of convolutions in the token-mixer from 3 to 13. Different from previous work [28], we did not observe a linear correlation between kernel size and performance, and the performance gap is relatively small, ranging from 73.2% to 73.5%. One possible reason would be that we explicitly introduced the global context to the local tokens, which already enlarges the receptive field to a global range. Hence, the influence of kernel size is largely diluted. By default, we set the value to 11.

Global Context. The kernel contribution of FCViT is to introduce the global context to local tokens, converting self-attention to local convolution. Here, we evaluate the necessity of our global context. We first build a plain FCViT version as our baseline, where the global context is removed. Then, we integrate the dynamic version and competitive information bottleneck progressively. Table 3 shows the

(a) FCViT-B12 token-global similarity.

(b) ViT-B query point and self-attention maps.

Figure 4. Visual comparisons of FCViT-B12 similarity and ViT-B attention map. We plot all the outputs of the last block for the two models (8 groups for FCViT and 12 heads for ViT). Compared to ViT, the results indicate that: **1)**, our FCViT focuses more on the objects; **2)** FCViT presents more diversities than multi-head attention, whose attention maps from different heads are nearly the same.

GC	Dy. GC.	Comp. Info.	params	FLOPs	top-1 (%)
✗	✗	✗	4.2M	0.8G	72.8$_{\uparrow 0.0}$
✓	✗	✗	4.4M	0.8G	73.0$_{\uparrow 0.2}$
✓	✓	✗	4.4M	0.8G	73.5$_{\uparrow 0.7}$
✓	✓	✓	4.5M	0.8G	73.7$_{\uparrow 0.9}$

Table 3. Ablation of Global Context. We individually ablate the global context, dynamic scheme, and competitive bottleneck.

effectiveness of each component. Without Global Context, a plain FCViT architecture reaches 72.8% top-1 accuracy, which is still a leading result in the first block of Table 1. By introducing the global context, we slightly improve the plain network by 0.2%. When integrating the global context as presented in Eq. 8, the performance is increased to 73.5%, indicating the effectiveness of the dynamic scheme. Remarkably, the computation and parameter overheads are negligible, leading to almost no additional inference time. When introducing the competitive information bottleneck, we further improve the performance to 73.7%, exhibiting 0.9% improvements over the plain network without global context. In the following experiments, we will consider the global context with the dynamic scheme and competitive information bottleneck by default.

Group	1	4	8	16	32
Acc.	73.2	73.4(↑)	**73.5**	73.3	73.3

Group Number for Token-Global Similarity. As introduced, we intentionally copy the design philosophy of multi-head attention to emphasize the similarity in different groups. The table on the right reveals the benefits of multi-group similarity. When the group number is increased to 8, we achieve the best result (0.3% improvements). Note that only memory operation is introduced for multi-group similarity (computational complexity and operational redundancy are almost the same), and the running speed is not increased. Empirically, we set the group number to 8 in all other experiments.

4.4. Multi-Group Token-Global Similarity *vs.* Multi-Head Self-Attention.

Similar to the multi-head self-attention mechanism, we consider the token-global similarity in a grouped-channel fashion. In Fig. 4a, we showcase the similarity scores of FCViT-B12 for all 8 groups in the last block. Similarly, we present the attention maps of ViT-B for all 12 heads in Fig. 4b. We randomly fix the query point for all validating images since Fig. 2 demonstrated that the attention map of the last block is nearly query-irrelevant.

The visualization results exhibit some interesting insights. **1)** Besides the query-irrelevance, we notice that the attention map of ViT is head-irrelevant as well, as shown in Fig. 4b. This is a departure from the design philosophy of multi-head attention mechanism, where different heads suppose to exhibit different attention. Differently, our FCViT presents diverse similarities among the groups. This phenomenon can be explained by the formulations. In multi-head self-attention, all channels are employed to map to multiple heads. Without constraints (*e.g.*, loss functions), the heads are hard to present meaningful diversities, especially when the model converges in the last layers. As a comparison, our channel-group similarity calculates the similarity for a group of channels individually, making each group self-contained and preventing the similarities collapsed. **2)** Our FCViT focuses more on the objects, while ViT cannot exhibit such a desirable property. Among all testing examples

backbone	Params	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
ResNet-18 [18]	31.2M	34.0	54.0	36.7	31.2	51.0	32.7
PoolFormer-S12 [48]	31.6M	37.3	59.0	40.1	34.6	55.8	36.9
PVT-Tiny [39]	32.9M	36.7	59.2	39.3	35.1	56.7	37.3
FCViT-B12 (ours)	34.3M	**42.3**	**64.2**	**46.2**	**38.6**	**61.1**	**41.3**
ResNet-50 [18]	44.2M	38.0	58.6	41.4	34.4	55.1	36.7
PoolFormer-S24 [48]	41.0M	40.1	62.2	43.4	37.0	59.1	39.6
PVT-Small [39]	44.1M	40.4	62.9	43.8	37.8	60.1	40.3
Swin-Tiny [14, 27]	47.8M	42.2	64.6	46.2	39.1	61.6	42.0
Swin-Tiny [27, 45]	47.8M	43.7	**66.6**	47.7	39.8	**63.3**	42.7
FCViT-B24 (ours)	43.1M	**44.1**	65.4	**48.4**	**39.9**	62.4	**42.7**

Table 4. COCO object detection and instance segmentation results using Mask-RCNN (1×).

Backbone	Params	mIoU(%)
ResNet18 [18]	15.5M	32.9
PVT-Tiny [39]	17.0M	35.7
PoolFormer-S12 [48]	15.7M	37.2
FCViT-B12 (ours)	17.8M	**43.3**
ResNet50 [18]	28.5M	36.7
PVT-Small [39]	28.2M	39.8
Poolformer-S24 [48]	23.2M	40.3
Twins-PCPVT-S [4]	28.4M	44.3
Twins-SVT-S [4]	28.3M	42.6
FCViT-B24 (ours)	25.3M	**45.5**

Table 5. Semantic segmentation performance of different backbones with Semantic FPN on the ADE20K validation set.

and groups, FCViT explicitly emphasizes the tokens within the objects. Recall that the similarity is calculated between local tokens and the global context. This result highlights the validity of our global context. Meanwhile, by dynamically fusing the global context, our FCViT further enhances the representational ability of local tokens in turn.

4.5. Object Detection and Instance Segmentation

We next probe the transferability of FCViT on downstream tasks, including object detection and instance segmentation. We conduct experiments on MS COCO 2017 benchmark [26]. We train and evaluate Mask R-CNN [17] with FCViT backbone initialized with classification pre-trained weights. For a fair comparison, we follow the settings in PVT [39] and PoolFormer [48] that adopts the 1× training schedule (*i.e.*, 12 epochs) based on the MMDetection [2] framework. Results in Table 4 demonstrate that our FCViT significantly outperforms related work by a clear margin. Particularly, our FCViT brings **5.0 points of mAP^{box} and 4.0 points of mAP^{mask}** against PoolFormer-S12 at comparable settings. Even compared with larger models (parameters in backbones are doubled) like PoolFormer-S24 and ResNet-50, our FCViT-Tiny still exhibits significantly better results, showing gratifying transferability.

4.6. Semantic Segmentation on ADE20K

We also evaluate our FCViT equipped with Semantic FPN [22] on ADE20K [52] dataset for the semantic segmentation task. ADE20K contains 150 semantic categories and consists of 20k, 2k, and 3k images for training, validation, and testing, respectively. We use the ImageNet-pretrained backbone model taken from Table 1. Following [48], we train all our models for 40k iterations with a batch size of 32. All our models are trained using AdamW optimizer with an initial learning rate of 2x10-4. We decay the learning rate by a polynomial decay schedule with a power of 0.9.

The results compared with previous work are presented in Table 5. Empirically, our FCViT achieves promising performance on the semantic segmentation task. Compared with PoolFormer-S12, FCViT-B12 improves the performance by 6.1% mIoU using a similar number of parameters. Compared with Twins-PCPVT-S, FCViT-B24 improves the performance by 1.2% mIoU using even fewer parameters. In line with the results in object detection and instance segmentation tasks, FCViT also demonstrated promising transferability for the semantic segmentation task.

5. Conclusion

In this paper, we first take a close look at two foundational token-mixers, attention and convolution. Observations of sparse and query-irrelevant attention maps motivate us to connect attention with convolution by abstracting a global context and dynamically fusing it with local tokens. The resulting model, Fully Convolutional Vision Transformer (FCViT), explicitly embraces the advantages of both convolution and attention, and exhibits promising efficiency and performance. The multi-group token-global similarity and competitive information bottleneck further boost the representational ability of our method. Experiments on multiple tasks validate the utility of FCViT and confirm our analysis. We hope that our empirical observations and FCViT design can bring new insights to the vision community.

References

[1] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *NeurIPS*, 2021. 3

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 8

[3] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *CVPR*, 2022. 3

[4] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *NeurIPS*, 2021. 8

[5] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020. 3

[6] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 2021. 3, 5, 6, 12

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 5

[8] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. *CVPR*, 2022. 1, 3

[9] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*. PMLR, 2021. 3

[10] 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. In *ICLR*, 2020. 1, 2, 3, 5

[11] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*. PMLR, 2013. 5

[12] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021. 3

[13] 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

[14] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *NeurIPS*, 2022. 8

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

[16] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *ICLR*, 2022. 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 8

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6, 8, 11, 12

[19] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *NeurIPS*, 2018. 12

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5, 11, 12

[21] Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *NAACL-HLT (1)*, pages 3543–3556, 2019. 3

[22] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 8

[23] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 1

[24] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 2

[25] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *ICLR*, 2022. 6, 12

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014. 5, 8

[27] 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 *ICCV*, 2021. 2, 6, 8, 11, 12

[28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CVPR*, 2022. 1, 2, 3, 5, 6, 11

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2017. 5

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[31] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. In *NeurIPS*, 2021. 6

[32] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, 2021. 6, 12

[33] Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. Sparse mlp for image

recognition: Is self-attention really necessary? *arXiv preprint arXiv:2109.05422*, 2021. 3

[34] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 2021. 6, 12

[35] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 6

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *ICML*, 2021. 1, 2, 5, 6

[37] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation. *arXiv preprint arXiv:2112.13692*, 2021. 6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPSs*, 2017. 2

[39] 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 *ICCV*, 2021. 2, 6, 8, 12

[40] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022. 5, 6, 12

[41] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 2, 5

[42] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 1

[43] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 12

[44] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021. 3, 6, 12

[45] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks, 2022. 2, 6, 8

[46] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *NeurIPS*, 2021. 12

[47] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S2-mlp: Spatial-shift mlp architecture for vision. In *WACV*, 2022. 12

[48] 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. *CVPR*, 2021. 2, 3, 4, 5, 6, 8, 11, 12

[49] 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 *ICCV*, 2021. 2, 6, 12

[50] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021. 3

[51] Qinglong Zhang and Yu-Bin Yang. Rest: An efficient transformer for visual recognition. *NeurIPS*, 2021. 2, 6

[52] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 5, 8

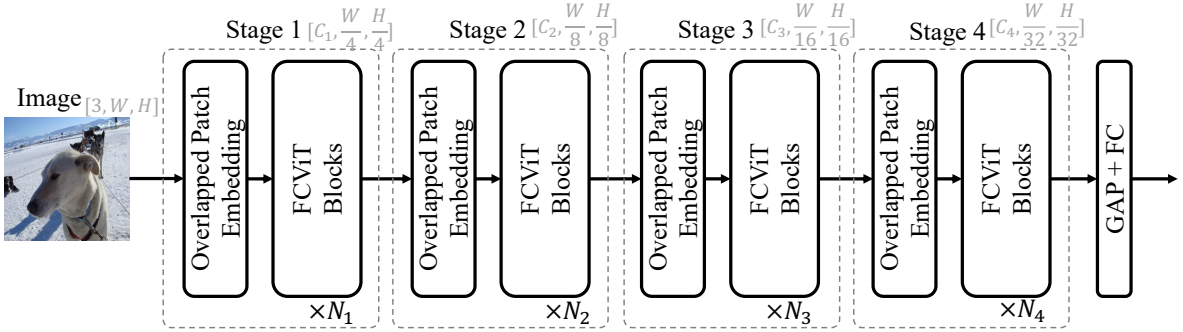


Figure 5. The overall architecture of FCViT. Given an input image, FCViT considers 4 stages to process and extract features. In each stage, one overlapped patch embedding is employed to reduce the spatial size, and a stack of FCViT blocks is employed to model feature relations.

A. FCViT Architecture & Configurations

A.1. FCViT Architecture

We first describe the high-level architecture overview of our FCViT. Following hierarchical Transformers [27] and classical ConvNets [18], our FCViT includes four stages to reduce the spatial size gradually. Specifically, we reduce the spatial size by a factor of 4 in the first stage and a factor of 2 for the rest. In the end, a classifier (which is combined by a global averaging pooling and a fully connected layer) is employed to give the classification logits.

Next, we exhaustively present the details of overlapped patch embedding. Different from the non-overlapping patch embedding used in the original ViT and ConvNeXt, our overlapped patch embedding introduces appropriate overlapping between neighbor patches. In our implementation, we achieve this by a convolution operation, where the stride size is smaller than the kernel size. For example, the kernel size is set to 7, and the step size is set to 4 in the first stage. As a result, half of the pixels in a patch are overlapped with neighbor patches. To facilitate the training, we add Layer Normalization after the convolution operation, which is consistent with previous work.

A.2. FCViT Configurations

The detailed configurations of our FCViT variants are shown in Table 6. We instantiate four variants of FCViT for different model capacities. Empirically, we set the MLP-ratio in the channel-mixer (FFN) to 8 for the first and second stages. For the blocks number, we follows the design in [28, 48]: the number is set to n, n, $3n$, and n for each stage.

B. More Experiments

We present more experiments, including large model comparisons and more ablation studies.

Larger Model. First, we introduce the performance of larger models as a supplementary. Clearly, our FCViT-B48 still dominates the classification performance with relatively few parameters for the larger models. This promising result indicates that our FCViT not only surpasses related work significantly with a small model capability, but also enjoys a good scalability.

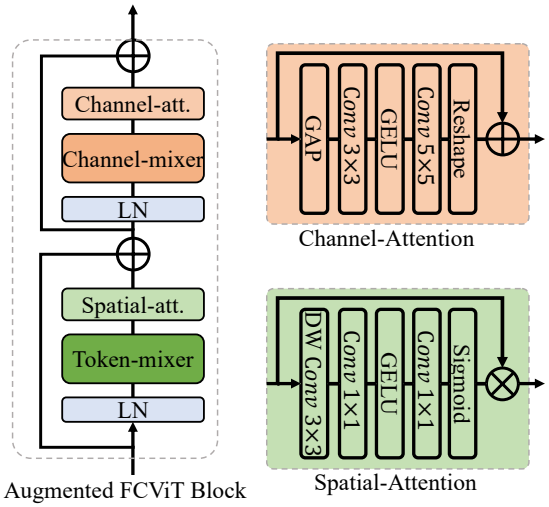


Figure 6. Attention module augmented FCViT block, and details in spatial- and channel-attention.

Integration of Attention Module. Some works introduce attention modules laterally to enhance the model with minimal additional overheads [20]. We examine the effectiveness of combining attention module with our FCViT to verify if attention modules can boost our FCViT as well.

To achieve this, we first build two fully convolutional attention modules, the spatial-attention module and the channel-attention module. Then, we plug the two modules into token-mixer and channel-mixer, respectively. Fig. 6 show the attention module augmented FCViT block. In spatial attention, we introduce two depth-wise convolutions and two point-wise convolutions to model the local spatial attention. In channel-attention, we first abstract the global feature via global average pooling, then two 1D convolutions are

Stage	Size	Layer	FCViT-Tiny	FCViT-B12	FCViT-B24	FCViT-B48
S1	56x56	Patch Embed.	$\begin{bmatrix}\text{patch\_size}=7\\ \text{stride}=4\\ \text{dim}=32\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=7\\ \text{stride}=4\\ \text{dim}=64\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=7\\ \text{stride}=4\\ \text{dim}=64\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=7\\ \text{stride}=4\\ \text{dim}=64\end{bmatrix}$
	56x56	FCViT Blocks	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=32\end{bmatrix}\times 3$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=64\end{bmatrix}\times 2$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=64\end{bmatrix}\times 4$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=64\end{bmatrix}\times 8$
S2	28x28	Patch Embed.	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=64\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=128\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=128\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=128\end{bmatrix}$
	28x28	FCViT Blocks	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=64\end{bmatrix}\times 3$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=128\end{bmatrix}\times 2$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=128\end{bmatrix}\times 4$	$\begin{bmatrix}\text{mlp\_r.}=8\\ \text{dim}=128\end{bmatrix}\times 8$
S3	14x14	Patch Embed.	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=160\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=320\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=320\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=320\end{bmatrix}$
	14x14	FCViT Blocks	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=160\end{bmatrix}\times 5$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=320\end{bmatrix}\times 6$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=320\end{bmatrix}\times 12$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=320\end{bmatrix}\times 24$
S4	7x7	Patch Embed.	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=320\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=512\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=512\end{bmatrix}$	$\begin{bmatrix}\text{patch\_size}=3\\ \text{stride}=2\\ \text{dim}=512\end{bmatrix}$
	7x7	FCViT Blocks	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=320\end{bmatrix}\times 2$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=512\end{bmatrix}\times 2$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=512\end{bmatrix}\times 4$	$\begin{bmatrix}\text{mlp\_r.}=4\\ \text{dim}=512\end{bmatrix}\times 8$

Table 6. Model configurations for our FCViT. Based on the framework of MetaFormer, we introduce four configurations FCViT-Tiny, FCViT-B12, FCViT-B24, and FCViT-B48, with different model scales and capacities.

Method	Param. (M)	FLOPs. (G)	Top-1. (%)
T2T-ViT$_t$-19 [49]	39.2	9.8	82.4
PVT-Medium [39]	44.2M	6.7	81.2
Swin-S [27]	49.6	8.7	83.0
PVTv2-B3 [40]	45.2	6.9	83.2
Focal-S [46]	51.1	9.1	83.5
T2T-ViT$_t$-24 [49]	64.0	15.0	82.3
AS-MLP-S [25]	50	8.5	83.1
Mixer-B/16 [34]	59.0	11.6	76.4
PoolFormer-M36 [48]	56	9.1	82.1
PoolFormer-M48 [48]	73	11.9	82.5
S2-MLP-deep [47]	51	10.5	80.7
CoAtNet-1 [6]	42	8.4	83.3
CvT-21 [44]	32	7.1	82.5
Conformer-S [32]	37.7	10.6	83.4
ResNet101 [18]	45	7.9	81.3
ConvNeXt-S [27]	50.1	8.7	83.1
FCViT-B48$_{(\text{ours})}$	49.1	9.2	**83.6**

Table 7. Comparison with larger SOTA backbones on ImageNet-1k benchmark. The best results are marked in **bold**.

applied along the channel dimension to module the channel dependencies. We employ the summation for the channel-attention in our implementation. Note that both our spatial- and channel-attention are based on convolutional operation, aligning with our motivation and design philosophy.

Spatial-att.	Channel-att.	top-1	top-5
✗	✗	74.9	92.6
✓	✗	74.9	92.6
✗	✓	75.0	92.6
✓	✓	75.1	92.5

Table 8. Ablation on attention modules.

Results in Tale 8 exhibit some interesting phenomena. Different from previous works [19, 20, 43], additional attention modules improve our FCViT marginally, only 0.1% to 0.2% top-1 accuracy and no improvements on top-5 accuracy. The following two aspects can explain this: 1) our FCViT explicitly inherits both global- and local-range feature modeling abilities, making additional attention modules show limited improvements; 2) the training recipe further limits the contributions of attention modules. Unlike conventional ConvNets trained with 100 epochs with simple data augmentation methods, recent works train all models by 300 epochs and with more and better training strategies. Besides our tailored convolutional attention modules, SE module can also be employed as a channel-attention module

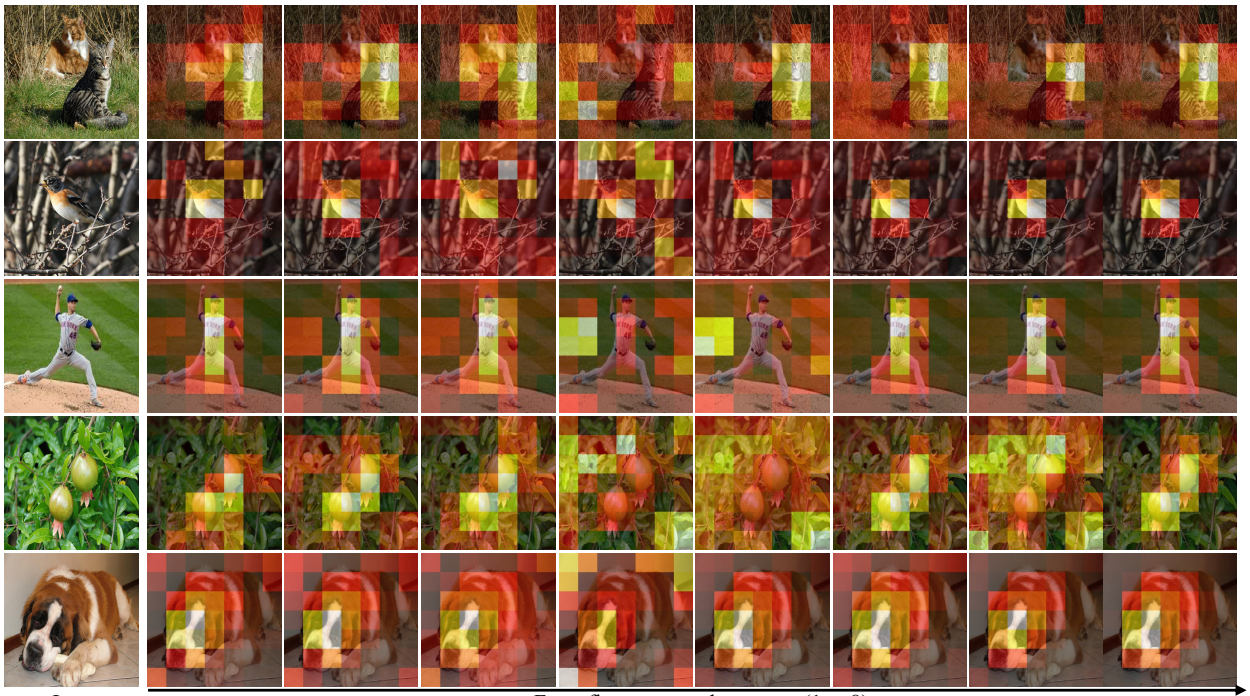


Figure 7. More token-global similarity visualization. Obviously, each group exhibits distinct similarities, and the similarities in all groups consistently emphasize the critical regions.

in our method, but no improvements were observed. Considering the computation and parameter overheads, we use our channel-attention module, which introduces 10 parameters. Considering the limited contribution of additional attention modules and strong baselines achieved by FCViT, we did not include convolutional attention modules in our architecture.

Repeated token-mixer. As shown in the FCViT Token-mixer, we repeat the global context and depth-wise convolution by two times. By reducing this operation repetition time to one, the performance decreases 0.9%. More repetitions would largely increase the computational overhead. Hence, we set the repetition to 2 in our implementation.

Depth-wise Convolution in Channel-Mixer(FFN). Results indicate that this simple modification can improve the performance by 1.2% (73.7% *vs.* 74.9%). Empirically, we set the kernel size to 3. Other values may lead to better performance, but not the key contributions to our work.

Removing each component in enhanced global context. Our enhanced global context has three components: dynamic global context, multi-group similarity, and competitive information bottleneck. We remove each component individually and present the results in Table 9. Clearly, each component

GC	Dynamic	Comp. Info	Group Sim.	top1
✓	✓	✓	✓	73.7
✗	✗	✗	✗	72.8 (↓0.9)
✓	✗	✓	✓	73.2 (↓0.5)
✓	✓	✗	✓	73.5 (↓0.2)
✓	✓	✓	✗	73.3 (↓0.4)

Table 9. Ablation of each component in enhanced global context.

Token-mixer	Param.	FLOPs	Blocks	Top-1
Self-attention	4.8M	0.8G	[3,3,5,2]	74.7
Conv-3x3	4.8M	0.9G	[3,3,9,2]	74.6
Conv-11x11	4.8M	1.0G	[3,3,7,2]	74.5
FCViT-Tiny	4.6M	0.8G	[3,3,5,2]	**74.9**

Table 10. Replacing Enhanced Global Context to Convolution or Attention. See the description for detail modifications.

can contribute to our FCViT. Combining them together, we achieve the best performance.

Replacing enhanced global context with conv or attention. Next, we replace the global context module with multi-head attention or convolution. It is relatively hard to compare FCViT and the related replacements fairly since the parameters and flops may change a lot. To this end, we tailor each variant to achieve a roughly fair comparison. For the

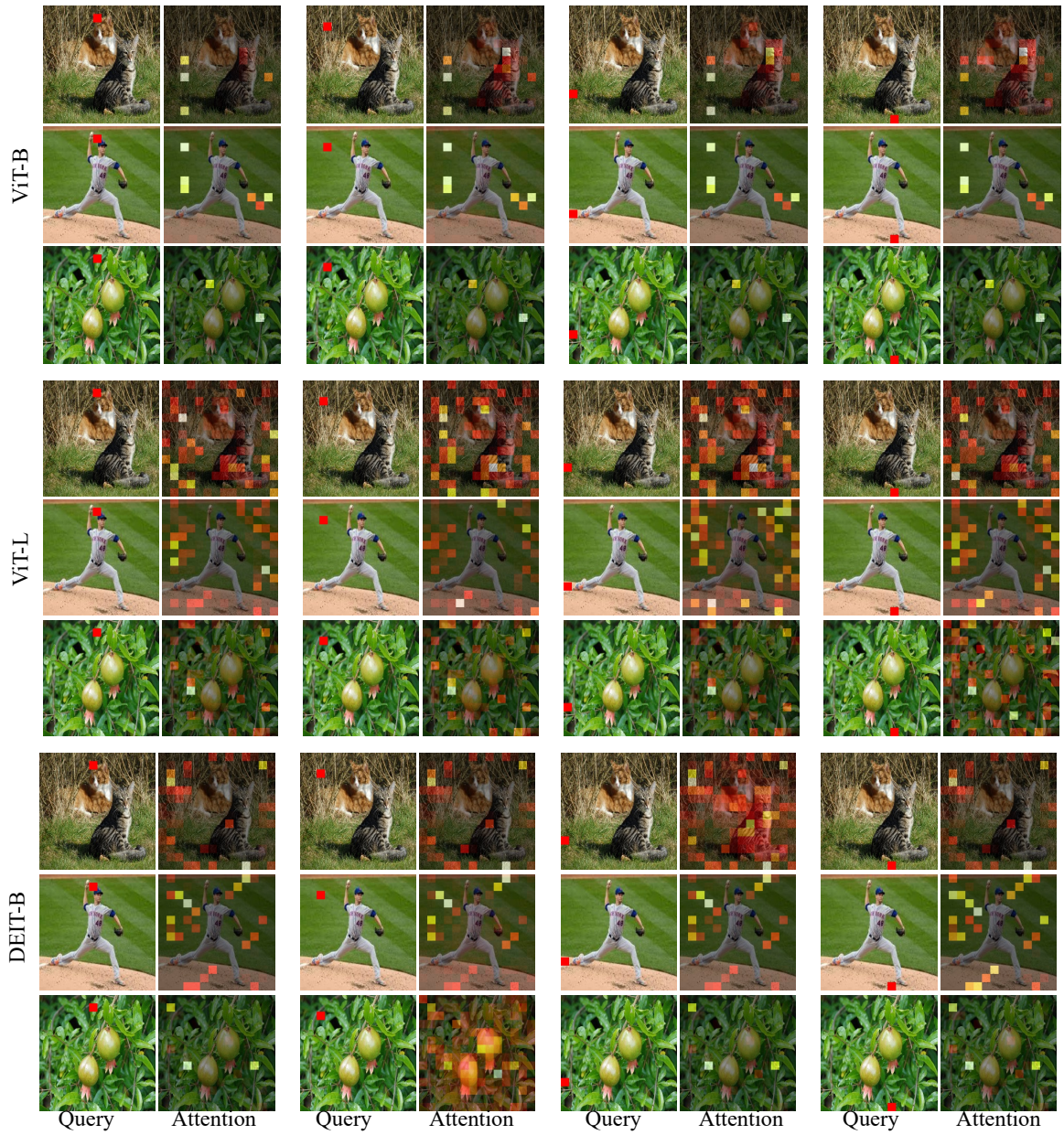


Figure 8. More attention maps of ViT variants. ViTs show a query-irrelevant behavior in general, while some bias may happen.

self-attention variant, we use adaptive-average-pooling (to a resolution of 7) to reduce the complexity. We consider 4 heads and 32 dimensions for each head. For convolution variants, we repeat convolution twice in each block and increase the block number in stage 3 to match the computations.

The results are presented in Table 10. Using even fewer parameters and FLOPs, our FCVIT-Tiny still achieves better results than the self-attention and convolution variants. Notice that convolution variants are deeper, and the self-attention variant introduces avg-pooling (which can be considered as convolution). All these variants train slower than our FCViT (around 1.2x to 1.3x time cost). Results showcased that our FCViT still outperforms all related variants, demonstrating the effectiveness of our method.

C. More Visualizations

Similarity map in FCViT. We also present more token-global similarity results on different examples, as shown in Fig 7. We use the pre-trained FCViT-B24 for illustration.

Attention Map in ViTs. To further support our motivation, we plot more attention maps in ViTs on more examples. The results are shown in Fig. 8.