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

Pengchuan Zhang[1] Xiyang Dai[1†] Jianwei Yang[1†] Bin Xiao[1] Lu Yuan[1]
Lei Zhang[2] Jianfeng Gao[1]
[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 images[1], 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

† indicates equal contributions.

[1]In this paper, encoding a high-resolution image means generating high-resolution feature maps for high-resolution images.

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 $4\times$ 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.

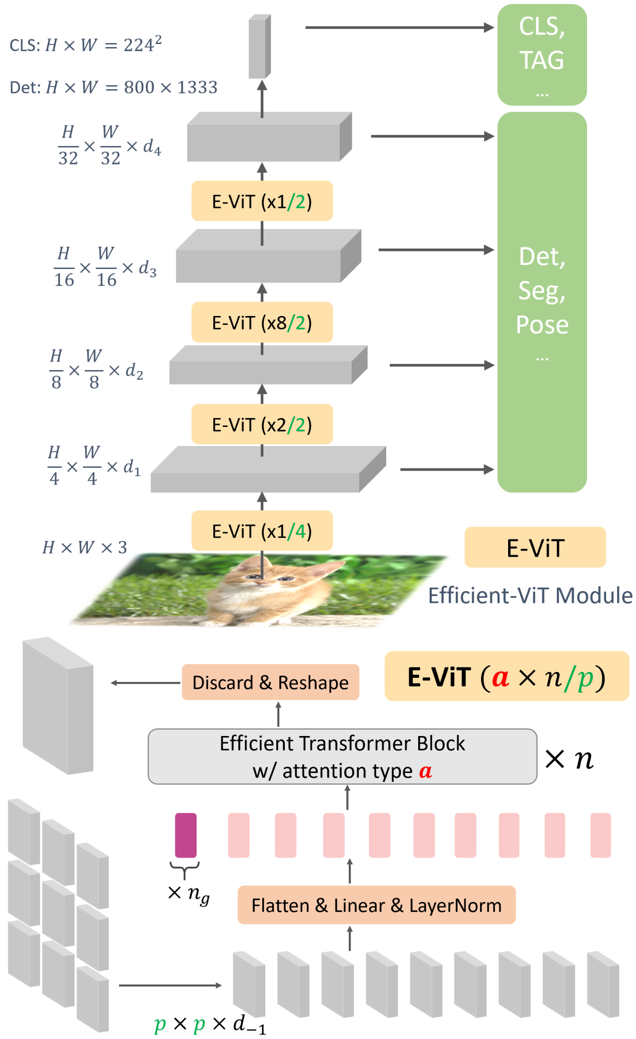

Figure 1. A Multi-scale vision Transformers (bottom) by stacking 4 E-ViT modules (Top). An E-ViT ($a \times n$ / p) module is a ViT encoder with an efficient attention mechanism a, n efficient transformer blocks, input patch size p. We add a LayerNorm after the patch embedding. We add n_g extra global tokens, as a form of global memory, and simply throw them away when going to the next stage.

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 = full) and one global token (i.e., the CLS token with $n_g = 1$), the deficient E-ViT(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:

$$z_0 = [x_g^1; \ldots; x_g^{n_g}; LN(x_p^1 E); \ldots; LN(x_p^{n_l} E)] + E_{ops}, \quad (1)$$

$$z'_k = MSA_a(LN(z_{k-1})) + z_{k-1}, \quad k = 1, .., n \quad (2)$$

$$z_k = MLP(LN(z'_k)) + z'_k, \quad k = 1, .., n, \quad (3)$$

where LN is the added Layer Normalization after the patch embedding E, MSA_a is the multi-head self-attention with attention type a, and MLP is the feed-forward block in a standard Transformer. When the absolute 2-D positional embedding is used, $E_{ops} \in \mathbb{R}^{(n_l+n_g)\times d}$ 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, $E_{ops} = 0$ and the per-head relative positional bias is directly added to the attention scores in the 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 (a) have different choices of number of global tokens n_g. But they share the same model configurations. Thus we do not specify a and n_g in Table 1. Please refer to the Supplementary for the complete list of model configurations used in this paper,

Size	Stage1 n,p,h,d	Stage2 n,p,h,d	Stage3 n,p,h,d	Stage4 n,p,h,d
Tiny	1,4,1,48	1,2,3,96	9,2,3,192	1,2,6,384
Small	1,4,3,96	2,2,3,192	8,2,6,384	1,2,12,768
Medium	1,4,3,96	4,2,3,192	16,2,6,384	1,2,12,768
Base	1,4,3,96	8,2,3,192	24,2,6,384	1,2,12,768

Table 1. Model architecture for multi-scale stacked ViTs. Architecture parameters for each E-ViT stage E-ViT($a \times n/p\ ; h, d$): 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

Model	#Params (M)	FLOPs (G)	Memory (M)	Top-1 (%)
DeiT-Small / 16 [43]	22.1	4.6	67.1	79.9
E-ViT(full/16)-APE	22.1	4.6	67.1	80.4/80.7
Full-Small-APE	24.63	6.95	488.3	81.9
ViL-Small-APE	24.63	4.86	116.8	82.0
ViL-Small-RPB	24.65	4.86	131.6	82.4

Table 2. Flat vs Multi-scale Models: Number of paramers, FLOPS, memory per image (with Pytorch Automatic Mixed Precision enabled), and ImageNet accuracy with image size 224. Since all our multi-scale models use average-pooled feature from the last stage for classification, we report Top-1 accuracy of "E-ViT(full/16)-APE" both with the CLS feature (first) and with the average-pooled feature (second). "Full-Small-APE" stands for a small-size multiscale ViT with a = full attention and with Absolute 2-D Positional Embedding (APE). The multi-scale models consistently outperform the flat models, but the memory usage of full attention quickly blows up when high-resolution blocks are introduced. The Vision Longformer ("ViL-") saves FLOPs and memory, without performance drop. Using relative positional bias ("ViL-***-RPB") further improves the performance.

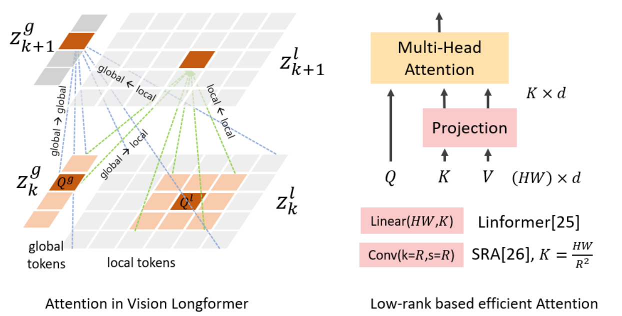

Figure 2. Left: the Vision Longformer attention mechanism. Right: the Low-rank based attention mechanism. Without "local→local" attentions in Vision Longformer, we get the Global Former. With a linear layer as the projection, we get Linformer[46]. With a conv layer with equal kernel size and stride, we get Spatial Reduction Attention (SRA)[47].

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., a = 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:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (4)$$

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 MSA_{ViL} block is $\mathcal{O}(n_g(n_g + n_l) + n_l w^2)$. 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 ViT/16 acted on 224×224 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[2] 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 MSA_{ViL} block is $\mathcal{O}((15^2 + 1)n_l)$, 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 56×56 (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.

[2]Equivalent in our sliding-chunk implementation (the default choice).

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 = 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 $\mathcal{O}(n_g(n_g + n_l))$, 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 = 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 $\mathcal{O}(Kn_l)$. 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 = 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 $\mathcal{O}(n_l^2/R^2)$, 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 = 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.

Attention-type	Tiny-size	Small-size	Trans2Det
Full	76.06	81.91	–
ViL	76.18	81.99	✓
Global	71.52	78.97	✓
Linformer [46]	74.71	80.98	✗
SRA/64[47]	69.08	76.37	✓
SRA/32[47]	73.22	79.9	–
Performer	71.12	78.72	✓
Par-Global	75.32	81.46	–
Par-Linformer	75.56	81.79	✗
Par-SRA/32	75.2	81.61	–
Par-Performer	75.34	81.72	–

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 manage-able for downstream high-resolution tasks.

Compare Vision Longformer with other attention mechanisms. On the ImageNet classification task in Table 3, all efficient attention mechanisms 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 mechanims 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 multi-scale 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 "$\times 1$" and "$\times 3$+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.

Model	#Params (M)	GFLOPs	Top-1 (%)	Throughput
R18	11.7	1.8	69.8	4367
DeiT-Tiny/16[43]	5.7	1.3	72.2	2532
PVT-Tiny[47]	13.2	1.9	75.1	1489
ViL-Tiny-APE	6.7	1.3	76.3	949
ViL-Tiny-RPB	6.7	1.3	76.7	901
R50	25.6	4.1	78.5	1206
DeiT-Small/16[43]	22.1	4.6	79.9	939
PVT-Small[47]	24.5	3.8	79.8	810
Swin-Tiny[26]	28	4.5	81.2	713
ViL-Small-APE	24.6	4.9	82.0	366
ViL-Small-RPB	24.6	4.9	82.4	350
R101	44.7	7.9	79.8	725
PVT-Medium[47]	44.2	6.7	81.2	515
Swin-Small[26]	50	8.7	83.2	420
ViL-Medium-APE	39.7	8.7	83.3	235
ViL-Medium-RPB	39.7	8.7	83.5	222
X101-64x4d	83.5	15.6	81.5	302
ViT-Base/16[12]	86.6	17.6	77.9	291
DeiT-Base/16[43]	86.6	17.6	81.8	291
PVT-Large[47]	61.4	9.8	81.7	368
Swin-Base[26]	88	15.4	83.5	282
ViL-Base-APE	55.7	13.4	83.2	149
ViL-Base-RPB	55.7	13.4	83.7	144

Table 4. Number of paramers, 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.

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 224^2 and finetuned on ImageNet-1K with image size 384^2. For ViT models on image size 384^2, there are in total 24×24 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 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].

Model	#Params (M)	No IN-21K		After IN-21K	
		GFLOPs	Top-1	GFLOPs	Top-1
ViT-Base/16[12]	86.6	17.6	77.9	49.3	84.0
ViT-Large/16[12]	307	61.6	76.5	191.1	85.2
BiT-152x4-M[19]	928	182	81.3	837	85.4
Swin-Base[26]	88	15.4	83.5	47.1	86.4
ViL-Medium-RPB	39.7	8.7	83.5	28.4	85.7
ViL-Base-RPB	55.7	13.4	83.7	43.7	86.2

Table 5. Trained purely on ImageNet-1K with image size 224 (No IN-21K). Pretained on ImageNet-21K with image size 224 and Finetuned on ImageNet-1K with image size 384 (After IN-21K), except BiT-M [19] fine-tuned with image size 480. Our ViL models are highlighted with gray background.

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

Backbone	#Params (M)	FLOPs (G)	RetinaNet 3x + MS schedule						Mask R-CNN 3x + MS schedule					
			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet18	21.3/31.2	190/207	35.4	53.9	37.6	19.5	38.2	46.8	36.9	57.1	40.0	33.6	53.9	35.7
PVT-Tiny[47]	23.0/32.9	–/–	39.4	59.8	42.0	25.5	42.0	52.1	39.8	62.2	43.0	37.4	59.3	39.9
ViL-Tiny-RPB	16.6/26.9	183/199	43.6	64.4	46.1	28.1	47.5	56.7	44.2	66.4	48.2	40.6	63.2	44.0
ResNet50	37.7/44.2	239/260	39.0	58.4	41.8	22.4	42.8	51.6	41.0	61.7	44.9	37.1	58.4	40.1
PVT-Small[47]	34.2/44.1	226/245	42.2	62.7	45.0	26.2	45.2	57.2	43.0	65.3	46.9	39.9	62.5	42.8
Swin-Tiny[26]	–/48	–/264	–	–	–	–	–	–	46.0	68.1	50.3	41.6	65.1	44.9
ViL-Small-RPB	35.7/45.0	255/277	45.9	66.6	49.0	30.9	49.3	59.9	47.1	68.7	51.5	42.7	65.9	46.2
ResNet101	56.7/63.2	315/336	40.9	60.1	44.0	23.7	45.0	53.8	42.8	63.2	47.1	38.5	60.1	41.3
ResNeXt101-32x4d	56.4/62.8	319/340	41.4	61.0	44.3	23.9	45.5	53.7	44.0	64.4	48.0	39.2	61.4	41.9
PVT-Medium[47]	53.9/63.9	283/302	43.2	63.8	46.1	27.3	46.3	58.9	44.2	66.0	48.2	40.5	63.1	43.5
Swin-Small[26]	–/69	–/354	–	–	–	–	–	–	48.5	70.2	53.5	43.3	67.3	46.6
ViL-Medium-RPB	50.8/60.1	330/352	47.9	68.8	51.3	32.4	51.9	61.8	48.9	70.3	54.0	44.2	67.9	47.7
ResNeXt101-64x4d	95.5/101.9	473/493	41.8	61.5	44.4	25.2	45.4	54.6	44.4	64.9	48.8	39.7	61.9	42.6
PVT-Large[47]	71.1/81.0	345/364	43.4	63.6	46.1	26.1	46.0	59.5	44.5	66.0	48.3	40.7	63.4	43.7
ViL-Base-RPB	66.7/76.1	421/439	48.6	69.4	52.2	34.1	52.5	61.9	49.6	70.7	54.6	44.5	68.3	48.0

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.

Attention	#Params (M)	AP^b	AP^m	FLOPs (G)	Memory (G)
SRA/64 [47]	73.3	36.4	34.6	224.1	7.1
SRA/32 [47]	51.5	39.9	37.3	268.3	13.6
Par-SRA/32	46.8	42.4	39.0	352.1	22.6
Global	45.2	34.8	33.4	226.4	7.6
Par-Global	45.1	42.5	39.2	326.5	20.1
Performer	45.0	36.1	34.3	251.5	8.4
Par-Performer	45.0	42.3	39.1	343.7	20.0
ViL	45.0	42.9	39.6	218.3	7.4
Par-ViL	45.0	43.3	39.8	326.8	19.5

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.

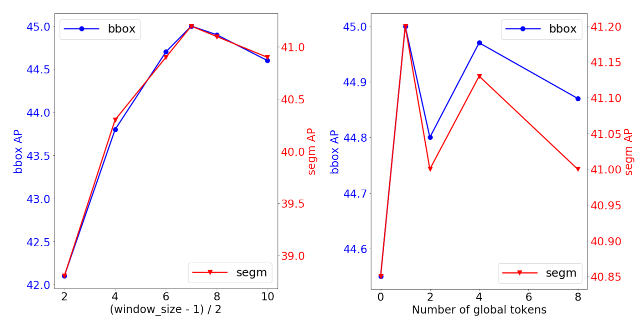

Figure 3. Effects of window size (Left) and number of global tokens (Right) in Vision Longformer for object detection with Mask R-CNN. All use the same ImageNet1K pre-trained checkpoint (ViL-Small-RPB in Table 4).

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 out-performs alternative efficient attention mechanisms.

References

[1] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. Etc: Encoding long and structured data in transformers. *arXiv preprint arXiv:2004.08483*, 2020.

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR, 2020.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019.

[9] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[10] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2020.

[11] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020.

[12] 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, 2020.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[15] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[16] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020.

[17] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[18] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

[19] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6, 2019.

[20] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[21] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[23] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[24] 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.

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

[26] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[29] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844, 2019.

[30] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[31] Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

[32] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[35] Aurko Roy*, Mohammad Taghi Saffar*, David Grangier, and Ashish Vaswani. Efficient content-based sparse attention with routing transformers, 2020.

[36] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[38] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019.

[39] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention in transformer models. *arXiv preprint arXiv:2005.00743*, 2020.

[40] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pages 9438–9447. PMLR, 2020.

[41] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[44] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. *arXiv preprint arXiv:2103.12731*, 2021.

[45] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020.

[46] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[47] 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. *arXiv preprint arXiv:2102.12122*, 2021.

[48] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020.

[49] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[50] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5800, 2020.

[51] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020.

[52] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020.

[53] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.

[54] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020.

[55] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *arXiv preprint arXiv:2012.07436*, 2020.

[56] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. *AAAI*, 2020.

[57] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[58] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.