Multi-Scale Vision Longformer: 
A New Vision Transformer for High-Resolution Image Encoding

Pengchuan Zhang¹ Xiyang Dai¹† Jianwei Yang¹† Bin Xiao¹ Lu Yuan¹
Lei Zhang² Jianfeng Gao¹
¹ Microsoft Corporation {penzhan,xidai,jianwei.yang,bin.xiao,luyuan,jfgao}@microsoft.com
² International Digital Economy Academy (IDEA) leizhang@idea.edu.cn

Abstract

This paper presents a new Vision Transformer (ViT) architecture Multi-Scale Vision Longformer, which significantly enhances the ViT of [12] for encoding high-resolution images using two techniques. The first is the multi-scale model structure, which provides image encodings at multiple scales with manageable computational cost. The second is the attention mechanism of Vision Longformer, which is a variant of Longformer [3], originally developed for natural language processing, and achieves a linear complexity w.r.t. the number of input tokens. A comprehensive empirical study shows that the new ViT significantly outperforms several strong baselines, including the existing ViT models and their ResNet counterparts, and the Pyramid Vision Transformer from a concurrent work [47], on a range of vision tasks, including image classification, object detection, and segmentation. The models and source code are released at https://github.com/microsoft/vision-longformer.

1. Introduction

Vision Transformer (ViT) [12] has shown promising results on image classification tasks for its strong capability of long range context modeling. But its quadratic increase of both computational and memory complexity hinders its application on many vision tasks that require high-resolution feature maps computed on high-resolution images¹, like object detection [34, 24], segmentation [27, 6], and human pose estimation [49, 37]. Vision-language tasks, like VQA, image captioning, and image-text retrieval, also benefit from high-resolution feature maps [16, 53], which are extracted with pre-trained CNN models. Developing a vision Transformer that can process high-resolution feature maps is a critical step toward the goal of unifying the model architecture of vision and language modalities and improving multi-modal representation learning. In this paper, we propose a new vision Transformer architecture Multi-Scale Vision Longformer, which significantly enhances the baseline ViT [12] for encoding high-resolution images using two techniques: (1) the multi-scale model structure, and (2) the attention mechanism of Vision Longformer.

Models with multi-scale (pyramid, hierarchical) structure provide a comprehensive encoding of an image at multiple scales, while keeping the computation and memory complexity manageable. Deep convolutional networks are born with such multi-scale structure, which however is not true for the conventional ViT architecture. To obtain a multi-scale vision Transformer, we stack multiple (e.g., four) vision Transformers (ViT stages) sequentially. The first ViT stage operates on a high-resolution feature map but has a small hidden dimension. As we go to later ViT stages, the feature map resolution reduces while the hidden dimension increases. The resolution reduction is achieved by performing patching embedding at each ViT stage. In our experiments, we find that with the same number of model parameters and the same model FLOPs, the multi-scale ViT achieves a significantly better accuracy than the vanilla ViT on image classification task. The results show that the multi-scale structure not only improves the computation and memory efficiency, but also boosts the classification performance. The proposed multi-scale ViT has the same network structure as conventional (multi-scale) CNN models such as ResNet [14], and can serve as a replace-and-plug-in choice for almost all ResNet applications. In this paper, we demonstrate this plausible property in image classification, object detection and instance segmentation.

The multi-scale structure alone is not sufficient to scale up ViT to process high-resolution images and feature maps, due to the quadratic increase of the computation and memory complexity with respect to the number of tokens in the
self-attention layers. Compared to natural language tasks where data is 1-D, this problem is more severe in vision tasks where the increase in complexity is quartic (fourth order) with the increase of image resolution. For example, the computational complexity of a 4 x higher resolution multi-head self attention (MSA) layer (hidden dimension reduced by 4, i.e., \(4H \times 4W \times \frac{D}{4}\)) equals to that of 64 layers in the original size (i.e., \(H \times W \times D\)). To address this challenge, we develop a 2-D version of Longformer[3], called Vision Longformer, to achieve a linear complexity w.r.t. the number of tokens (quadratic w.r.t. resolution). Our experiments show that compared to the baseline ViT, Vision Longformer shows no performance drop while significantly reduces the computational and memory cost in encoding images. The result indicates that the “local attention + global memory” structure in Vision Longformer is a desirable inductive bias for vision Transformers. We also compare Vision Longformer with other efficient attention mechanisms. The result again validates its superior performance on both image classification and object detection tasks.

The main contributions of this paper are two-fold: (1) We propose a new vision Transformer that uses the multi-scale model structure and the attention mechanism of 2-D Longformer for efficient high-resolution image encoding. (2) We perform a comprehensive empirical study to show that the proposed ViT significantly outperforms previous ViT models, their ResNet counterparts, and ViTs with several other efficient attention mechanisms, on image classification, object detection and segmentation tasks.

2. Related Work

The Vision Transformer (ViT) [12] applies a standard Transformer, originally developed for natural language processing (NLP), for image encoding by treating an image as a word sequence, i.e., splitting an image into patches (words) and using the linear embeddings of these patches as an input sequence. ViT has shown to outperform convolution neural network (CNN) models such as the ResNet [14], achieving state-of-the-art performance on multiple image classification benchmarks, where training data is sufficient. DeiT [43] is another computer vision model that leverages Transformer. It uses a teacher-student strategy specific to Transformers to improve data efficiency in training. Thus, compared to ViT, it requires much less training data and computing resources to produce state-of-the-art image classification results. In addition to image classification, Transformers have also been applied to other computer vision tasks, including object detection [4, 58, 54, 11], segmentation [45, 48], image enhancement [5, 50], image generation [30, 7], video processing [52, 57], and vision-language tasks [28, 38, 8, 36, 22, 21, 56, 23].

Developing an efficient attention mechanism for high-resolution image encoding is the focus of this work. Our model is inspired by the efficient attention mechanisms developed for Transformers, most of which are for NLP tasks. These mechanisms can be grouped into four categories. The first is the sparse attention mechanism, including content-independent sparsity [30, 9, 31, 15] and content-dependent sparsity [18, 35, 40, 55]. Axial Transformer [15] and Image Transformer [30] are among few sparsity-based efficient attentions that are developed for image generation. The second is the memory-based mechanism, including Compressive Transformers [32] and Set Transformer [20]. These models use some extra global tokens as static memory and allow all the other tokens to attend only to those global tokens. The third is the low-rank based mechanism. For example the Linformer [46] projects the input key-value pairs into a smaller chunk, and performs cross-attention between the queries and the projected key-value pairs. The fourth is the (generalized) kernel-based mechanism, including Performer[10] and Linear Transformers [17]. Many models utilize hybrid attention mechanisms. For example, Longformer[3], BigBird [51] and ETC [1] combine the sparsity and memory mechanisms; Synthesizers [39] combines the sparsity and low-rank mechanisms. We refer to [42] and [41] for a comprehensive survey and benchmarks.

In this paper, we developed a 2-D version of Longformer[3], called Vision Longformer, which utilizes both the sparsity and memory mechanisms. Its conv-like sparsity mechanism is conceptually similar to the sparsity mechanism used in the Image Transformer [30].

The multi-scale vision Transformer architecture is another technique we use in our proposed high-resolution Vision Longformer. The hierarchical Transformers [29] for NLP contain two stages, with the first stage processing overlapping segments and the second stage using the embeddings of the CLS tokens from all segments as input. In our proposed Vision Longformer, size reduction is performed by the patch embedding at the beginning of each stage, by merging all tokens in a patch from previous stage into a single token at the current stage. We typically use 4 stages for our model since we have empirically verified that using 4 stages is better than using 2 or 3 stages, especially for object detection tasks. Informer[55] takes a similar stacked multi-stage approach to encoding long sequences, where the size reduction between stages is achieved by max-pooling.

Pyramid Vision Transformer (PVT) [47], Swin Transformer [26] and HanoNet [44] are concurrent works of ours. All these works use a multi-scale architecture where multiple (slightly modified) ViTs are stacked. The authors of PVT propose the spatial-reduction attention (SRA) to alleviate the cost increase in self-attention layers. However, the computation and memory complexity of PVT still increases quartically w.r.t. resolution (with a much smaller constant). Swin Transformer [26] and HanoNet [44] utilizes similar local attention mechanism as our Vision Longformer.
3. Multi-Scale Stacked Vision Transformers

3.1. Multi-Scale Model Architecture

Efficient ViT (E-ViT). As shown in Figure 1 (Bottom), we improve the encoding efficiency of vision Transformer by making the following modifications. The modified ViT is referred to as Efficient ViT (E-ViT). (1) We add a Layer Normalization (LayerNorm) after the patch embedding. (2) We define a number of global tokens, including the CLS token. Correspondingly, the tokens associated with image and feature patches are referred to as local tokens afterwards. (3) We replace the vanilla full self-attention with an efficient attention mechanism, denoted by \( a \), which will be described in detail in Sections 3.2 and 3.3. (4) We use either an Absolute 2-D Positional Embedding (APE for short, separately encoding \( x \) and \( y \) coordinates and concatenating them) or a Relative Positional Bias (RPB for short) to replace the original absolute 1-D positional embedding.

Except for attention \( a \), E-ViT has the following architecture parameters inherited from the vanilla ViT: input patch size \( p \), number of attention blocks \( n \), hidden dimension \( d \) and number of heads \( h \), denoted as E-ViT(\( a \times n/p ; h, d, n_g \)) using the full attention mechanism (i.e., \( a = \text{full} \)) and one global token (i.e., the CLS token with \( n_g = 1 \)), the deficient E-ViT(\( \text{full} \times 12/16 ; h, d, 1 \)) models still achieve better ImageNet classification performance than the baseline ViT for both tiny (\( h = 3, d = 192 \)) and small (\( h = 6, d = 384 \)) model sizes, as shown in Table 2. The performance gain is attributed to the added LayerNorm, as we show in the Supplementary.

Mathematically, an E-ViT(\( a \times n/p ; h, d, n_g \)) encoding module can be written as:

\[
\begin{align*}
  z_0 &= [x_1^g; \ldots; x_n^g; LN(x_1^p E); \ldots; LN(x_n^p E)] + E_{\text{ops}}, \\
  z'_k &= \text{MSA}_a(LN(z_{k-1})), \quad k = 1, \ldots, n \\
  z_k &= \text{MLP}(LN(z_k^g)) + z'_k, \quad k = 1, \ldots, n,
\end{align*}
\]

where \( LN \) is the added Layer Normalization after the patch embedding \( E \), \( \text{MSA}_a \) is the multi-head self-attention with attention type \( a \), and \( \text{MLP} \) is the feed-forward block in a standard Transformer. When the absolute 2-D positional embedding is used, \( E_{\text{ops}} \in \mathbb{R}^{(n_l+n_g) \times d} \) contains the 2-D positional embedding of \( n_l \) local tokens and the 1-D positional embedding of \( n_g \) global tokens. When the relative positional bias is used, \( E_{\text{ops}} = 0 \) and the per-head relative positional bias is directly added to the attention scores in the \( \text{MSA}_a \) modules, as in Equation (4).

Stack multiple E-ViT modules as multi-scale vision Transformers. As illustrated in Figure 1 (Top), a multi-scale Vision Transformer is built by stacking multiple E-ViT modules. In the following, we describe several design choices we have made when building the multi-scale ViT.

What are the patch size and hidden dimension at each stage? As required in object detection and human pose estimation, for models with 4-scale feature maps, the first feature map needs to down-sample the image by 4 and thus stage 1 can be written as E-ViT(\( a_1 \times n_1/4 ; h_1, d_1, n_g,1 \)). We typically use only one attention block, i.e., \( n_1 = 1 \). The first stage generates the highest-resolution feature map, which consumes lots of memory, as shown in Table 2. We also construct several 3-stage models, whose first stage patch size is 8. For later stages, the patch sizes are set to 2, which downsizes the feature map resolution by 2. Following the practice in ResNet, we increase the hidden dimension twice when downsizing the feature map resolution by
2. We list a few representative model configurations in Table 1. Different attention types (α) have different choices of number of global tokens n_g. But they share the same model configurations. Thus we do not specify α and n_g in Table 1. Please refer to the Supplementary for the complete list of model configurations used in this paper.

<table>
<thead>
<tr>
<th>Size</th>
<th>Stage1 n,p,h,d</th>
<th>Stage2 n,p,h,d</th>
<th>Stage3 n,p,h,d</th>
<th>Stage4 n,p,h,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny</td>
<td>1.4,1.48</td>
<td>1.2,3.96</td>
<td>9.2,3.192</td>
<td>1.2,6,384</td>
</tr>
<tr>
<td>Small</td>
<td>1.4,3.96</td>
<td>2.2,3.192</td>
<td>8.2,6,384</td>
<td>1.2,12,768</td>
</tr>
<tr>
<td>Medium</td>
<td>1.4,3.96</td>
<td>4.2,3.192</td>
<td>16.2,6,384</td>
<td>1.2,12,768</td>
</tr>
<tr>
<td>Base</td>
<td>1.4,3.96</td>
<td>8.2,3.192</td>
<td>24.2,6,384</td>
<td>1.2,12,768</td>
</tr>
</tbody>
</table>

Table 1. Model architecture for multi-scale stacked ViTs. Architecture parameters for each E-ViT stage E-ViT(α × n/p ; h, d): number of attention blocks n, input patch size p, number of heads h and hidden dimension d. See the meaning of these parameters in Figure 1 (Bottom).

How to connect global tokens between consecutive stages? The choice varies at different stages and among different tasks. For the tasks in this paper, e.g., classification, object detection, instance segmentation, we simply discard the global tokens and only reshape the local tokens as the input for next stage. In this choice, global tokens only plays a role of an efficient way to globally communicate between distant local tokens, or can be viewed as a form of global memory. These global tokens are useful in vision-language tasks, in which the text tokens serve as the global tokens and will be shared across stages.

Should we use the average-pooled layer-normed features or the LayerNormed CLS token’s feature for image classification? The choice makes no difference for flat models. But the average-pooled feature performs better than the CLS feature for multi-scale models, especially for the multi-scale models with only one attention block in the last stage (including all models in Table 1). Please refer to the Supplementary for an ablation study.

As reported in Table 2, the multi-scale models outperform the flat models even in ImageNet classification, demonstrating the importance of multi-scale structure. However, the full self-attention mechanism suffers from the quartic computation/memory complexity w.r.t. the resolution of feature maps, as shown in Table 2. Thus, it is impossible to train 4-stage multi-scale ViTs with full attention using the same setting (batch size and hardware) in DeiT.

3.2. Vision Longformer: A “Local Attention + Global Memory” Mechanism

We propose to use the “local attention + global memory” efficient mechanism, as illustrated in Figure 2 (Left), to reduce the computational and memory cost in the E-ViT module. The 2-D Vision Longformer is an extension of the 1-D Vision Longformer [3] originally developed for NLP tasks. We add n_g global tokens (including the CLS token) that are allowed to attend to all tokens, serving as global memory. Local tokens are allowed to attend to only global tokens and their local 2-D neighbors within a window size. After all, there are four components in this “local attention + global memory” mechanism, namely global-to-global, local-to-global, global-to-local, and local-to-local, as illustrated in Figure 2 (Left). In Equation (2), a Multi-head Self-Attention (MSA) block with the Vision Longformer attention mechanism is denoted as MSA_{ViL}, i.e., α = ViL in Equation (2).

Relative positional bias for Vision Longformer. Following [33, 2, 26], we add a relative positional bias B to each head when computing the attention score:

\[
\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d} + B)V, \tag{4}
\]
where $Q, K, V$ are the query, key and value matrices and $d$ is the query/key dimension. This relative positional bias makes Vision Longformer translational invariant, which is a desired property for vision models. We observe significant improvements over the absolute 2-D positional embedding, as shown in Table 2 for ImageNet classification and Section 4.4 for COCO object detection.

**Theoretical complexity.** Given the numbers of global and local tokens, denoted by $n_g$ and $n_l$ respectively, and local attention window size $w$, the memory complexity of the $\text{MSA}_{\text{VL}}$ block is $O(n_g(n_g + n_l) + n_l w^2)$. Although [3] points out that separating the attention parameters for global and local tokens is useful, we do not observe obvious gain in our experiments and thus simply let them share the same set of attention parameters. We empirically set the window size $w$ to 15 for all E-ViT stages, which makes our model comparable with the global attention window size 14 of ViT16 acted on $224 \times 224$ images. With such a window size, only attentions in the first two stages (in 4-stage multi-scale ViTs) are local. The attentions in the later two stages are equivalent to full attention. In our experiments, we find that it is sufficient to use only one global token ($n_g = 1$) for ImageNet classification problems. So, the effective memory complexity of the $\text{MSA}_{\text{VL}}$ block is $O((15^2 + 1)n_l)$, which is linear w.r.t. the number of tokens.

**Superior performance in ImageNet classification.** Results in Table 2 show that in comparison with the full attention models, the proposed multi-scale Vision Longformer achieves a similar or even better performance, while saving significant memory and computation cost. The memory saving is significant for feature maps with resolution $56 \times 56$ (i.e., the feature maps in the first stage of a 4-stage multi-scale model). The savings are even more significant for higher resolution feature maps. This makes Vision Longformer scalable to high-resolution vision tasks, such as object detection and segmentation. When equipped with relative positional bias, Vision Longformer outperforms the full attention models with absolute positional embedding. This indicates that the “local attention + global memory” mechanism is a good inductive bias for vision Transformers.

**Three implementations of Vision Longformer and its random-shifting training strategy.** Vision Longformer is conceptually similar to conv-like local attention. We have implemented Vision Longformer in three ways: (1) using Pytorch’s unfold function (nn.unfold or tensor.unfold), (2) using a customized CUDA kernel and (3) using a sliding chunk approach. The unfold implementation is simple but very slow, i.e., 24 times slower than full attention on $40 \times 40 \times 768$ feature map. The implementation using the customized CUDA kernel is about 20% faster than the full attention in the same setting, while achieving the theoretical memory complexity. The sliding-chunk approach is the fastest, which is 60% faster than the full attention with a cost of consuming slightly more memory than the theoretical complexity. With the sliding chunk implementation, we also propose a random-shifting training strategy for Vision Longformer, which further improves the training speed and memory consumption during training. Please refer to the Supplementary for details of these implementations and the random-shifting training strategy.

### 3.3. Other Efficient Attention Mechanisms

We compare Vision Longformer with the following alternative choices of efficient attention methods. We put detailed descriptions of these methods and their experimental setup in the Supplementary.

**Pure global memory ($a = \text{global}$).** In Vision Longformer, see Figure 2 (Left), if we remove the local-to-local attention, then we obtain the pure global memory attention mechanism (called Global Attention hereafter). Its memory complexity is $O(n_g(n_g + n_l))$, which is also linear w.r.t. $n_l$. However, for this pure global memory attention, $n_g$ has to be much larger than 1. We gradually increase $n_g$ (by 2 each time) and its performance gets nearly saturated at 128. Therefore, $n_g = 128$ is the default for this Global attention.

**Linformer[46] ($a = \text{LIN}$)** projects the $n_l \times d$ dimensional keys and values to $K \times d$ dimensions using additional projection layers, where $K \ll n_l$. Then the $n_l$ queries only attend to these projected $K$ key-value pairs. The memory complexity of Linformer is $O(K n_l)$. We gradually increase $K$ and its performance gets nearly saturated at 256. Therefore, we set $K = 256$ for this Linformer attention, which is the same with its recommended value. Notice that Linformer’s projection layer (of dimension $K \times n_l$) is specific to the current $n_l$, and cannot be transferred to higher-resolution tasks that have a different $n_l$.

**Spatial Reduction Attention (SRA) [47] ($a = \text{SRA}$)** is similar to Linformer, but uses a convolution layer with kernel size $R$ and stride $R$ to project the key-value pairs, hence resulting in $n_l/R^2$ compressed key-value pairs. Therefore, the memory complexity of SRA is $O(n_l^2/R^2)$, which is still quadratic w.r.t. $n_l$ but with a much smaller constant $1/R^2$. When transferring the ImageNet-pretrained SRA-models to high-resolution tasks, SRA still suffers from the quartic computation/memory blow-up w.r.t. the feature map resolution. Pyramid Vision Transformer [47] uses this SRA to build multi-scale vision transformer backbones, with different spatial reduction ratios ($R_1 = 8, R_2 = 4, R_3 = 2, R_4 = 1$) for each stage. With this PVT’s setting, the key and value feature maps at all stages are essentially with resolution $H/32 \times W/32$.

**Performer [10] ($a = \text{performer}$)** uses random kernels to approximate the Softmax computation in MSA, and achieves a linear computation/memory complexity with respect to $n_l$ and the number of random features. We use the
default 256 orthogonal random features (OR) for Performer, and provide other details in the Supplementary.

<table>
<thead>
<tr>
<th>Attention-type</th>
<th>Tiny-size</th>
<th>Small-size</th>
<th>Trans2Det</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>76.06</td>
<td>81.91</td>
<td></td>
</tr>
<tr>
<td>VIL</td>
<td>76.18</td>
<td>81.99</td>
<td>✓</td>
</tr>
<tr>
<td>Global</td>
<td>71.52</td>
<td>78.97</td>
<td>✓</td>
</tr>
<tr>
<td>Linformer [46]</td>
<td>74.71</td>
<td>80.98</td>
<td></td>
</tr>
<tr>
<td>SRA/64[47]</td>
<td>69.08</td>
<td>76.37</td>
<td>✓</td>
</tr>
<tr>
<td>SRA/32[47]</td>
<td>73.22</td>
<td>79.9</td>
<td>–</td>
</tr>
<tr>
<td>Performer</td>
<td>71.12</td>
<td>78.72</td>
<td>✓</td>
</tr>
<tr>
<td>Par-Global</td>
<td>75.32</td>
<td>81.46</td>
<td>–</td>
</tr>
<tr>
<td>Par-Linformer</td>
<td>75.56</td>
<td>81.79</td>
<td>×</td>
</tr>
<tr>
<td>Par-SRA/32</td>
<td>75.2</td>
<td>81.61</td>
<td>–</td>
</tr>
<tr>
<td>Par-Performer</td>
<td>75.34</td>
<td>81.72</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 3. Overall comparison of different attention mechanisms with 2D absolute positional embedding on ImageNet classification top-1 accuracy (%), with input size 224, “Par-xformer” indicates multi-scale ViTs with multiple attention types: the first two stages utilize the “xformer” efficient attention and the last two stages still use full attention. In the “Trans2Det” columns, ✓ indicates that the ImageNet-pre-trained model can be used to initialize detection backbones, × means not. – means that it can be transferred, but the corresponding detection models consumes prohibitively large memory due to the need of using high resolution feature maps. SRA/32 downsizes key/value feature maps with the same schedule in PVT[47], while SRA/64 downsizes more aggressively to make the memory manageable for downstream high-resolution tasks.

**Compare Vision Longformer with other attention mechanisms.** On the ImageNet classification task in Table 3, all efficient attention mechanisms abe above show a large performance gap from Vision Longformer. Linformer performs very competitively. Global attention and Performer have a similar performance with the DeiT model (72.2 for tiny and 79.8 for small). We use spatial reduction ratios 16, 8, 4, 2 from stage1 to stage4 for the multi-scale SRA model, which is different from the reduction ratios 8, 4, 2, 1 in PVT [47]. This more aggressive spatial reduction makes the classification performance worse in Table 3, but makes the memory cost manageable when transfer to detection tasks for input image size 800 × 1333. For a more complete comparison of these models, including model parameters, FLOPs and memory usage, please refer to the Supplementary.

**Why is Longformer better?** One possible reason is that the conv-like sparsity is a good inductive bias for vision transformers, compared with other attention mechanisms. This is supported by the visualization of the attention maps of the pretrained DeiT models [43] in our Supplementary. Another explanation is that Vision Longformer keeps the key and value feature maps high resolution. However, low resolution-based attention mechanisms like Linformer and SRA and pure global attention lose the high-resolution information in the key and value feature maps.

**Mixed attention mechanisms (Partial X-former) for classification tasks.** For classification tasks with 224 × 224 image size as input, the feature map size at Stage3 in multiscale ViTs is 14 × 14. This is the same as the feature map size in ViT and DeiT, and best suits for full attention. A natural choice is to use efficient attention in the first two stages (with high-resolution feature map but with small number of blocks) and to use full attention in the last two stages. Multi-scale ViTs with this mixed attention mechanisms are called “Partial X-former”. We also report these Partial X-formers’ performance in Table 3. All these Partial X-formers perform well on ImageNet classification, with very little (even no) gap between Full Attention and Vision Longformer. These Partial X-forms achieve very good accuracy-efficiency performance for low-resolution classification tasks. We do not have “Partial ViL” for classification because ViL’s window size is 15, and thus its attention mechanism in the last two stages is equivalent to the full attention.

### 3.4. Transfer to High-resolution Vision Tasks

Similar to the transfer-ability of ImageNet-pretrained CNN weights to downstream high-resolution tasks, such as object detection and segmentation, multi-scale Vision Longformer pretrained on ImageNet can be transferred to such high-resolution tasks, as we will show in Section 4.3.

However, Linformer is not transferable because the weights of the linear projection layer is specific to a resolution. The Partial X-formers and Multi-scale ViT with full attention are not transferable due to its prohibitively large memory usage after transferred to high-resolution tasks. In Table 7, we also show the superior performance of Vision Longformer over other attention mechanisms, on the object detection and segmentation tasks.

### 4. Experiments

In this section, we show the final performance of Vision Longformer (short for ViL) on ImageNet classification in Section 4.1 & 4.2 and downstream high-resolution detection tasks in Section 4.3. We follow the DeiT training configuration for ImageNet classification training, and use the standard “×1” and “×3+MS” training schedules with the “AdamW” optimizer for detection tasks. We refer to the Supplementary for detailed experimental settings.

#### 4.1. ImageNet Classification

Following the setting in DeiT [43], we train multi-scale ViLs purely on ImageNet1K. In Table 4, we report our results and compare with ResNets[14], ViT [12], DeiT [43] and PVT [47]. Our models outperform other models in the same scale by a large margin. We again confirm that the relative positional bias (RPB) outperforms the absolute 2-D positional embedding (APE) on Vision Longformer.
4.2. ImageNet-21K Pretraining

When trained purely on ImageNet-1K, the performance gain from ViL-Medium to ViL-Base is very marginal. This is consistent with the observation in ViT [12]: large pure transformer based models can be trained well only when training data is sufficient. Therefore, we conducted experiments in which ViL-Medium/Base models are first pretrained on ImageNet-21k with image size $384^2$ and fine-tuned on ImageNet-1K with image size $384^2$. For ViT models on image size $384^2$, there are in total $24 \times 24$ tokens with full attention. For ViL models on image size $384^2$, we set the window sizes to be $(13, 17, 25, 25)$ from Stage1 to Stage4. Therefore, in the last two stages, the ViL models’ attention is still equivalent to full attention.

As shown in Table 5, the performance gets significantly boosted after ImageNet-21K pretraining for both ViL medium and base models. We want to point out that the performance of ViL-Medium model has surpassed that of ViT-Base/16, ViT-Large/16 and BiT-152x4-M, in the ImageNet-21K pretraining setting. The performance of ViL-Base models are even better. This shows the superior performance and parameter efficiency of ViL models.

4.3. Detection Tasks

We apply our ViL to two representative object detection pipelines including RetinaNet [24] and Mask-RCNN [13].

<table>
<thead>
<tr>
<th>Model</th>
<th>#Params (M)</th>
<th>GFLOPs</th>
<th>Top-1 (%)</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>R18</td>
<td>11.7</td>
<td>1.8</td>
<td>69.8</td>
<td>4367</td>
</tr>
<tr>
<td>DeiT-Tiny/16[43]</td>
<td>5.7</td>
<td>1.3</td>
<td>72.2</td>
<td>2532</td>
</tr>
<tr>
<td>PVT-Tiny[47]</td>
<td>13.2</td>
<td>1.9</td>
<td>75.1</td>
<td>1489</td>
</tr>
<tr>
<td>ViL-Tiny-APE</td>
<td>6.7</td>
<td>1.3</td>
<td>76.3</td>
<td>949</td>
</tr>
<tr>
<td>ViL-Tiny-RPB</td>
<td>6.7</td>
<td>1.3</td>
<td>76.7</td>
<td>901</td>
</tr>
<tr>
<td>R50</td>
<td>25.6</td>
<td>4.1</td>
<td>78.5</td>
<td>1206</td>
</tr>
<tr>
<td>DeiT-Small/16[43]</td>
<td>22.1</td>
<td>4.6</td>
<td>79.9</td>
<td>939</td>
</tr>
<tr>
<td>PVT-Small[47]</td>
<td>24.5</td>
<td>3.8</td>
<td>80.9</td>
<td>810</td>
</tr>
<tr>
<td>Swin-Tiny[26]</td>
<td>28</td>
<td>4.5</td>
<td>81.2</td>
<td>713</td>
</tr>
<tr>
<td>ViL-Small-APE</td>
<td>24.6</td>
<td>4.9</td>
<td>82.0</td>
<td>366</td>
</tr>
<tr>
<td>ViL-Small-RPB</td>
<td>24.6</td>
<td>4.9</td>
<td>82.4</td>
<td>350</td>
</tr>
<tr>
<td>R101</td>
<td>44.7</td>
<td>7.9</td>
<td>79.8</td>
<td>725</td>
</tr>
<tr>
<td>PVT-Medium[47]</td>
<td>44.2</td>
<td>6.7</td>
<td>81.2</td>
<td>515</td>
</tr>
<tr>
<td>Swin-Small[26]</td>
<td>50</td>
<td>8.7</td>
<td>83.2</td>
<td>420</td>
</tr>
<tr>
<td>ViL-Medium-APE</td>
<td>39.7</td>
<td>8.7</td>
<td>83.3</td>
<td>235</td>
</tr>
<tr>
<td>ViL-Medium-RPB</td>
<td>39.7</td>
<td>8.7</td>
<td>83.5</td>
<td>222</td>
</tr>
<tr>
<td>X101-64x4d</td>
<td>83.5</td>
<td>15.6</td>
<td>81.5</td>
<td>302</td>
</tr>
<tr>
<td>ViT-Base/16[12]</td>
<td>86.6</td>
<td>17.6</td>
<td>77.9</td>
<td>291</td>
</tr>
<tr>
<td>DeiT-Base/16[43]</td>
<td>86.6</td>
<td>17.6</td>
<td>81.8</td>
<td>291</td>
</tr>
<tr>
<td>PVT-Large[47]</td>
<td>61.4</td>
<td>9.8</td>
<td>81.7</td>
<td>368</td>
</tr>
<tr>
<td>Swin-Big[26]</td>
<td>88</td>
<td>15.8</td>
<td>83.5</td>
<td>282</td>
</tr>
<tr>
<td>ViL-Base-APE</td>
<td>55.7</td>
<td>13.4</td>
<td>83.2</td>
<td>149</td>
</tr>
<tr>
<td>ViL-Base-RPB</td>
<td>55.7</td>
<td>13.4</td>
<td>83.7</td>
<td>144</td>
</tr>
</tbody>
</table>

Table 4. Number of parameters, FLOPS and ImageNet accuracy. Trained on ImageNet-1K with image size 224. Our ViL models are with gray background. See Table 1 for detailed model configs.

We follow the conventional setting to use our Vision Longformer as the backbone to generate feature maps for both detection pipelines. Similar to [47], we extract the features from all four scales and then feed them to the detection and/or instance segmentation head. To adapt the learned relative positional bias to the higher image resolution in detection, we perform bilinear interpolation on it prior to the training. In our experiments, all models are evaluated on COCO dataset [25], with 118k images for training and 5k images for evaluation. We report the results for both $1 \times$ and $3 \times+MS$ training schedules, and compare them with two backbone architectures: ResNet [14] and PVT [13].

As shown in Table 6, our ViL achieves significantly better performance than the ResNet and PVT architecture, for both the RetinaNet and Mask R-CNN pipelines. The improvements are uniform over all model sizes (tiny, small, medium, base) and over all object scales ($AP_S$, $AP_M$, $AP_L$). The improvement is so large that ViL-Tiny with “3x+MS” schedule already outperforms the ResNeXt101-64x4d and the PVT-Large models. When compared with the concurrent Swin Transformer [26], our model also outperforms it with fewer parameter and FLOPs. More specifically, our ViL-Small achieves $47.1 \ AP_b$ with 45M parameters, while Swin-Tiny achieves $46.0 \ AP_b$ with 48M parameters. These consistent and significant improvements with both RetinaNet and Mask R-CNN demonstrate the promise of our proposed ViL when using it as the image encoder for high-resolution dense object detection tasks.

4.4. Ablation Study for Detection Tasks

Compare with other efficient attention mechanisms. Similar to Sec 4.4, we study SRA [47], Global Transformer and Performer and their corresponding partial version with Mask R-CNN pipeline (trained with the $1 \times$ schedule). As we can see in Table 7, when efficient attention mechanisms are used in all stages, ViL achieves much better performance than the other three mechanisms. Specifically, our ViL achieves $42.9 \ AP_b$ while the other three are all around
Table 6. Object detection and instance segmentation performance on the COCO val2017. The numbers before and after “/” at column 2 and 3 are the model size and complexity for RetinaNet and Mask R-CNN, respectively. The FLOPs (G) are measured at resolution 800 × 1333. “–” means data publicly unavailable. Our ViL-Tiny and ViL-Small models are pre-trained on ImageNet-1K, our ViL-Medium and ViL-Base models are pre-trained on ImageNet-21k. ViL results are highlighted with gray background.

Table 7. Comparing different efficient attention mechanisms for object detection with Mask R-CNN. All use small model size and absolute 2-D positional embedding (APE) for fair comparison. Run-time memory cost when training each model is also reported.

36.0 $AP^b$. When efficient attention mechanisms are only used in the first two stages (Par-Xformer), the gaps between different mechanisms shrink to around 1.0 point while our ViL still outperform all others. Moreover, the ViL model outperforms the partial models of all other attention mechanisms and has a very small gap (0.4 $AP^b$) from the Partial-ViL model. These results show that the “local attention + global memory” mechanism in Vision Longformer can retain the good performance of the full attention mechanism in ViT, and that it is a clear better choice than other efficient attention mechanisms for high-resolution vision tasks.

**The effects of window size and number of global tokens** are not obvious in ImageNet classification, as long as the last two stages use full attention. For different window sizes in [9, 15, 21] and different number of global tokens in [0, 1, 2, 4, 8], the final top-1 accuracy differs by at most 0.2 for ViL-Small models. Meanwhile, their effects are significant in high-resolution tasks, where ViL models use local attention in all stages. In Figure 3, we report their effects in COCO object detection with Mask R-CNN. We notice that the window size plays a crucial role and the default window size 15 gives the best performance. Smaller window sizes lead to serious performance drop. As shown in Figure 3 (Right), as long as there is one global token, adding more global tokens does not improve the performance any more.

5. Conclusions

In this paper, we have presented a new Vision Transformer (ViT) architecture Multi-Scale Vision Longformer to address the computational and memory efficiency that prevents the vanilla ViT model from applying to vision tasks requiring high-resolution feature maps. Through a comprehensive empirical study on both classification and detection tasks, we demonstrate that the attention mechanism of ViL outperforms alternative efficient attention mechanisms.
References


