LightViT: Towards Light-Weight Convolution-Free Vision Transformers

Tao Huang[1,2] **Lang Huang**[3] **Shan You**[1*] **Fei Wang**[4] **Chen Qian**[1] **Chang Xu**[2]
[1]SenseTime Research
[2]School of Computer Science, Faculty of Engineering, The University of Sydney
[3]The University of Tokyo [4]University of Science and Technology of China

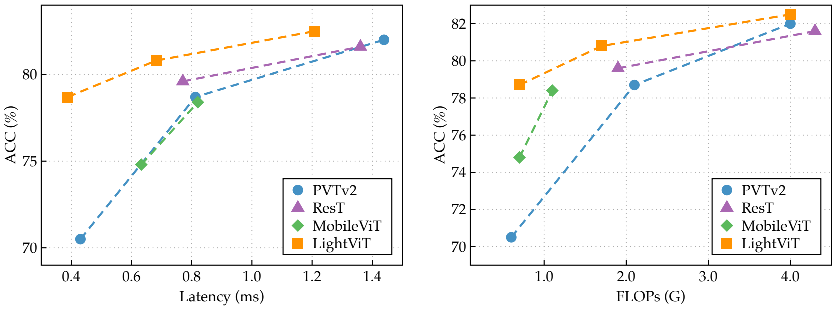


Figure 1: Comparisons of the proposed LightViT and other efficient ViTs on ImageNet.

Abstract

Vision transformers (ViTs) are usually considered to be less light-weight than convolutional neural networks (CNNs) due to the lack of inductive bias. Recent works thus resort to convolutions as a plug-and-play module and embed them in various ViT counterparts. In this paper, we argue that the convolutional kernels perform information aggregation to connect all tokens; however, they would be actually unnecessary for light-weight ViTs if this explicit aggregation could function in a more homogeneous way. Inspired by this, we present LightViT as a new family of light-weight ViTs to achieve better accuracy-efficiency balance upon the pure transformer blocks without convolution. Concretely, we introduce a global yet efficient aggregation scheme into both self-attention and feed-forward network (FFN) of ViTs, where additional learnable tokens are introduced to capture global dependencies; and bi-dimensional channel and spatial attentions are imposed over token embeddings. Experiments show that our model achieves significant improvements on image classification, object detection, and semantic segmentation tasks. For example, our LightViT-T achieves 78.7% accuracy on ImageNet with only 0.7G FLOPs, outperforming PVTv2-B0 by 8.2% while 11% faster on GPU. Code is available at https://github.com/hunto/LightViT.

1 Introduction

Recently, vision transformers (ViTs) have gained noticeable success on vision tasks such as image classification [6, 20, 27, 29], object detection [16, 20], and semantic segmentation [25, 35]. However, despite the state-of-the-art performance in large-size ViT models, their light-weight counterpart will lose its advantage over the typical convolutional neural networks (CNNs). For instance, it is observed that DeiT-Ti [29] and PVTv2-B0 [31] can achieve 72.2% and 70.5% accuracies on ImageNet, while

[*] Correspondence to: Shan You <youshan@sensetime.com>.

Preprint. Under review.

the typical CNN model RegNetY-800M [22] achieves 76.3% accuracy with similar FLOPs, which seems a catastrophic failure for ViTs in terms of light-weight models.

It is recognized that CNNs are generally more efficient for their intrinsically-biased architecture designs, such as parameter sharing, local information aggression, and spatial reduction. Therefore, to enhance the light-weight property of ViTs, recent works mainly borrow the inductive bias from CNNs to develop various counterparts in a hybrid or heterogeneous manner, *i.e.* integrating convolutions into transformer blocks as a plug-and-play module. For example, ResT [41] proposes to leverage convolutions to reduce the spatial dimensions of key and values in self-attention; LVT [36] adopts convolutions to perform local self-attention for low-level features and multi-scale attentions for high-level features. Moreover, some methods [21, 24, 33] aim to improve CNNs via interpreting self-attentions into existing CNN blocks. A recent study MobileViT [21] incorporates transformer into MobileNetV2 [23] to obtain global representations in the upper stages.

So far, the community shows that convolution seems to be essential for efficient ViTs. However, *is convolution really necessary for light-weight ViTs? Can't we have an efficient homogeneous ViT with no convolution but only the transformer block?* In this paper, we get down to investigating this problem and hope to push the limit of light-weight ViT one step further. By rewinding the convolution in hybrid ViTs, we regard it as a way of information aggregation since it builds explicit connections to all the tokens through shared kernels. In this way, we are inspired to introduce these aggregation priors to the ViTs as well, which stimulates new design for the two key components in transformer blocks, *i.e.*, self-attention and feed-forward network (FFN):

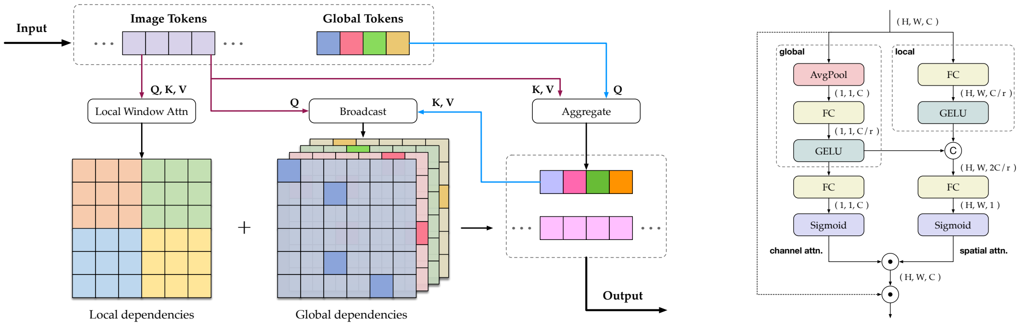
- For self-attention, we leverage the local window attention [20] for effective spatial priors and efficient calculation. In particular, we propose to introduce learnable *global tokens* to aggregate the information of local tokens by modeling their global dependencies. Then these global dependencies are broadcast into every local token. In this way, each image token could be more informative since it benefits from both local and global features as Figure 2 (a). Note that the global dependencies can be calculated quite efficiently.
- For FFN, as the only non-linearity in plain transformer block, it plays an important role in feature extractions by modeling feature patterns and implicitly capturing the spatial dependencies. However, its representation power would be restricted due to the small channel dimensions in light-weight models. Therefore, we propose a bi-dimensional attention module to explicitly aggregate the global dependencies among spatial and channel dimensions, thus the capacity would be enhanced since features will be filtered more adaptively.

Based on the new self-attention and FFN, we also make empirical studies to give a more practical design for efficient ViTs, which help us achieve a better efficiency-accuracy tradeoff. For example, we observe that the early stages in hierarchical ViTs are inefficient due to a large number of tokens in self-attention, and thus propose to build ViT stages from a moderate dimension ($\text{stride} = 8$) such as discarding the Stage 0 as Figure 3. As a result, we can develop a new family of light-weight convolution-free ViTs dubbed *LightViT*. Extensive experiments show that our LightViT does enjoy significant superiority and efficiency advantage over various computer vision benchmarks. For example, as shown in Figure 1, our LightViT-S achieves 80.8% accuracy on ImageNet, significantly outperforms ResT-Small [41] by 1.2% with 0.2G smaller FLOPs and 14% faster in inference.

2 Related Work

2.1 Efficient vision transformers

Recent approaches [30, 31, 41] on efficient ViTs mainly focus on interpreting convolutions into transformer blocks. PVT [30] conducts a CNN-like hierarchical structure and adopts convolutions to reduce the spatial dimensions of self-attention and perform feature downsampling. PVTv2 [31] further improves PVT by introducing overlapping patch embedding and convolutional feed-forward network. ResT [41] proposes a memory-efficient self-attention by compressing the spatial dimensions and projects the interaction across the attention-heads dimension using convolutions. LVT [36] introduces convolutions in self-attention to perform local self-attention for low-level features and multi-scale attention for high-level features. Different from previous models built upon the ViT structure, MobileViT [21] aims to improve mobile CNNs by incorporating attention into MobileNetV2



(a) **Local-global broadcast of attention.**

(b) **Bi-dimensional attn.**

Figure 2: **Our proposed efficient feature aggregation on attention and FFN.** (a) Besides local window self-attention (window size is 3 here), we propose local-global broadcast of attention to broadcast the global information using additional global tokens. (b) The architecture of our bi-dimensional attention on FFN.

blocks for better global representations. This paper investigates a new variant of efficient ViTs without using convolutions in blocks.

2.2 Window-based vision transformers

Despite the success of plain ViTs on image classification, it remains challenging for downstream tasks since the computation cost would grow quadratic to the image size on these high-resolution tasks. Recent works [3, 20, 30, 31] conduct hierarchical structures with multiple stages like CNNs to make ViTs more efficient and friendly to existing frameworks. Among these methods, window-based methods [3, 20] adopt local window attention to partition image tokens into multiple non-overlapped windows and perform self-attentions inside each window, yielding a linear computation complexity to the image size.

However, local window attention has been observed to have limited receptive fields and weak long-range dependencies. Therefore, some methods propose to bring global interactions to the local window attention. Twins [3] applies global attention to image tokens (queries) and window representations (keys and values) summarized by convolutions. MSG-Transformer [7] binds learnable message tokens on each local window and adopts channel shuffle among these tokens to exchange information. Focal transformer [37] performs local window attentions using keys and values down-sampled with different strides, thus aggregating information on multiple receptive fields. Nevertheless, these global information aggregations still have a quadratic computation cost to the input image size and encounter large computation cost especially on large resolutions. In this paper, we introduce global tokens to aggregate the global information freely on the whole feature map, which only has linear computation complexity to the input image size and brings noticeable improvements with negligible FLOPs increment.

3 Efficient feature aggregation for LightViT

In this section, we formally illustrate the two key designs for our LightViT, namely aggregated self-attention and FFN, which leverage local-global broadcast of attention and bi-dimensional attention, respectively.

3.1 Aggregated self-attention with local-global broadcast

Self-attention among all the tokens is one of the key advantages of ViTs compared to local convolution. However, directly applying self-attention to the whole image requires quadratic computation complexity to the input image size. To reduce the computation cost, the typical local window self-attention [20] partitions the feature map into multiple non-overlapping windows, then performs self-attention independently in each window. This paper utilizes the local window self-attention as the base module.

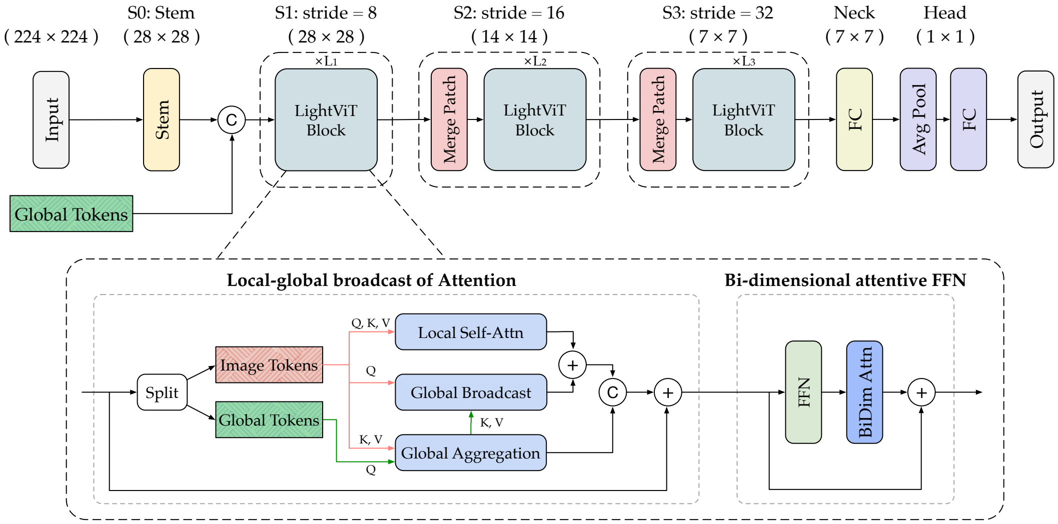


Figure 3: **LightViT architecture.** We follow a hierarchical structure design, but remove the stride = 4 stage for better efficiency. ⓒ denotes concatenation in the token axis, and ⊕ denotes element-wise addition.

Local self-attention. Given an input feature map $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$, instead of calculating attention on the flattened $H \times W$ patches, we partition $\boldsymbol{X}$ into non-overlapping windows with shape $(\frac{H}{S} \times \frac{W}{S}, S \times S, C)$, where S denotes window size (we use $S = 7$ following Swin [20]), then apply self-attention within each local window, which is equivalent to the local window attention in Swin and Twins. Formally, the local self-attention is computed as

$$\boldsymbol{X}_{\text{local}} = \text{Attention}(\boldsymbol{X}_q, \boldsymbol{X}_k, \boldsymbol{X}_v) := \text{SoftMax}(\boldsymbol{X}_q\boldsymbol{X}_k^\top)\boldsymbol{X}_v, \tag{1}$$

where $\boldsymbol{X}_q$, $\boldsymbol{X}_k$, and $\boldsymbol{X}_v$ are produced by Q, K, and V projections, respectively. As a result, the computation complexity $(H \times W)^2$ of self-attention is reduced to $(\frac{H}{S} \times \frac{W}{S}) \times (S \times S)^2 = H \times W \times S \times S$.

The local self-attention can be an efficient and effective way to aggregate local dependencies with window priors. However, it has a drawback of lacking long-range dependencies and large receptive fields. For a light way to obtain global interactions, this paper proposes to first gather the valuable global dependencies to a small feature space, then broadcast the aggregated global information to the local features. This light information squeeze-and-expand scheme can enhance the local features with negligible computation cost, and we find it sufficient and effective in experiments.

Global aggregation. To gather global information in $\boldsymbol{X}$, we propose a learnable embedding $\boldsymbol{G} \in \mathbb{R}^{T\times C}$, which is computed along with image tokens in all LightViT blocks. The proposed embedding $\boldsymbol{G}$, dubbed as *global token*, has two functions: global information aggregation and broadcast. As illustrated in Figure 2 (a), it first aggregates the global representations on the whole image feature map, then broadcasts the global information into the feature maps. All the information exchanges are performed in a homogeneous way using attention. Specifically, along with the computation of local self-attention, we gather the global representations using input global tokens $\boldsymbol{G}$ (queries) and image tokens $\boldsymbol{X}$ (keys and values), *i.e.*,

$$\hat{\boldsymbol{G}} = \text{Attention}(\boldsymbol{G}_q, \boldsymbol{X}_k, \boldsymbol{X}_v), \tag{2}$$

the output new tokens $\hat{\boldsymbol{G}}$ are then used in global broadcast and passed to next block for usage.

Global broadcast. With the aggregated global information, the aim is to broadcast it back to the image tokens, thus the image features can be enhanced by receiving global dependencies from tokens outside the local window. We perform this broadcast by adopting global tokens $\hat{\boldsymbol{G}}$ as keys and values in attention:

$$\boldsymbol{X}_{\text{global}} = \text{Attention}(\boldsymbol{X}_q, \hat{\boldsymbol{G}}_k, \hat{\boldsymbol{G}}_v), \tag{3}$$

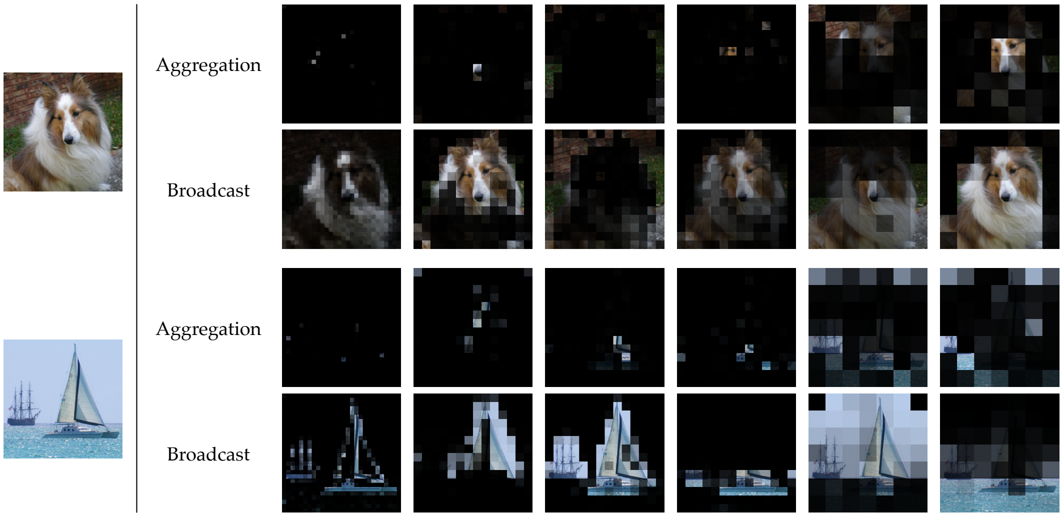


Figure 4: **Visualization of the learned global attentions of LightViT-T on ImageNet.** Our global tokens *aggregate* key parts (*e.g.*, nose and eyes of the dog) of the image (1st row), then *broadcast* the information to related pixels globally (2nd row) to enhance and highlight the target objects.

then the final output image tokens are computed through element-wise addition on local and global features, *i.e.*,

$$\boldsymbol{X}_{\text{new}} = \boldsymbol{X}_{\text{local}} + \boldsymbol{X}_{\text{global}}. \tag{4}$$

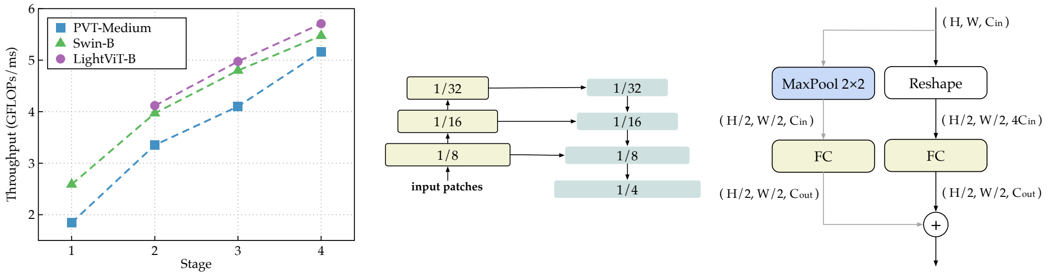
Note that the computation complexities of global aggregation and global broadcast ($H \times W \times T$) are negligible, as the number of global tokens T (*e.g.*, $T = 8$ in LightViT-T) is much smaller than image size $H \times W$ and window size $S \times S$ in LightViT.

We visualize the learned global attentions in Figure 4. We can see that, the global tokens first aggregate key information (*e.g.*, nose and eyes of the dog) of the feature map through our global aggregation, then deliver the information to related pixels using global broadcast, and thus the features of target objects (*e.g.*, dog and boat) can be enhanced and highlighted with global information.

3.2 Aggregated FFN with bi-dimensional attention

As the only non-linear part in transformer block, feed-forward network (FFN) plays an important role in feature extraction. Since all the tokens are forwarded point-wisely and share the same linear layers in FFN, the non-linear activations are usually conducted on enlarged channel dimensions produced by a linear layer for an effect and sufficient capture of feature patterns. However, the dimensions of channels are still insufficient in light-weight models, where the channels are limited to small ones for reducing the computation cost, and thus their performance is severely restricted. Another drawback of the plain FFN is the lack of explicitly dependency modeling on the spatial level, which is highly important to vision tasks. Though the spatial feature aggregations can be performed implicitly through the weight sharing among tokens, it is still challenging for the light ViTs to capture these implications. To this end, some ViT variants [17, 31, 39] propose to aggregate spatial representations before the activation layer using convolutions, resulting in a noticeable increase in computation cost.

In this paper, inspired by the attention mechanisms [10, 32], which are widely adopted to explicitly model the feature relationships in light-weight CNNs [9, 28], we propose a bi-dimensional attention module to capture the spatial and channel dependencies and refine the features. As shown in Figure 2 (b), the module consists of two branches: channel attention branch and spatial attention branch. The channel attention branch first averages the input features on spatial dimension to aggregate global representations, which are then used to compute the channel attention with a linear transformation. For spatial attention, we model the pixel-wise relations by concatenating the global representations to every token features (local representations). To reduce the FLOPs, we add a linear reduction layer before the attention fully-connected (FC) layers following SE [10], and set the reduction ratio r to 4 in our models.



(a) **Throughput of each stage.** (b) **FPN on 3 stages.** (c) **Residual patch merging.**

Figure 5: (a): Throughput of each stage shows that the earlier stages are less efficient. (b): Architecture of our feature pyramid on object detection. (c): We extend the patch merging module in Swin [20] with a cheap residual branch.

Our proposed bi-dimensional attention module can be used as a plug-and-play module for existing ViT variants. With only a slight increase in computation cost, it can explicitly model the spatial and channel relationships and enhance the representation power of FFN.

4 Practical design for more efficient LightViT

In this section, we formulate our design choices for LightViT through experiments. We empirically find that several improvements on the model components can lead to better performance and efficiency, and thus make our LightViTs more efficient. For fair comparisons, we keep the same FLOPs by adjusting the channels uniformly in experiments.

4.1 Hierarchical structures with fewer stages

Vision transformers with hierarchical structures [20, 30] have shown great performance on image classification and downstream tasks. However, these methods have smaller inference speeds compared to the vanilla ViT. For example, the vanilla ViT model DeiT-S has a throughput of 961 on GPU with 4.6G FLOPs, while PVTv2-B2 only has 695 with 4.0G FLOPs. One major reason is that the earlier stages in hierarchical structures have larger numbers of tokens, making self-attention less efficient. As shown in Figure 5 (a), we measure the inference efficiency (speed / FLOPs) of hierarchical ViTs, and find that the earlier stages are less FLOPs-efficient than the later stages. As a result, in this paper, we remove the first $\mathrm{stride} = 4$ stage and keep the later $\mathrm{stride} = \{8, 16, 32\}$ stages in the hierarchical structures. The experimental results on ImageNet and COCO detection in the following table show that, our LightViT-T achieves significant efficiency improvement by removing the first stage, and even achieves higher accuracy.

Method	Params (M)	FLOPs (G)	Throughput (images/s)	Top-1 (%)	COCO det mAP
w/ 4 stages	9.2	0.73	1643	77.9	37.5
LightViT-T	9.4	0.73	2578	**78.7**	**37.8**

On downstream tasks such as object detection, it usually adopts a 4-stage feature pyramid network (FPN) [18]. Removing the first stage may have a possible risk of weakening the transfer performance. In this paper, we show that directly adopting a 3-stage FPN as Figure 5 (b) suffices, and could also achieve competitive performance compared to those 4-stage backbones (see the above table). Moreover, recent works [1, 16] also show that, the plain ViTs can achieve promising performance on downstream tasks with minor modifications in FPN.

Downsample with residual patch merging. To conduct feature downsampling in hierarchical ViTs, two commonly-used modules are stride-2 convolution [30, 31] and linear patch merging in Swin [20]. In this paper, we adopt a patch merging module for better efficiency and a more homogeneous way in the transformer, with an additional cheap residual branch for better gradient flows, as shown in Figure 5 (c). The following table shows that our residual patch merging achieves slightly higher accuracy with negligible efficiency drop.

Method	Params (M)	FLOPs (G)	Throughput (images/s)	Top-1 (%)
w/o residual in patch merging	9.3	0.73	2597	78.4
LightViT-T	9.4	0.73	2578	**78.7**

Overlapping patch embedding. Previous methods [31, 34] show that, replacing the original patch embedding in plain ViTs with overlapping path embedding (OPE) could benefit the performance and training robustness. In this paper, we also conduct an OPE stem (see Figure 3), and obtain higher performance on ImageNet as in the table below.

Method	Params (M)	FLOPs (G)	Throughput (images/s)	Top-1 (%)
w/o OPE	9.5	0.74	2652	77.4
LightViT-T	9.4	0.73	2578	**78.7**

Table 1: **LightViT architecture variants.** `B`: number of blocks. `C`: number of channels. `H`: number of heads. We set numbers of global tokens to $[8, 16, 24]$ (T, S, B).

Stage	Stride	LightViT-T	LightViT-S	LightViT-B
S0: Stem	$^1/_8$	`C=64`	`C=96`	`C=128`
S1: LightViT-Block	$^1/_8$	`B=2 C=64 H=2`	`B=2 C=96 H=3`	`B=3 C=128 H=4`
S2: LightViT-Block	$^1/_{16}$	`B=6 C=128 H=4`	`B=6 C=192 H=6`	`B=8 C=256 H=8`
S3: LightViT-Block	$^1/_{32}$	`B=6 C=256 H=8`	`B=6 C=384 H=12`	`B=6 C=512 H=16`

4.2 Architecture variants

We design a series of LightViT models on different scales to validate our effectiveness on light-weight models. The macro structure of our models is illustrated in Figure 3. We first conduct a stem block to embed the input image into image tokens with $\mathrm{stride} = 8$, where several convolution layers are included. For the main body of our network, we construct three stages (S1-S3) with the same LightViT blocks inside, and a residual patch merging layer is conducted before S2 and S3 for feature downsampling. The window size S of attention is set to 7, and reduction ratios r of spatial and channel attentions in FFN are equal to 4. Detailed settings of our variants are summarized in Table 1.

5 Experiments

We validate the efficacy of our proposed model on various vision tasks: image classification, object detection, instance segmentation.

5.1 Image classification on ImageNet

Training strategy. We train our models on ImageNet-1K dataset [5], and validate the top-1 accuracy on ImageNet validation set. We adopt common data augmentations on ViTs including RandAugmentation [4], MixUp [40], *e.t.c.* Detailed training strategy refers to Table 2.

Experimental results. Our performance on ImageNet validation set is summarize in Table 3. LightViT outperforms recent efficient ViTs under the basic 224×224 resolution, especially having large improvements on light-weight scales (less than 2G FLOPs). For instance, LightViT-T obtains a record 78.7% accuracy with 0.7G FLOPs, significantly outperforms those ViT variants with attention-convolution hybrid blocks. Besides, our model also obtains higher throughput compared to existing efficient ViTs, and achieves better FLOPs-accuracy and latency-accuracy trade-offs, as shown in Figure 1.

Table 2: **Training settings on ImageNet dataset.**

Config	Value
Batch size	1024
Optimizer	AdamW
Weight decay	0.04
LR decay	cosine
Base LR	1e-3
Minimum LR	1e-6
Warmup LR	1e-7
Warmup epochs	20
Training epochs	300
Augmentation	RandAug (2, 9) [4]
Color jitter	0.3
Mixup alpha	0.2
Cutmix alpha	1.0
Erasing prob.	0.25
Drop path rate	0.1 (T, S), 0.3 (B)

Table 3: **Image classification performance on ImageNet validation dataset.** Throughput is measured on a single V100 GPU following [20, 29]. *Hybrid* denotes using both attention and convolution in blocks. All models are trained and evaluated on 224 × 224 resolution. †: accuracy reported by DeiT [29].

Model	Block	Params	FLOPs	Throughput	Top-1
	type	(M)	(G)	(image/s)	(%)
RegNetY-800M [22]	CNN	6.3	0.8	3321	76.3
PVTv2-B0 [31]	Hybrid	3.4	0.6	2324	70.5
SimViT-Micro [15]	Hybrid	3.3	0.7	1004	71.1
MobileViT-XS [21]	Hybrid	2.3	0.7	1581	74.8
LVT [36]	Hybrid	5.5	0.9	1545	74.8
LightViT-T	Transformer	9.4	0.7	2578	**78.7**
RegNetY-1.6G [22]	CNN	11.2	1.6	1845	78.0
MobileViT-S [21]	Hybrid	5.6	1.1	1219	78.4
PVTv2-B1 [31]	Hybrid	13.1	2.1	1231	78.7
ResT-Small [41]	Hybrid	13.7	1.9	1298	79.6
DeiT-Ti [29]	Transformer	5.7	1.3	2612	72.2
LightViT-S	Transformer	19.2	1.7	1467	**80.8**
RegNetY-4G† [22]	CNN	21.0	4.0	1045	80.0
Twins-PCPVT-S [3]	Hybrid	24.1	3.8	807	81.2
ResT-Base [41]	Hybrid	30.3	4.3	735	81.6
PVTv2-B2 [31]	Hybrid	25.4	4.0	695	82.0
DeiT-S [29]	Transformer	22	4.6	961	79.8
Swin-T [20]	Transformer	29	4.9	765	81.3
LightViT-B	Transformer	35.2	3.9	827	**82.1**

5.2 Object detection and instance segmentation

Training strategy. We conduct experiments on MS-COCO dataset [19] and adopt the Mask R-CNN architecture with an FPN [18] neck for fair comparisons. Since LightViT only has three stage, we make a simple modification to make it compatible with the existing architecture. As shown in Figure 5(b), we append a 2 × 2 transposed convolution with stride 2 to upsample the largest output of the top-down path to the size of $1/4$, forming a pyramid of 4 levels. The models are fine-tuned from the ImageNet pre-trained weights. We use the AdamW [14] optimizer for training for which the hyper-parameters are: the batch size as 16, the learning rate as $1\mathrm{e}^{-4}$, weight decay as 0.05, and the stochastic depth [11] ratios as the ones used in the pre-training. Following the common practice, we adopt the same data augmentations scheme as [20] and the 1×/3× training schedule of mmdetection [2], which has 12/36 training epochs in total and decays the learning rate by a factor of 10 at the $3/4$ and $11/12$ of the total epochs. The standard metrics of the COCO dataset are used here to evaluate the performance, including the Average Precision (AP), AP$_{50}$, and AP$_{75}$ for both object detection and instance segmentation.

Experimental results. We report the results on MS-COCO dataset in Table 4. LightViT performs on par with the recent 4-stage ViTs with a similar amount of FLOPs. Specifically, LightViT-S achieves 40.0% AP^b and 37.4% AP^m using the 1× schedule, which is better than the hybrid method PVT-T with $\sim$200 GFLOPs.

5.3 Ablation studies

Ablation on proposed aggregation scheme in attention and FFN. In LightViT, we improve the vanilla local window self-attention with additional global attention for global representations. While in FFN, we propose a bi-dimensional attention module to refine the features for more efficient filtering. Here we conduct experiments for ablations of these components in Table 5. **Local self-attention vs.**

Table 4: **Object detection and instance segmentation performance on COCO** `val2017`**.** The FLOPs are measured on 800 × 1280. All the models are pretrained on ImageNet-1K.

Backbone	Params	FLOPs	Mask R-CNN 1x schedule						Mask R-CNN 3x + MS schedule					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-18 [8]	31	207	34.0	54.0	36.7	31.2	51.0	32.7	36.9	57.1	40.0	33.6	53.9	35.7
ResNet-50 [8]	44	260	38.0	58.6	41.4	34.4	55.1	36.7	41.0	61.7	44.9	37.1	58.4	40.1
ResNet-101 [8]	101	493	40.4	61.1	44.2	36.4	57.7	38.8	42.8	63.2	47.1	38.5	60.1	41.3
PVT-T [30]	33	208	36.7	59.2	39.3	35.1	56.7	37.3	39.8	62.2	43.0	37.4	59.3	39.9
PVT-S [30]	44	245	40.4	62.9	43.8	37.8	60.1	40.3	43.0	65.3	46.9	39.9	62.5	42.8
PVT-M [30]	64	302	42.0	64.4	45.6	39.0	61.6	42.1	44.2	66.0	48.2	40.5	63.1	43.5
LightViT-T	28	187	37.8	60.7	40.4	35.9	57.8	38.0	41.5	64.4	45.1	38.4	61.2	40.8
LightViT-S	38	204	40.0	62.9	42.6	37.4	60.0	39.3	43.2	66.0	47.4	39.9	63.0	42.7
LightViT-B	54	240	41.7	64.5	45.1	38.8	61.4	41.4	45.0	67.9	48.8	41.2	64.8	44.2

vanilla global self-attention: The vanilla self-attention module in ViT [6] can perform global and dense attentions on tokens. We compare it with the widely-used local-window self-attention. The results show that local-window self-attention has higher accuracy on light-weight model due to better inductive bias. **+ global attention**: Our proposed global attention gains significant improvements (76.9% $\sim$ 78.0%) over the local self-attention baseline, and only has a minor increase on FLOPs. **+ spatial attention.** Spatial attention in FFN further achieves an 0.4% higher accuracy with negligible computation cost, as it explicitly captures the spatial dependencies and selectively focus on the salient tokens to capture the image structure better. **+ channel attention.** Our final architecture with channel attention on LightViT achieves the best 78.7% accuracy. Compared with our local window self-attention baseline, LightViT-T gains a significant 1.8% improvement, with better feature aggregation scheme equipped in attention and FFN.

Table 5: **Ablation of the components of our architecture with LightViT-T settings.**

Attention		FFN		Params	FLOPs	Top-1
local self-attn.	global attn.	spatial attn.	channel attn.	(M)	(G)	(%)
✗	✗	✗	✗	8.0	0.88	76.5
✓	✗	✗	✗	8.0	0.66	76.9
✓	✓	✗	✗	8.8	0.71	78.0
✓	✓	✓	✗	9.1	0.73	78.4
✓	✓	✓	✓	9.4	0.73	**78.7**

6 Conclusion

This paper proposes a new series of light-weight ViTs dubbed LightViT. While most recent works aim to combine convolutions and transformers in efficient ViTs, we seek a better performance-efficiency trade-off on the pure ViT blocks without convolution. This paper proposes a different and more homogeneous way to perform better information aggregation on self-attention and feed-forward networks and achieve better performance on ImageNet classification and object detection than those hybrid models. We hope the work can help future related research study the difference between convolutions and transformers, and look forward to exploring better feature aggregation schemes upon them.

References

[1] A. Ali, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34, 2021. 6

[2] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 8

[3] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 8

[4] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 7

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 1, 9

[7] J. Fang, L. Xie, X. Wang, X. Zhang, W. Liu, and Q. Tian. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. *arXiv preprint arXiv:2105.15168*, 2021. 3

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9

[9] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 5

[10] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[11] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016. 8

[12] T. Huang, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu. Greedynasv2: greedier search with a greedy path filter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11902–11911, 2022.

[13] T. Huang, S. You, B. Zhang, Y. Du, F. Wang, C. Qian, and C. Xu. Dyrep: Bootstrapping training with dynamic re-parameterization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2022.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 8

[15] G. Li, D. Xu, X. Cheng, L. Si, and C. Zheng. Simvit: Exploring a simple vision transformer with sliding windows, 2021. 8

[16] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 1, 6

[17] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 5

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6, 8

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 8

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2, 3, 4, 6, 8

[21] S. Mehta and M. Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 2, 8

[22] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 2, 8

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[24] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021. 2

[25] R. Strudel, R. Garcia, I. Laptev, and C. Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1

[26] X. Su, T. Huang, Y. Li, S. You, F. Wang, C. Qian, C. Zhang, and C. Xu. Prioritized architecture sampling with monto-carlo tree search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10968–10977, 2021.

[27] X. Su, S. You, J. Xie, M. Zheng, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu. Vision transformer architecture search. *arXiv e-prints*, pages arXiv–2106, 2021. 1

[28] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[29] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 8

[30] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 2, 3, 6, 9

[31] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, pages 1–10, 2022. 1, 2, 3, 5, 6, 7, 8

[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[33] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 2

[34] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021. 7

[35] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[36] C. Yang, Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin, and A. Yuille. Lite vision transformer with enhanced self-attention. *arXiv preprint arXiv:2112.10809*, 2021. 2, 8

[37] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 3

[38] S. You, T. Huang, M. Yang, F. Wang, C. Qian, and C. Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2008, 2020.

[39] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021. 5

[40] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 7

[41] Q. Zhang and Y.-B. Yang. Rest: An efficient transformer for visual recognition. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 8

[42] M. Zheng, F. Wang, S. You, C. Qian, C. Zhang, X. Wang, and C. Xu. Weakly supervised contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10042–10051, 2021.

[43] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu. Ressl: Relational self-supervised learning with weak augmentation. *Advances in Neural Information Processing Systems*, 34:2543–2555, 2021.

7 Appendix

7.1 More ablation studies

Number of global tokens. We investigate the effects of the numbers of global tokens in our model. As shown in Table 6, we train our LightViT-T with 0, 2, 4, 8, 16, and 32 global tokens. We can see that the performance can be improved with only a small number of tokens, *e.g.*, only 2 tokens can improve the baseline by 0.5% with 0.01G addition on FLOPs. Besides, the number of 16 seems to be saturate for the LightViT-T model, as there are no further gains on 32 tokens. As a result, we set the tokens to a moderate size of 8 for a better efficiency-accuracy trade-off.

Table 6: **Ablation of numbers of global tokens.**

#global tokens	FLOPs (G)	Top-1 (%)
0	0.68	77.3
2	0.69	77.8
4	0.70	78.3
8	0.73	78.7
16	0.80	78.8
32	0.92	78.8

7.2 Discussion

Limitations. LightViT could significantly improve the transformer blocks without leveraging convolutions. However, its inference efficiency on edge devices might be worse than the convolutions, as current inference frameworks on edge devices are better optimized for convolution computations.

Society impacts. Investigating the effects of the proposed model requires large consumptions on computation resources, which can potentially raise the environmental concerns. However, it is valuable for us to explore efficient models, which can save a large volume of computation resources in training and deployment.

7.3 Implementation of local-global broadcast of attention

The pseudo code of our attention module is shown in Figure 6.

```python
def attention(image_tokens, global_tokens):
    img_q, img_k, img_v = qkv(image_tokens)
    glb_q = q(global_tokens)
    # local window self-attention
    local_q, local_k, local_v = window_partition(img_q, img_k, img_v)
    local_img = attn(local_q, local_k, local_v)
    local_img = window_reverse(local_img)
    # global aggregation
    global_tokens = attn(glb_q, img_k, img_v)
    # global broadcast
    glb_k, glb_v = glb_kv(global_tokens)
    global_img = attn(img_q, glb_k, glb_v)

    image_tokens = local_img + global_img
    return image_tokens, global_tokens
```

Figure 6: The PyTorch-like pseudo code of our attention module.

7.4 Implementation of bi-dimensional attention on FFN

The pseudo code of our bi-dimensional attention on FFN is shown in Figure 7.

```
global_reduce = nn.Linear(C, C/r)
local_reduce = nn.Linear(C, C/r)
channel_select = nn.Linear(C/r, C)
spatial_select = nn.Linear(C/r*2, 1)

def FFN(x):
    # original ffn
    x = mlp(x)  # [B, N, C]
    # bi-dimensional attention
    x_global = x.mean(1, keepdim=True)  # [B, 1, C]
    x_global = GELU(global_reduce(x_global))  # [B, 1, C/r]
    x_local = GELU(local_reduce(x))  # [B, N, C/r]
    # channel attention
    c_attn = channel_select(x_global).sigmoid()  # [B, 1, C]
    # spatial attention
    x_spatial = torch.cat(x_local, x_global.expand(-1, N, -1))
    s_attn = spatial_select(x_spatial).sigmoid()  # [B, N, 1]

    x = (c_attn * s_attn) * x
    return x
```

Figure 7: The PyTorch-like pseudo code of our bi-dimensional attention on FFN.