You Only Need Less Attention at Each Stage in Vision Transformers

Shuoxi Zhang[1] , Hanpeng Liu[1*], Stephen Lin[2], Kun He[1†]
{zhangshuoxi, hanpengliu, brooklet60}@hust.edu.cn, stevelin@microsoft.com
[1]Huazhong University of Science and Technology, [2]Microsoft Research Asia

Abstract

The advent of Vision Transformers (ViTs) marks a substantial paradigm shift in the realm of computer vision. ViTs capture the global information of images through self-attention modules, which perform dot product computations among patchified image tokens. While self-attention modules empower ViTs to capture long-range dependencies, the computational complexity grows quadratically with the number of tokens, which is a major hindrance to the practical application of ViTs. Moreover, the self-attention mechanism in deep ViTs is also susceptible to the attention saturation issue. Accordingly, we argue against the necessity of computing the attention scores in every layer, and we propose the Less-Attention Vision Transformer (LaViT), which computes only a few attention operations at each stage and calculates the subsequent feature alignments in other layers via attention transformations that leverage the previously calculated attention scores. This novel approach can mitigate two primary issues plaguing traditional self-attention modules: the heavy computational burden and attention saturation. Our proposed architecture offers superior efficiency and ease of implementation, merely requiring matrix multiplications that are highly optimized in contemporary deep learning frameworks. Moreover, our architecture demonstrates exceptional performance across various vision tasks including classification, detection and segmentation.

1. Introduction

In recent years, computer vision has experienced rapid growth and development, primarily owing to the advances in deep learning and the accessibility of large datasets. Among the prominent deep learning techniques, Convolutional Neural Networks (CNNs) [8] have proven particularly effective, demonstrating exceptional performance across a wide range of applications, including image classification [8, 28], object detection [5, 22], and semantic segmentation [1, 23].

Inspired by the great success of Transformers [27] in natural language processing, Vision Transformers (ViTs) [4] divide each image into a set of tokens. These tokens are then encoded to produce an attention matrix that serves as a fundamental component of the self-attention mechanism. The computational complexity of the self-attention mechanism grows quadratically with the number of tokens, and the computational burden becomes heavier with higher-resolution images. Some researchers attempt to reduce token redundancy through dynamic selection [9, 21] or token pruning [33] to alleviate the computational burden of the attention computation. These approaches have demonstrated comparable performance to the standard ViT. However, methods involving token reduction and pruning necessitate meticulous design of the token selection module and may result in the inadvertent loss of critical tokens. In this work, we explore a different direction and rethink the mechanism of self-attention. In the attention saturation problem raised in [38], as the layers of ViTs are progressively deepened, the attention matrix tends to remain largely unaltered, mirroring the weight allocation observed in the preceding layers. Taking these considerations into account, we are prompted to pose the following question:

Is it really necessary to consistently apply the self-attention mechanism throughout each stage of the network, from inception to conclusion?

In this paper, we propose to modify the fundamental architecture of standard ViT by introducing the **L**ess-**A**ttention **Vi**sion **T**ransformer (LaViT). Our framework, as depicted in Fig. 1, consists of Vanilla Attention (VA) layers and our proposed Less Attention (LA) layers to capture the long-range relationships. In each stage, we exclusively compute the traditional self-attention and store the attention scores in a few initial Vanilla Attention (VA) layers. In subsequent layers, we efficiently generate attention scores by utilizing the previously calculated attention matrices, thereby mitigating the quadratic computational expense associated with self-attention mechanisms. Moreover, we integrate residual connections within the attention

*Equal contribution
†Corresponding author

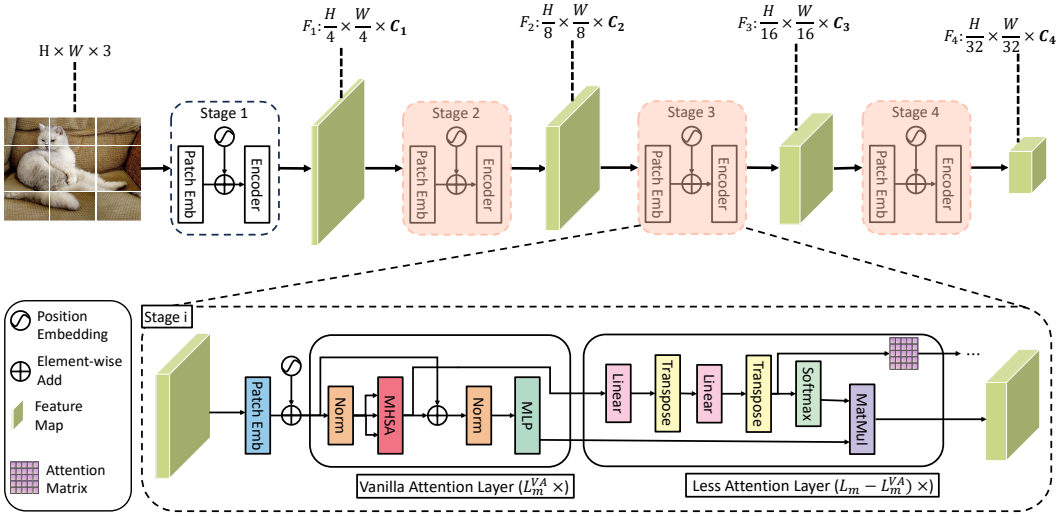


Figure 1. The architecture of our **L**ess-**A**ttention **Vi**sion **T**ransformer (LaViT). The bottom part: the proposed Less-Attention layer, which together with conventional Transformer blocks in the preceding layers constitutes the feature extraction module of this stage.

layers during downsampling across stages, allowing for the preservation of crucial semantic information learned in earlier stages while still transmitting global contextual information through alternate pathways. Finally, we carefully design a novel loss to preserve the diagonality of attention matrices during the transformation process. These key components enable our proposed ViT model to diminish both computational complexity and attention saturation, ultimately leading to notable performance improvements with reduced floating-point operations per second (FLOPs) and considerable throughput.

To verify the effectiveness of our proposed approach, we conduct comprehensive experiments on various benchmark datasets, comparing the performance of our model with existing state-of-the-art ViT variants (also recent efficient ViTs). The experimental results demonstrate the efficacy of our approach in addressing attention saturation and achieving superior performance in visual recognition tasks.

Our main contributions are summarized as follows:

- We present a novel ViT architecture that generates attention scores by re-parameterizing the attention matrix computed by preceding layers. This approach addresses both the attention saturation and the associated computational burden.
- Moreover, we propose a novel loss function that endeavors to preserve the diagonality of attention matrices during the process of attention re-parameterization. We posit that this is essential to uphold the semantic integrity of attention, ensuring that the attention matrices accurately reflect the relative importance among input tokens.
- Our architecture consistently performs favorably against several state-of-the-art ViTs, while having similar or even reduced computational complexity and memory consumption, across various vision tasks including classification, detection and segmentation.

2. Related Work

2.1. Vision Transformers

The Transformer architecture, initially introduced for machine translation [27], has since been applied to computer vision tasks through the growth of ViT [4]. The key innovation of ViT lies in its capability to capture long-range dependencies between distant regions of the image, achieved through the incorporation of self-attention mechanisms.

Drawn from the triumph of ViT, a plethora of variant models have emerged, each devised to ameliorate specific constraints inherent to the original architecture. For instance, DeiT [25] enhances data efficiency during training by incorporating the distillation token. Additionally, CvT [32] and CeiT [36] integrate the convolutional structure into the ViT framework to combine the strengths of CNNs (spatial invariance) and ViTs (long-range dependency modeling). These advancements underscore the ongoing evolution of transformer-based architectures in the field of computer vision.

2.2. Efficient Vision Transformers

Though highly effective, ViTs suffer from a huge computational burden. The research on efficient vision transformers addresses the quadratic cost of the self-attention operation by including hierarchical downsampling operations [17, 29, 30], token reduction [9, 21, 33], or lightweight architectural designs [18, 19]. Hierarchical downsampling operations address the quadratic computation of self-attention by reducing the token numbers gradually across the stages [17, 18, 29, 30] and enable ViTs to learn hierarchical structures. Another research direction introduces the token selection module to eliminate the least meaningful tokens and reduce the computational burden. For instance, [9, 21, 33] reorganize image tokens by preserving informa-

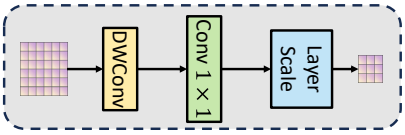


Figure 2. The downsampling on attention across stages.

tive image tokens and dropping ones with little attention to expedite subsequent MHSA and FFN computations.

2.3. Attention Mechanisms

The key component of ViTs is the attention mechanism, which computes pairwise interactions between all patches, resulting in quadratic complexity with respect to the input size. This problem leads to heavy inference computation, which hinders the practical application of ViTs in the real world. Several studies argue that the computational burden can be alleviated by utilizing sparse attention mechanisms, which selectively attend to a subset of patches based on their relevance or proximity. One notable approach is the Adaptive Sparse Token Pruning framework [31], which induces a sparse attention matrix, effectively addressing computational efficiency concerns. Furthermore, employing techniques like structured sparsity patterns [3, 6, 21] can further reduce computational complexity, thereby enhancing the overall efficiency of ViTs. Another urgent issue to be addressed is the problem of attention saturation, where the attention matrix displays limited variation as the layer depth increases. This issue has been acknowledged in studies such as DeepViT [38] and CaiT [26], which report that attention saturation hinders the ability of deep ViTs to capture additional semantic information and may even reduce training stability. Therefore, it is essential to carefully design the self-attention mechanism in ViTs to avoid sub-optimal solutions.

3. Methodology

In this section, we first review the basic design of the hierarchical vision transformer. Then, we discuss two major weaknesses of its attention mechanism, and propose that self-attention can be grasped with *less attention at each stage*. We dynamically re-parameterize the attention scores by utilizing the stored attention matrix from the previous layer, effectively mitigating the issue of attention saturation. Additionally, we integrate residual connections to facilitate the transfer of global relationships from earlier stages. Last but not least, we introduce a novel loss, the Diagonality Preserving loss, to preserve basic properties in the transformed attention (*i.e.*, representing the relationships among tokens).

3.1. Vision Transformer

Let $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ represent an input image, where $H \times W$ denotes the spatial resolution and C the number of channels. We first tokenize the image by partitioning it into $N = {}^{HW}/_{p^2}$ patches, where each patch $P_i \in \mathbb{R}^{p\times p\times C}$ $(i \in \{1, \ldots, N\})$ has a size of $p \times p$ pixels and C channels. The patch size p is a hyper-parameter that determines the granularity of the token. The patch embedding can be extracted by using a convolution operator with the stride and kernel size equal to the patch size. Each patch is then projected to the embedding space $\boldsymbol{Z} \in \mathbb{R}^{N\times D}$ through non-overlapping convolution, where D represents the dimension of each patch.

Multi-Head Self-Attention. We first provide a brief overview on the vanilla self-attention mechanism that processes the embedded patches and functions within the framework of Multi-Head Self-Attention blocks (MHSAs). In the l-th MHSA block, the input $\boldsymbol{Z}_{l-1}, l \in \{1, \cdots, L\}$, is projected into three learnable embeddings $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{N\times D}$. The multi-head attention aims to capture the attention from different views; for simplicity, we choose H heads, where each head is a matrix with the dimension $N \times \frac{D}{H}$. The h-th head attention matrix $\mathbf{A}_h$ can be calculated by:

$$\mathbf{A}_h = \text{Softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^{\mathsf{T}}}{\sqrt{d}}\right) \in \mathbb{R}^{N\times N}. \tag{1}$$

$\mathbf{A}_h, \mathbf{Q}_h$, and $\mathbf{K}_h$ are the attention matrix, query, and key of the h-th head, respectively. We also split the value $\mathbf{V}$ into H heads. To avoid vanishing gradients caused by the sharpness of the probability distribution, we divide the inner product of $\mathbf{Q}_h$ and $\mathbf{K}_h$ by $\sqrt{d}$ $(d = D/H)$. The attention matrix is concatenated as:

$$\begin{aligned} \mathbf{A} &= \text{Concat}(\mathbf{A}_1, \cdots, \mathbf{A}_h, \cdots, \mathbf{A}_H); \\ \mathbf{V} &= \text{Concat}(\mathbf{V}_1, \cdots, \mathbf{V}_h, \cdots, \mathbf{V}_H). \end{aligned} \tag{2}$$

The calculated attention among spatially split tokens may guide the model to focus on the most valuable tokens within the visual data. Subsequently, weighted linear aggregation is applied to the corresponding value $\mathbf{V}$:

$$\boldsymbol{Z}^{\text{MHSA}} = \mathbf{AV} \in \mathbb{R}^{N\times D}. \tag{3}$$

Downsampling Operation. Several studies have incorporated hierarchical structures into ViTs, drawing inspiration from the success of hierarchical architectures in CNNs. These works partition the Transformer blocks into M stages and apply downsampling operations before each Transformer stage, thereby reducing the sequence length. In our study, we employ a downsampling operation using a convolutional layer with the kernel size and stride set to 2. This approach permits the flexible adjustment of the feature map's scale at each stage, thereby establishing a Transformer hierarchical structure that mirrors the organization of the human visual system.

3.2. The Less-Attention Framework

The overall framework of our network architecture is illustrated in Fig. 1. In each stage, we extract the feature representation in two phases. At the initial several Vanilla At-

tention (VA) layers, we conduct the standard MHSA operation to capture the overall long-range dependencies. Subsequently, we simulate the attention matrices to mitigate quadratic computation and address attention saturation at the following Less-Attention (LA) layers by applying a linear transformation to the stored attention scores. Herein, we denote the attention score before the Softmax function of the initial l-th VA layer in the m-th stage as $\mathbf{A}_m^{\text{VA},l}$, which is computed by the following standard procedure:

$$\mathbf{A}_m^{\text{VA},l} = \frac{\mathbf{Q}_m^l(\mathbf{K}_m^l)^\top}{\sqrt{d}}, \ \ l \leq L_m^{\text{VA}}. \tag{4}$$

Here, $\mathbf{Q}_m^l$ and $\mathbf{K}_m^l$ represent the queries and keys from the l-th layer of the m-th stage, following the downsampling from the preceding stage. And L_m^{VA} is used to denote the number of VA layers. After the initial vanilla attention phase, we discard the traditional quadratic MHSA and apply transformations on $\mathbf{A}_m^{\text{VA}}$ to lessen the amount of attention computation. This process entails two linear transformations with a matrix transposition operation in between. To illustrate, let us consider the attention matrix in the l-th ($l > L_m^{\text{VA}}$) layer (LA layer) of the stage:

$$\begin{aligned} \mathbf{A}_m^l &= \Psi(\Theta(\mathbf{A}_m^{l-1})^\top)^\top, \ \ L_m^{\text{VA}} < l \leq L_m, \\ \mathbf{Z}^{\text{LA},l} &= \text{Softmax}(\mathbf{A}_m^l)\mathbf{V}^l. \end{aligned} \tag{5}$$

In this context, the transformations denoted by Ψ and Θ refer to linear transformation layers with a dimension of $\mathbb{R}^{N\times N}$. Here, L_m, L_m^{VA} represent the number of layers and the number of VA layers in the m-th stage, respectively. The insertion of the transposition operation between these two linear layers serves the purpose of maintaining the matrix similarity behavior. This step is essential due to the fact that the linear transformation in a single layer conducts the transformations row-wise, which could potentially result in the loss of diagonal characteristics.

3.3. Residual-based Attention Downsampling

When the computation traverses across stages in hierarchical ViTs, downsampling operations are often employed on the feature map. While this technique reduces the token numbers, it may result in the loss of essential contextual information. Consequently, we posit that the attention affinity learned from the preceding stage could prove advantageous for the current stage in capturing more intricate global relationships. Drawing inspiration from ResNet [7], which introduces shortcut connections to mitigate feature saturation issues, we adopt a similar concept and incorporate it into the downsampling attention computation within our architecture. By introducing a shortcut connection, we can introduce the inherent bias into the current MHSA block. This allows the attention matrix from the previous stage to effectively guide the attention computation of the current stage, thereby preserving crucial contextual information.

However, directly applying the shortcut connection to the attention matrix might pose challenges in this context, primarily due to the difference in the attention dimensions between the current stage and the preceding stage. Here, we design an Attention Residual (AR) module that consists of a depth-wise convolution (DWConv) and a $\text{Conv}_{1\times1}$ layer to downsample the attention map from the previous stage while keeping the semantic information. We denote the last attention matrix (at the L_{m-1} layer) of the previous stage (the $m-1$-th stage) as $\mathbf{A}_{m-1}^{\text{last}}$, and the downsampled initial attention matrix of the current (the m-th) stage as $\mathbf{A}_m^{\text{init}}$. $\mathbf{A}_{m-1}^{\text{last}}$ has the dimension of $\mathbb{R}^{B\times H\times N_{m-1}\times N_{m-1}}$ (N_{m-1} denotes the token number at the $m-1$-th stage). We view the multi-head dimension H as the channel dimension in regular image space, thus with the DWConv operator (stride $=2$, kernel size $=2$), we may capture the spatial dependencies among the tokens during attention downsampling. The output matrix after the DWConv transformation fits the size of the attention matrix of the current stage, *i.e.*, $\mathbb{R}^{B\times H\times N_m\times N_m}$ ($N_m = {}^{N_{m-1}}\!/_2$ in our case). After depth-width convolution on the attention matrix, we then perform $\text{Conv}_{1\times1}$ to exchange information across different heads. Our attention downsampling is illustrated in Fig. 2, and the transformation from $\mathbf{A}_{m-1}^{\text{last}}$ to $\mathbf{A}_m^{\text{init}}$ can be expressed as:

$$\mathbf{A}_m^{\text{init}} = \text{Conv}_{1\times1}\left(\text{Norm}(\text{DWConv}(\mathbf{A}_{m-1}^{\text{last}}))\right), \tag{6}$$

$$\mathbf{A}_m^{\text{VA}} \leftarrow \mathbf{A}_m^{\text{VA}} + \text{LS}(\mathbf{A}_m^{\text{init}}), \tag{7}$$

where LS is the Layer-Scale operator introduced in [26] to alleviate attention saturation. $\mathbf{A}_m^{\text{VA}}$ is the attention score for the first layer in the m-th stage, which is calculated by adding the standard MHSA with Eq. 4 and the residual calculated by Eq. 6.

Two fundamental design principles guide our attention downsampling module. First, we utilize DWConv to capture spatial local relationships during downsampling, thereby enabling the efficient compression of attention relationships. Second, the $\text{Conv}_{1\times1}$ operation is utilized to exchange the attention information across heads. This design is pivotal as it facilitates the efficient propagation of attention from the preceding stage to the subsequent stage. Incorporating the residual attention mechanism necessitates only minor adjustments, typically involving adding a few lines of code to the existing ViT backbone. It is worth emphasizing that such a technique can be seamlessly applied to various versions of the Transformer architecture. The only prerequisite is to store the attention scores from the previous layer and establish the skip-connections to this layer accordingly. The importance of this module will be further illuminated through comprehensive ablation studies.

3.4. Diagonality Preserving Loss

We have carefully designed the Transformer modules by incorporating attention transformation operators, aiming to mitigate the issues of computational cost and attention saturation. However, a pressing challenge remains—ensuring that the transformed attention preserves the inter-token relationships. It is well established that applying transformations to attention matrices can compromise their capacity to capture similarities, largely due to the linear transformation treating the attention matrix row-wise. Thus, we design a an alternative approach to guarantee that the transformed attention matrix retains the fundamental properties necessary to convey associations among tokens. A conventional attention matrix should possess the following two properties, *i.e.*, diagonality and symmetry:

$$\begin{aligned} &\mathbf{A}_{ij} = \mathbf{A}_{ji}, \\ &\mathbf{A}_{ii} > \mathbf{A}_{ij}, \forall j \neq i. \end{aligned} \tag{8}$$

Thus, we design the diagonality preserving loss of the l-th layer to keep these two basic properties as:

$$\begin{aligned} \mathcal{L}_{\text{DP},l} = &\sum_{i=1}^{N}\sum_{j=1}^{N} |\mathbf{A}_{ij} - \mathbf{A}_{ji}| \\ &+ \sum_{i=1}^{N}((N-1)\mathbf{A}_{ii} - \sum_{j\neq i}\mathbf{A}_{j}). \end{aligned} \tag{9}$$

Here, $\mathcal{L}_{\text{DP}}$ is the **D**iagonality **P**reserving loss aiming at preserving the properties of attention matrix of Eq. 8. We add our Diagonality Preserving Loss on all transformation layers with the vanilla cross-entropy (CE) loss [4], thus the total loss in our training can be presented as:

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{CE}} + \sum_{m=1}^{M}\sum_{l=1}^{L_m}\mathcal{L}_{\text{DP},l}, \\ \mathcal{L}_{\text{CE}} &= \text{cross-entropy}(Z_{\texttt{Cls}}, y), \end{aligned} \tag{10}$$

where $Z_{\texttt{Cls}}$ is the classification token of the representation in the last layer.

3.5. Complexity Analysis

Our architecture consists of four stages, each comprising L_m layers. The downsampling layer is applied between each consecutive stage. As such, the computational complexity of traditional self-attention is $\mathcal{O}(N_m^2 D)$, whereas the associated K-Q-V transformation incurs a complexity of $\mathcal{O}(3N_m D^2)$. In contrast, our method leverages an $N_m \times N_m$ linear transformation within the transformation layer, thereby circumventing the need for computing the inner products. Consequently, the computation complexity of our attention mechanism in the transformation layer is reduced to $\mathcal{O}(N_m^2)$, representing a reduction factor of D. Additionally, since our method calculates the query embeddings solely within the Less-Attention layer, our K-Q-V transformation complexity is likewise diminished by a factor of 3.

In the downsampling layer between consecutive stages, considering a downsample rate of 2 as an example, the computational complexity of the DWConv in the attention downsampling layer can be calculated as Complexity $= 2\times2\times\frac{N_m}{2}\times\frac{N_m}{2}\times D = \mathcal{O}(N_m^2 D)$. Similarly, the complexity of the $\text{Conv}_{1\times1}$ operation in the attention residual module is also $\mathcal{O}(N_m^2 D)$. However, it is important to note that attention downsampling only occurs once per stage. Therefore, the additional complexity introduced by these operations is negligible when compared to the complexity reduction achieved by the Less-Attention layer.

Table 1. Detailed configurations of the LaViT series. 'Blocks' and 'Heads' refer to the number of blocks ($[L^1, L^2, L^3, L^4]$) and heads in four stages, respectively. 'Channels' refers to the input channel dimensions across the four stages. And 'N_{LA}' denotes the layer within each stage at which the utilization of the Less-Attention layer begins.

Models	Channels	Blocks	Heads	N_{LA}
LaViT-T	[64,128,320,512]	[2,2,2,2]	[1,2,5,8]	[0,0,2,2]
LaViT-S	[64,128,320,512]	[3,4,6,3]	[1,2,5,8]	[0,0,3,2]
LaViT-B	[64,128,320,512]	[3,3,18,3]	[1,2,5,8]	[0,2,4,3]

4. Experiments

In this section, we evaluate our model's performance on two benchmark datasets: ImageNet-1K [8] for classification, COCO2017 [10] for detection and ADE20K [37] for segmentation. We compare our model with other state-of-the-art works on these datasets to demonstrate its effectiveness and efficiency. Furthermore, we perform ablation studies to investigate the necessity and contributions of each component in the proposed model. This analysis provides valuable insights into the role of each part and helps to establish the efficacy of our approach.

4.1. Architecture Variants

To ensure an equitable comparison with other models while maintaining a similar level of computational complexity, we establish three models: LaViT-T, LaViT-S, and LaViT-B. The detailed configuration information is provided in Table 1, and we follow the same network structure as

[1]The 90% notation in brackets indicates that we keep the token ratio of 90% to represent the visual data during the training of the corresponding ViTs—DynamicViT and EviT, respectively. Additionally, given our aim to strike a balance between efficiency and effectiveness, we will not compare our results to high-performance but computationally intensive models, such as Swin-B-V2 [14] and ConvNeXt-B [15].

Model	Params (M)	FLOPs (G)	Throughput (image/s)	Top1 (%)
ResNet-18	11.7	1.8	**4454**	69.8
RegNetY-1.6G	11.2	1.6	1845	78.0
DeiT-T	**5.7**	**1.3**	3398	72.2
PVT-T	13.2	1.9	1768	75.1
PVTv2-b1	13.1	2.1	1231	78.7
LaViT-T	10.9	1.6	2098	**79.2**
ResNet-50	25.0	4.1	1279	76.2
RegNetY-4G	20.6	4.0	1045	79.4
EfficientNet-B4	**19.0**	4.2	387	82.4
EfficientViT-B2	24.0	4.5	1587	82.1
DeiT-S	22.1	4.6	1551	79.9
DeepViT-S	27.0	6.2	1423	82.3
PVT-S	24.5	3.8	1007	79.8
CvT-S	25.8	7.1	636	82.0
Swin-T	28.3	4.5	961	81.2
PVTv2-b2	25.4	4.0	695	82.0
DynamicViT-S (90%)	24.1	4.0	1524	79.8
EViT-S (90%)	23.9	4.1	**1706**	79.7
LiT-S	27.0	4.1	1298	81.5
PPT-S	22.1	3.1	1698	79.8
LaViT-S	22.4	**3.3**	1546	**82.6**
ResNet-101	45.0	7.9	722	77.4
ViT-B	86.6	17.6	270	77.9
DeiT-B	86.6	17.5	582	81.8
Swin-S	49.6	8.7	582	83.1
Swin-B	87.8	15.4	386	**83.4**
DynamicViT-B (90%)	76.6	14.1	732	81.5
EViT-B (90%)	78.6	15.3	852	81.3
LiT-M	48.0	8.6	638	83.0
PPT-B	86.0	14.5	714	81.4
PVT-M	44.2	6.7	680	81.2
PVT-L	61.4	9.8	481	81.7
LaViT-B	**39.6**	**6.1**	**877**	83.1

Table 2. Comparison of different backbones on ImageNet-1K classification. Except for EfficientNet (EfficientNet-B4), all models are trained and evaluated with an input size of 224 $\times$ 224. The least computations and fastest throughput appear in **blue bold**, and the best results appear in **bold**.[1]

PVT [29, 30] except for introducing the Less-Attention Transformer encoder and skip-connection attention downsampling. The number of blocks, channels, and heads affects the computational cost.

4.2. Baselines

We conduct a thorough experimental evaluation of our proposed method by comparing it with various CNNs, ViTs, and hierarchical ViTs. Specifically, the following baselines are used:

- **CNNs:** ResNet [7], RegNet [20] and EfficientNet [24].
- **ViTs:** ViT [4], DeiT [25], CvT [36],DeepViT [38], FocalViT [35] and SwinTransformer [13].
- **Efficient ViTs:** HVT [17], PVT [29], DynamicViT [21], EViT [9], LiT [19], EfficientViT [2] and PPT [33].

4.3. Image Classification on ImageNet-1K

Settings. The image classification experiments are conducted on the ImageNet-1K dataset. Our experimental protocol follows the procedures outlined in DeiT [25], with the exception of the model itself. Specifically, we apply the same data augmentation and regularization techniques employed in DeiT. We utilize the AdamW optimizer [16] to train our models from scratch for 300 epochs (with a 5-epoch warm-up). The initial learning rate is set to 0.005 and varies according to a cosine scheduler. The global batch size is set to 1024, distributed across 4 GTX-3090 GPUs. During the test on the validation set, the input images are first resized to 256 pixels, followed by a center crop of 224 x 224 pixels to evaluate the classification accuracy.

Results. We present the classification results on ImageNet-1K in Table 2. The models are classified into three groups based on their computational complexity: tiny (approximately 2G), small (approximately 4G), and base (approximately 9G). Our approach achieves competitive performance compared to state-of-the-art ViTs with markedly reduced computational requirements. Specifically, in the tiny and small model clusters, our method surpasses all other existing models by at least 0.2% and 0.5%, respectively, while maintaining a substantially lower computational cost, which is our principal concern. In the base-size models, our architecture, which incorporates the base structure of PVT but includes the Less-Attention component, demonstrates superior performance over two PVT-based models (PVT-M and PVT-L). Furthermore, we also compare our architecture to several efficient ViT designs (DynamicViT, EViT, LiT, efficientViT and PPT). We observe that our results reflect a better balance between effectiveness and efficiency. Note that our design necessitates reduced computation cost owing to our resource-efficient Less-Attention mechanism, rendering our lightweight module an attractive option for implementing ViT on mobile platforms.

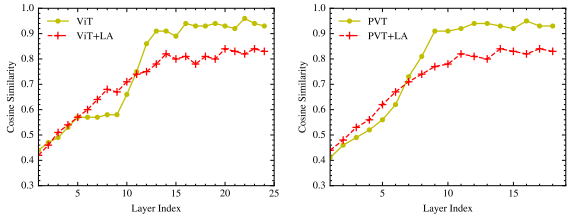
4.4. Object Detection on COCO2017

Settings. We conduct the detection experiments on COCO 2017 [10] dataset. We test the model effectiveness on RetinaNet [12]. We follow the common practice by initializing the backbone with pre-trained weights obtained from ImageNet-1K. In addition, we use AdamW [16] optimizer, and train the network with the batchsize of 16 on 8 GPUs.

Results. We present the results of object detection in Table 3. It is evident that our LaViT model exhibits a notable advantage over both its CNN and Transformer counterparts. Specifically, with the 1$\times$ schedule, our tiny version LaViT-T achieves 9.9-12.5 AP^b against ResNet under comparable settings, while the small version LaViT-S outperforms its CNN counterpart by 8.1-10.3 AP^b. This trend persists with the 3$\times$ schedule, as our LaViT consistently demonstrates competitive performance. Particularly noteworthy is our architecture's ability to consistently outperform the Swin Transformer in terms of detection performance while imposing a smaller training burden. Thus, the results on COCO2017 reaffirm our assertion that our

Backbone	#Param.	FLOPs	RetinaNet 1×						RetinaNet 3× + MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L	AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L
ResNet50	38	239	36.3	55.3	38.6	19.3	40.0	48.8	39.0	58.4	41.8	22.4	42.8	51.6
PVT-Small	34	226	40.4	61.3	43.0	25.0	42.9	55.7	42.2	62.7	45.0	26.2	45.2	57.2
Swin-T	39	245	41.5	62.1	44.2	25.1	44.9	55.5	43.9	64.8	47.1	28.4	47.2	57.8
LaViT-T(ours)	33	202	**46.2**	**67.2**	**49.1**	**29.6**	**50.2**	**61.3**	**48.4**	**69.9**	**51.7**	**31.8**	**52.2**	**64.1**
ResNet101	58	315	38.5	57.8	41.2	21.4	42.6	51.1	40.9	60.1	44.0	23.7	45.0	53.8
PVT-M	54	283	41.9	63.1	44.3	25.0	44.9	57.6	43.2	63.8	46.1	27.3	46.3	58.9
Swin-S	60	335	44.5	65.7	47.5	27.4	48.0	59.9	46.3	67.4	49.8	31.1	50.3	60.9
LaViT-S(ours)	47	290	**46.7**	**68.3**	**49.7**	**29.9**	**50.7**	**61.7**	**48.9**	**70.3**	**52.2**	**33.1**	**52.6**	**65.4**

Table 3. Results on COCO object detection using the RetinaNet [12] framework. 1× refers to 12 epochs, and 3× refers to 36 epochs. MS means multi-scale training. AP^b and AP^m denotes box mAP and mask mAP, respectively. FLOPs are measured at resolution 800×1280.



(a) ViT (b) PVT

Figure 3. The similarity ratio of the generated self-attention maps of the current layer with its previous layer.

carefully designed LaViT model enhances feature extraction with reduced computational overhead.

4.5. Semantic Segmentation on ADE20K

Settings. We conduct experiments on semantic segmentation using the ADE20K dataset, which comprises 150 classes and 20,000 images for training, and 2,000 images for validation. Our backbone networks for segmentation are Semantic FPN [11] and UperNet [34]. We follow the training settings established in [13] and resize images to 512×512 for training. We train UperNet for 160k iterations and SemanticFPN for 80k iterations. The initial learning rate is set to 6×10^{-5}, utilizing a poly scheduler for learning rate decay. The experiment is conducted by using the batch size of 16 across 4 GTX3090 GPUs.

Results. Table 4 provides an overview of the segmentation results. Our model demonstrates superiority over Swin Transformer, exhibiting an mIoU improvement of +2.6 with Semantic FPN and +2.7 with UperNet. In the Semantic FPN test, our LaViT-S achieves a relatively modest increase of +0.9 mIoU compared to the baseline (PVT-S), but notably with significantly fewer computations. When integrated into the UperNet architecture, LaViT achieves substantial improvements of +2.7 mIoU, +1.0 mIoU, and +1.4 mIoU when compared to various mainstream models. These competitive results are maintained even when employing test time augmentation. In particular, LaViT-S outperforms Focal-T by +1.4 mIoU and +2.5 MS mIOU. These findings underscore LaViT's ability to produce high-quality semantic segmentation outputs while operating within the framework of its computation-efficient attention mechanism.

4.6. Ablation Study

Attention Saturation. To demonstrate the efficacy of our Less-Attention module in addressing attention saturation, we present the attention similarity ratio (cosine similarity computed by the attention map in the current layer and its previous layer) in Figure 3. We conduct the comparison using two backbones, namely, ViT and PVT. In 3a, we select the ViT architecture with 25 layers and no hierarchical structure. In 3b, we employ PVT-M as our baseline and assess the attention similarity at the 3rd stage, which consists of 18 layers. Both sets of results clearly illustrate that the original architecture encounters a significant attention saturation issue. However, this phenomenon is effectively mitigated by incorporating our modules, enabling deep attention to fulfill its intended role.

Extendability of Less-Attention Module. We extend our Less-Attention module to various ViT architectures, and report the results in Table 5. The incorporation of the Less-Attention layer into any of the foundational Transformer architectures leads to enhancements in accuracy while concurrently reducing computational demands. Notably, the most significant improvement is observed when incorporating the module into the vanilla ViT/DeiT architecture. This may be attributed to the fact that the vanilla ViT/DeiT does not have a hierarchical structure, thereby experiencing considerable attention saturation issues. Moreover, when integrating our method into DeepViT, we observe the most substantial decrease in computational resources. These findings jointly underscore the scalability of our method, demonstrating that the application of LA module can render existing ViT architectures more practical and feasible.

Importance of Each Component. We conduct ablation studies on the proposed module with the ImageNet-1k dataset, and the results are shown in Table 6. On both networks (*i.e.*, tiny and small), our proposed modules prove to be indispensible for Transformer training. The baseline, which replaces the Less-Attention layer with MHSA, corresponds exactly to the PVT model, exhibiting a decrease in predictive accuracy by 0.5% and 0.6% compared to our model. Additionally, removing the attention residual modules, denoted as "w/o AR", results in a reduction of predictive accuracy by 0.2% and 0.4%. Lastly, and most

Backbone	Semantic FPN 80k			UperNet 160K			
	Param (M)	FLOPs (G)	mIOU (%)	Param (M)	FLOPs (G)	mIOU (%)	MS mIOU (%)
ResNet-50	28.5	183	36.7	-	-	-	-
Swin-T	31.9	182	41.5	59.9	945	44.5	45.8
PVT-S	30.2	146	43.2	-	-	-	-
Twin-S	28.3	144	43.2	54.4	932	46.2	47.1
LiT-S	32.0	172	41.3	57.8	978	44.6	45.9
Focal-T	-	-	-	62.0	998	45.8	47.0
LaViT-S	**25.1**	**122**	**44.1**	**52.0**	**920**	**47.2**	**49.5**

Table 4. Segmentation performance of different backbones in Semantic FPN and UpperNet framework on ADE20K. The least computation appears in **blue bold**, and the best results appear in **bold**.

Backbone	Tiny		Small	
	Top-1 Acc(%)	FLOPs (G)	Top-1 Acc(%)	FLOPs (G)
ViT	72.2	1.4	79.1	4.6
ViT$_{+LA}$	73.2(↑ 1.0)	1.2(↓ 14.2%)	80.0(↑ 0.9)	4.0(↓ 13.1%)
DeiT	72.2	1.4	79.9	4.7
DeiT$_{+LA}$	73.4(↑ 1.2)	1.2(↓ 14.2%)	80.4(↑ 0.5)	4.2(↓ 10.6%)
DeepViT	73.4	1.5	80.9	4.8
DeepViT$_{+LA}$	73.8(↑ 0.4)	1.1(↓ 25.8%)	81.4(↑ 0.5)	4.2(↓ 12.6%)
CeiT	76.2	1.2	82.0	4.5
CeiT$_{+LA}$	76.7(↑ 0.5)	1.1(↓ 9.0%)	82.4(↑ 0.4)	4.1(↓ 8.8%)
HVT	75.7	1.4	80.4	4.6
HVT$_{+LA}$	76.2(↑ 0.5)	1.2(↓ 15.2%)	80.8(↑ 0.4)	4.2(↓ 13.4%)
PVT	75.1	1.9	79.8	3.8
PVT$_{+LA}$	75.9(↑ 0.8)	1.4(↓ 25.6%)	80.4(↑ 0.6)	3.2(↓ 15.7%)
Swin	81.2	4.5	83.2	8.7
Swin$_{+LA}$	81.7(↑ 0.5)	4.0(↓ 11.1%)	83.5(↑ 0.3)	7.8(↓ 10.3%)

Table 5. Top-1 classification accuracy on ImageNet-1k using different transformer backbones and their corresponding Less-Attention plug-in variants. Footnote 'LA' indicates the addition of our Less-Attention module to the respective backbone architectures. ↑ and ↓ denote the increase in Top-1 accuracy and the percentage of FLOPs reduction, respectively.

Model	Module			Tiny	Small
	AR	LA	$\mathcal{L}_{DP}$		
w/o LA	-	-	-	78.7	82.0
w/o AR	-	✓	✓	79.0	82.2
LaViT	✓	✓	✓	79.2	82.6
w/o $\mathcal{L}_{DP}$	✓	✓	-	59.1(↓ 20.1)	57.1(↓ 25.5)

Table 6. Ablation study of the proposed module on the ImageNet-1k dataset. Baseline means we remove all the proposed modules, resulting in the PVT Transformer baseline. 'AR' and 'LA' indicate the Attention-Residual and Less-Attention modules, respectively. 'w/o $\mathcal{L}_{DP}$' indicates we remove the Diagonality Preserving loss.

importantly, we assert that the additional loss function to preserve diagonal correlations is vital for effectively comprehending semantic information in visual data. When relying solely on the CE loss, the model's predictions deteriorate. This might be attributed to the potential limitation of relying solely on the transformation for attention matrices, which could compromise their capacity to express correlations among tokens. All these experimental findings collectively emphasize the contribution of each component within our model architecture.

Model	Stage 3			Stage 4		
	L2	L3	L4	L1	L2	L3
LaViT-S	79.1	82.6	82.4	80.1	82.6	82.3
LaViT-B	78.9	82.5	83.1	80.4	82.3	83.1

Table 7. Ablation study on the layer where Less-Attention starts. We conduct experiments on the last two stages—Stage 3,4. 'L2' beneath Stage 3 means we use the Less-Attention layer to replace the vanilla encoder from the second layer in the third Stage.

Less-Attention Selection. In deep ViTs, careful selection of the starting layer for Less-Attention is crucial. Thus, we design experiments to select the starting layer for Less-Attention in the network architecture, and the results are presented in Table 7. As shown in the table, directly using the Less-Attention layer from the second layer in the stage leads to a decrease in model performance. This phenomenon could be attributed to overly relying on the semantics of the first MHSA layer. Thus, leveraging the Less-Attention layer at deeper layers in the stage may mitigate this issue. Furthermore, while utilizing the Less-Attention layer at relatively deeper layers does not affect the model performance much, it may lead to increased computational costs. This contradicts the design objective of our architecture to reduce the computational overhead.

5. Conclusion

Aiming to reduce the costly self-attention computations, we proposed a new model called Less-Attention Vision Transformer (LaViT). LaViT leverages the computed dependency in Multi-Head Self-Attention (MHSA) blocks and bypasses the attention computation by re-using attentions from previous MSA blocks. We additionally incorporated a straightforward Diagonality Preserving loss, designed to promote the intended behavior of the attention matrix in representing relationships among tokens. Notably, our Transformer architecture effectively captures cross-token associations, surpassing the performance of the baseline while maintaining a computationally efficient profile in terms of quantity of parameters and floating-point operations per second (FLOPs). Comprehensive experimentation has confirmed the efficacy of our model as a foundational architecture for multiple downstream tasks. Specifically, the proposed model demonstrates superiority over previous Transformer architectures, resulting in state-of-the-art performance in classification and segmentation tasks.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, pages 2481–2495, 2017. 1

[2] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. In *IEEE/CVF ICCV*, 2023. 6

[3] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *NeurIPS*, pages 19974–19988, 2021. 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 5, 6

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *TPAMI*, pages 142–158, 2015. 1

[6] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv:2106.04263*, 2021. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 6

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Commun.ACM*, pages 84–90, 2017. 1, 5

[9] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv:2202.07800*, 2022. 1, 2, 6

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 5, 6

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 7

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6, 7

[13] 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*, pages 9992–10002, 2021. 6, 7

[14] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, pages 12009–12019, 2022. 5

[15] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 5

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[17] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *ICCV*, pages 377–386, 2021. 2, 6

[18] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. *NeurIPS*, pages 14541–14554, 2022. 2

[19] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *AAAI*, pages 2035–2043, 2022. 2, 6

[20] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, pages 10428–10436, 2020. 6

[21] Yongming rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 34:13937–13949, 2021. 1, 2, 3, 6

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 1

[24] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 6

[25] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 2, 6

[26] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, pages 32–42, 2021. 3, 4

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 2

[28] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017. 1

[29] 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*, pages 568–578, 2021. 2, 6

[30] 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*, pages 415–424, 2022. 2, 6

[31] Cong Wei, Brendan Duke, Ruowei Jiang, Parham Aarabi, Graham W Taylor, and Florian Shkurti. Sparsifiner: Learning sparse instance-dependent attention for efficient vision transformers. In *CVPR*, pages 22680–22689, 2023. 3

[32] 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. 2

[33] Xinjian Wu, Fanhu Zeng, Xiudong Wang, Yunhe Wang, and Xinghao Chen. Ppt: Token pruning and pooling for efficient vision transformers. *arXiv:2310.01812*, 2023. 1, 2, 6

[34] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 7

[35] 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. *arXiv:2107.00641*, 2021. 6

[36] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *ICCV*, pages 579–588, 2021. 2, 6

[37] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.*, pages 302–321, 2019. 5

[38] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv:2103.11886*, 2021. 1, 3, 6