Grafting Vision Transformers

Jongwoo Park[1], Kumara Kahatapitiya[1], Donghyun Kim[2], Shivchander Sudalairaj[2], Quanfu Fan[3]*, Michael S. Ryoo[1]

[1]Stony Brook University, [2]MIT-IBM Watson AI Lab, [3]Amazon

{jongwopark@, kkahatapitiy@, mryoo@}.cs.stonybrook.edu
{dkim, shiv.sr}@ibm.com
{quanfu}@amazon.com

Abstract

*Vision Transformers (ViTs) have recently become the state-of-the-art across many computer vision tasks. In contrast to convolutional networks (CNNs), ViTs enable global information sharing even within shallow layers of a network, i.e., among high-resolution features. However, this perk was later overlooked with the success of pyramid architectures such as Swin Transformer, which show better performance-complexity trade-offs. In this paper, we present a simple and efficient add-on component (termed **GrafT**) that considers global dependencies and multi-scale information throughout the network, in both high- and low-resolution features alike. It has the flexibility of branching out at arbitrary depths and shares most of the parameters and computations of the backbone. GrafT shows consistent gains over various well-known models which includes both hybrid and pure Transformer types, both homogeneous and pyramid structures, and various self-attention methods. In particular, it largely benefits mobile-size models by providing high-level semantics. On the ImageNet-1k dataset, GrafT delivers **+3.9**%, **+1.4**%, and **+1.9**% top-1 accuracy improvement to DeiT-T, Swin-T, and MobViT-XXS, respectively. Our code and models are at https://github.com/jongwoopark7978/Grafting-Vision-Transformer.*

1. Introduction

Self-attention mechanism in Transformers [1] has been widely-adopted in language domain for some time now. It can look into pairwise correlations between input sequences, learning long-range dependencies. More recently, following the seminal work in Vision Transformers (ViT) [9], the vision community has also started exploiting this property, showing state-of-the-art results on various tasks including classification, segmentation and detection, outperforming convolutional networks (CNNs) [2, 12, 15, 16, 28, 33]. Motivated by this success, many variants of vision Transformers (*e.g.,* DeiT [34], CrossViT [4], TNT [11]) emerged, inheriting the same homogeneous structure of ViT (i.e., a structure w/o downsampling). However, due to the quadratic complexity of attention, such a structure becomes expensive, especially for high-resolution inputs and does not benefit from the semantically-rich information present in multi-scale representations.

To address these shortcomings, Transformers with pyramid structures (i.e., structures w/ downsampling) such as Swin [25] were introduced with hierarchical downsampling and window-based attention, which can learn multi-scale representations at a computational complexity linear with input resolution. As a result, pyramid structures become more suited for tasks such as segmentation and detection. However, still, multiple scales arise deep into the network due to stage-wise downsampling, meaning that only the latter stages of the model may benefit from them. Thus, we pose the question: what if we can introduce multi-scale information even at the early stages of a Transformer without incurring a heavy computational burden? In particular, can CNN-based hybrid Transformers reap the benefits of GrafT as they heavily rely on the local information from CNNs?

Previous work has also looked into the direction above, both in CNNs [31] and in Transformers [3, 4]. However, models such as CrossViT requires carefully tuning the spatial ratio of two feature maps in two branches and RegionViT needs considerable modifications to handle multi-scale training. To mitigate these issues, in this paper, we propose a simple and efficient add-on component called **GrafT** (see

*Part of work was done while at MIT-IBM AI Watson Lab

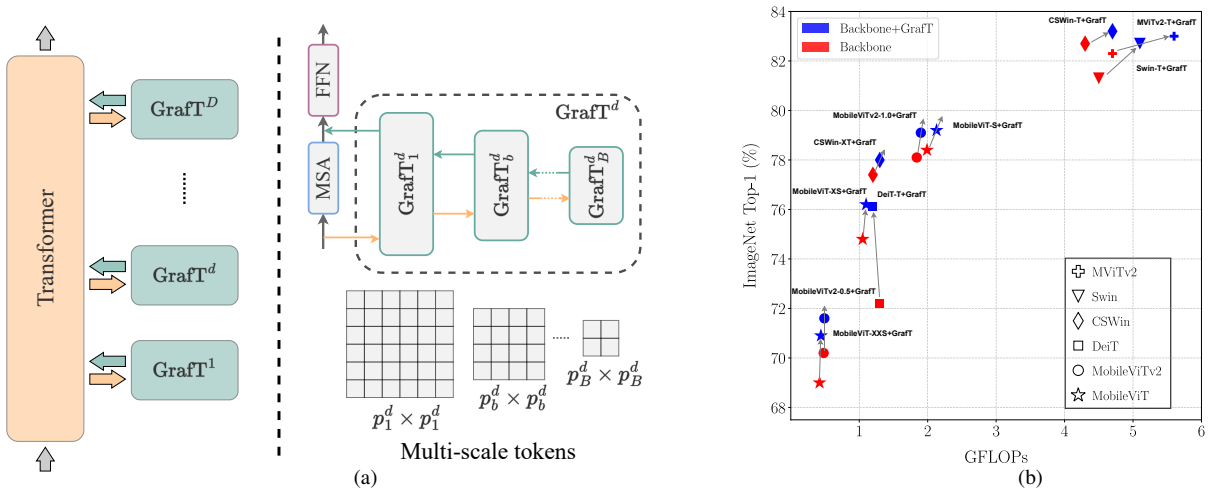

Figure 1. We introduce GrafT, an add-on component that makes use of global and multi-scale dependencies at arbitrary depths of a network. (a) An overview of how GrafT modules are branched-out (or grafted) from a backbone Transformer and generate multiple scales of features. (b) Performance-complexity trade-off of integrating GrafT in various backbones incorporating hybrid or pure architectures, homogeneous or pyramid structures, and various self-attention methods. GrafT shows consistent and considerable gains with a minimal increment in complexity. DeiT-T+GrafT uses fewer FLOPs because local attention is used to adopt GrafT as explained in 3.1

Figure 1-(a)). It can be easily adopted in existing hybrid or pure Transformers, homogeneous or pyramid architectures, and various self-attenion methods, enabling multi-scale features throughout a network (even in shallow layers) and showing consistent performance gains while being computationally lightweight. GrafT is applicable at any arbitrary layer of a network.

It consists of three main components: (1) a *left-right pathway* for downsampling, (2) a *right-left pathway* for upsampling, and (3) a *bottom-up connection* for information sharing at each scale. The left-right pathway uses a series of average pooling operations to create a set of multi-scale representations. For instance, if GrafT is attached to a layer with (56×56) resolution, it can create scales of (28×28), (14×14) and (7×7). We then process information at each scale with a *L-MSA* block, a local self-attention mechanism (*e.g.,* window-attention)— which becomes global-attention in the coarsest scale, as window-size becomes the same as the resolution. Next, the right-left pathway uses a series of learnable and window-based bi-linear interpolation (*W-Bilinear*) operations to generate high-resolution features by upsampling the low-resolution outputs of L-MSA— which contains global (or high-level) semantics extracted efficiently, at a lower resolution. Such upsampled features are merged with high-resolution features of the branch-to-the-left, which contain lower-level semantics, as also done in Feature Pyramid Networks [22]. Refer to Figure 2-(b) and Section 2.2 for more details.

GrafT is unique in the sense that it can extract multi-scale information at any given layer of a Transformer while also being efficient. It relies on the backbone to do the heavy-lifting, by using a minimal computation overhead within grafted branches, in contrast to having completely-separate branches as in CrossViT [4]. In our evaluations, we observe that GrafT delivers gains to both hybrid (CNN + ViT) Transformers (MobViT [26], MobViTv2 [27], MViTv2 [20]) and pure Transformers (DeiT [34], Swin [25], CSWin [8]). In particular, GrafT helps light-weight hybrid models become high-performing general-purpose networks with a minimal increase in parameters and FLOPs. For the light-weight network MobViT-XXS, GrafT increases the top-1 accuracy by **+1.9**% in classification as shown in Table 2 and improves the mAP by **+0.7**% in object detection as shown in Table 3. In addition, GrafT can be smoothly integrated with various self-attention methods as it worked well with regular multi-head self-attention (MHSA) in DeiT and MViTv2, shifted-window self-attention in Swin, cross-shaped window self-attention in CSWin, inter-patch self-attention in MobViT, and separable self-attention in MobViTv2. We summarize the performance of GrafT by grouping models into homogeneous (ViT [9]) and pyramid (Swin-style [25] and MobileNet-style [16]) architectures. On ImageNet-1K [7], GrafT improves the top-1 accuracy by **+3.9**% for DeiT-T as shown in Table 1, by **+1.9**% for MobViT-XXS by **+1.4**% for Swin-T, and by **+0.5**% for CSWin-T as shown in Table 2. We also observe significant gains in performance when grafted models are used as feature backbones for object detection and segmentation tasks. We believe multi-scale high-level semantics from GrafT help models to identify objects in various sizes. On COCO 2017 [23], GrafT provides **+1.6** mAP for MobViT-XS, **+1.1** AP^b and **+0.8** AP^m for Swin-T. On ADE20K [39] semantic segmentation, GrafT

provides **+1.0** mIOU^{ss}, **+1.3** mIOU^{ms} with Swin-T+GrafT. Figure 1-(b) shows the performance-complexity trade-off when integrating GrafT in the backbone on ImageNet-1K.

2. Grafting Vision Transformers

Our goal is to provide multi-scale global information to the backbone Transformer from the bottom layer so that high-level semantics from GrafT can help the Transformer to construct more efficient features. Since Graft is modular, it can be applied to various Transformer architectures. We select backbones incorporating various architectural characteristics to show that GrafT is a general-purpose module. The backbones cover hybrid and pure Transformer, homogeneous (ViT [9]) and pyramid (Swin-style [25], MobileNet-style [16]) structures, and different types of self-attention methods.

2.1. Overall Architecture

The overview of how GrafT modules are branched-out from a backbone Transformer is illustrated in Figure 1-(a). We start with a backbone Transformer (*i.e.,* ViT or Swin) and simply attach GrafT to some vertical layers of the backbone Transformer. The vertical dimension signifies the axis along which the hierarchical transitions occur within the pyramid Transformer. For an input image with size of $H \times W \times 3$, the patch tokens to the first vertical layer is $\frac{H}{4} \times \frac{W}{4} \times C$ after the patch embedding.

GrafT is a horizontal pyramid structure which consists of left-right pathway (series of GrafT downsampling), right-left pathway (series of GrafT upsampling), and bottom-up connection (multiple GrafT L-MSA) as shown in Figure 2-(b). The input feature to GrafT goes through left-right pathway (downsampling), right-left pathway (upsampling), bottom-up connection (L-MSA) and becomes a feature having strong high-level semantics at multiple scales which then gets fused into the backbone Transformer. Specifically, for the GrafT attached to the vertical layer at S^{th} stage, the input feature to the GrafT has the size of $\frac{H}{4r^{S-1}} \times \frac{W}{4r^{S-1}} \times C$ where $r = 2$ for pyramid structure and $r = 1$ for homogeneous structure. Then, we fuse the feature from GrafT to the original backbone, which will be described in the later section.

2.2. Transformer+GrafT Blocks

In this section, we describe each operation in Figure 2-(b). Let $X^{d,b}$ as an input tensor at vertical layer d and horizontal downsampling level b in GrafT with the shape: $X^{d,b} \in \mathbb{R}^{H^b \times W^b \times C}$. H^b, W^b, C is the height, width, channels of the feature map $X^{d,b}$ at a horizontal level b respectively.

Left-right pathway (downsampling): The left-right pathway creates serial feature maps at several scales with the downsampling rate r which follows the vertical pyramid downsampling rate. For example, in the first stage of Swin+GrafT, GrafT creates three downsampled feature maps $\{X^1, X^2, X^3\}$ by downsampling the input feature map X^0 from the backbone with downsampling rate 2. In left-right pathway, we use adaptive average pooling for downsampling:

$$X^{d,b+1} = A(X^{d,b}) = \rho(\text{GELU}(\text{LN}(X^{d,b}))) \quad (1)$$

A is a lightweight downsampling function which consists of LayerNorm (LN), $GELU$, and adaptive average pooling (ρ) , which is more cost-efficient than other downsampling methods such as cross attention and linear projection (see Table 6). It maps the input $X^{d,b}$ to downsampled feature $X^{d,b+1}$: $\mathbb{R}^{H^b \times W^b} \mapsto \mathbb{R}^{H^{b+1} \times W^{b+1}}$ where $H^{b+1} < H^b, W^{b+1} < W^b$. X^{b+1} is the input at the horizontal downsampling level $b+1$. Downsampling happens sequentially over horizontal layers to progressively abstract the fine information in coarse features.

Right-left pathway (upsampling): The right-left pathway hallucinate serial higher resolution feature maps by upsampling low resolution feature maps. These upsampled feature maps are enhanced by feature maps from bottom-up connection that retain spatially more accurate activations and lower-level semantics. This enhancement process is similar to FPN [22]. In right-left pathway, GrafT upsampling uses learnable W-Bilinear (window-base bilinear) interpolation which is more cost-efficient than other upsampling methods as shown in Table 6. Learnable W-Bilinear interpolation solves the aliasing problem by embedding anti-aliasing weights, the sigmoid of positional embeddings, in the feature maps. Anti-aliasing weights learns perturbations for each grid in the feature map that can prevent aliasing effect. W-Bilinear interpolation is adopted to address the semantics discontinuity between local regions. Finally, our upsampling is defined as:

$$\begin{aligned} \bar{Z}^{d,b+1} &= \Phi(\tilde{Z}^{d,b+1}) \\ &= \Phi(E_{aa} \odot \alpha(Z^{d,b+1})) \\ &= T(Z^{d,b+1}) \end{aligned} \quad (2)$$

$$\bar{Z}^{d,b+1}_{u_m,v_n} = \Phi(\tilde{Z}^{d,b+1}_{i_m,j_n}) \quad (3)$$

T is a lightweight upsampling function mapping the representation back to the spatial resolution in the previous horizontal level. This function is designed to upsample output features from L-MSA. First, given the tensor $Z^{d,b+1}$, the output from L-MSA, channel mixing (α) is applied to align channels before interpolation. α consists of LayerNorm (LN), $GELU$, and a linear layer. Next, anti-aliasing embeddings (E_{aa}) are multiplied to resolve the

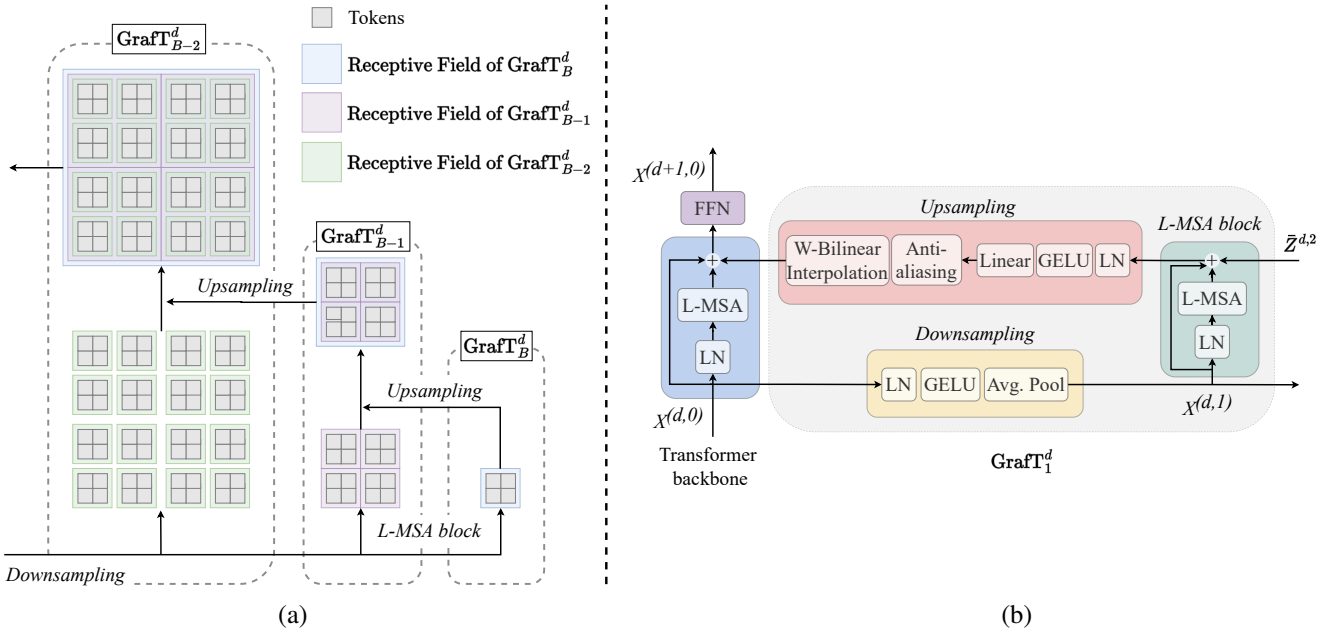

Figure 2. An overview view of GrafT. **(a):** Here, we show the receptive field of attention mechanisms in each scale and how it changes by merging with information from the corresponding lower-resolution scale (i.e., from branch-to-the-right). Multiple scale are created with downsampling, processed with local attention (in *L-MSA*)— which becomes global attention in coarsest scale (GrafT^d_B), and merged back with upsampling. When a lower-resolution representation is upsampled and merged, the effective receptive field increases, essentially giving access to efficiently-extracted global (or larger-local) information. **(b):** We present all the components and grafting/ merging points in GrafT. We graft prior to self-attention block in the backbone, and merge prior to FFN so that we can reuse the heavy computations. Downsampling (*left-right pathway*) uses light-weight average pooling to create a lower-resolution features, whereas upsampling (*right-left pathway*) uses learnable window-based bi-linear interpolation (*W-Bilinear*) to upscale. The processing unit within a GrafT module is a *L-MSA* block, which performs local-attention. When merging features to higher-resolution, we use element-wise addition.

aliasing problem. The proposed anti-aliasing embeddings are the output of sigmoid function on position embeddings $Z^{j+1}_{pos} \in R^{H^{j+1} \times W^{j+1} \times C}$. It learns to provide perturbations in the spatial dimension that prevents interpolation from suffering the aliasing problem. It is a simpler and lighter method compared to 3x3 convolutions [22]. Lastly, W-Bilinear interpolation (Φ) maps $\tilde{Z}^{d,b+1} \in \mathbb{R}^{H^{b+1} \times W^{b+1}}$ to $\bar{Z}^{d,b+1} \in \mathbb{R}^{H^b \times W^b}$. It can be described as mapping the low resolution feature in each (m, n)th window $\bar{Z}^{d,b+1}_{i_m,j_n}$ into high resolution feature in (m, n)th window $\bar{Z}^{d,b+1}_{u_m,v_n}$. (i_m, j_n) is a spatial position (i, j) in (m, n)th window where $i \in I$, $j \in J$, $m \in M$, $n \in N$. (u_m, v_n) is a spatial position (u, v) in (m, n)th window where $u \in r_h I, v \in r_w J$. r_h, r_w are the height and width ratio between feature resolutions in two consecutive horizontal levels. Upsampling happens sequentially over horizontal layers to progressively upscale spatial resolution of coarse features.

L-MSA in GrafT: The local self-attention method from the backbone Transformer limits the self-attention in a local region, which causes the discrepancy of semantics at the boundary of local regions. Therefore, bilinear interpolation is applied in each local region multiple times instead of the entire feature map at once. The bottom-up connection merges lower-level feature maps enhanced by L-MSA and higher-level feature maps upsampled by W-Bilinear interpolation by element-wise addition. The merging process iterates until it generates the high-level feature map that has the same spatial size as the input feature map X^0 from the backbone. In the example above, the coarsest feature map X^3 goes through L-MSA and becomes $\bar{Z}^3$, a feature map having the highest-level semantics. $\bar{Z}^2$ is generated by merging upsampled $\bar{Z}^3$ with output feature from L-MSA which applies local self-attention on X^2. This process iterates until the finest feature map $\bar{Z}^0$ which has the same spatial size of the X^0 is produced. $\bar{Z}^0$ is then fused into the output feature map from L-MSA in the backbone Transformer and proceed to FFN block.

$$Z^{d,b+1} = X^{d,b+1} + [\text{L-MSA}(\text{LN}(X^{d,b+1})) + \bar{Z}^{d,b+2}] \tag{4}$$

It is a simple and light block that fuses multi-scale features. It uses a standard window-based MSA from Swin [25] to encode fine features $X^{d,b+1}$. It uses simple element-wise addition to fuse this fine feature and coarse feature $\bar{Z}^{d,b+2} \in \mathbb{R}^{H^{b+1} \times W^{b+1}}$ coming from one deeper

horizontal level. Lastly, Skip connection is added to produce output $Z^{d,b+1} \in \mathbb{R}^{H^{b+1} \times W^{b+1}}$

Bottom-up connection with multiple high-level semantics: In GrafT, L-MSA uses local self-attention to enhance downsampled feature maps but it limits receptive field within each local region. It is important to exchange information among local regions so that feature can embed not only local structure but also global structure. In Figure 2-(a), the features at the bottom are the output of GrafT L-MSA where receptive field is limited to each local region. The feature map at $b+2$ level $\tilde{Z}^{b+2}$ has a receptive field, a blue color line, that covers the entire feature map. The feature map at $b+1$ level, $\tilde{Z}^{b+1}$, originally suffers the information discrepancy between local regions due to local receptive field. When $\tilde{Z}^{b+2}$ is merged into $\tilde{Z}^{b+1}$, $\tilde{Z}^{b+1}$ inherits a receptive field from $\tilde{Z}^{b+2}$ and this newly added global receptive field provides higher-level semantics that can understand the relation between local regions. $\tilde{Z}^{b}$, the feature map at b level, also originally suffers the information discrepancy between local regions. When $\tilde{Z}^{b+1}$ is merged into $\tilde{Z}^{b}$, $\tilde{Z}^{b}$ inherits receptive fields from both $\tilde{Z}^{b+1}$ $\tilde{Z}^{b+2}$ and this newly added receptive fields provides multi-scale high-level semantics that can understand the relation between local regions in multiple aspects. Multi-scale receptive fields which are progressively generated from multi-scale features resolve the drawback of local receptive field formed by L-MSA in the backbone Transformer. The L-MSAin the original backbone Transformer is defined as:

$$Y^{d,0} = X^{d,0} + [\text{L-MSA}(X^{d,0}) + \bar{Z}^{d,1}] \tag{5}$$

L-MSA is the local multi-self attention that the Transformer in the main branch is using to encode fine feature $X^{d,0} \in \mathbb{R}^{H^0 \times W^0 \times C}$. The encoded fine feature is element-wise added with coarse feature $\bar{Z}^{d,1}$ from the GrafT. Lastly, skip connection is added to produce output $Y^{d,0} \in \mathbb{R}^{H^0 \times W^0 \times C}$. It is interesting that a simple element-wise addition successfully fuse the fine feature in the main branch and the coarse feature from GrafT. Thanks to the power of GrafT upsampling method for the robust fusion of the multi-scale features encoded by various MSA. Then, we use the same FFN in the backbone transfomers:

$$X^{d+1,0} = Y^{d,0} + \text{MLP}(\text{LN}((Y^{d,0}))) \tag{6}$$

It is a standard Transformer block performing channel mixing via LayerNorm (LN) and MLP on the output of L-MSA block and adds the skip connection to generate output $X^{d+1,0}$ which is the input to the next vertical layer.

Computation complexity: Average pooling and bilinear interpolation runs in $\Theta(HW)$ and L-MSA in GrafT follows the complexity of L-MSA in the backbone Transformer.

Table 1. Performance of GrafT with homogeneous architectures on ImageNet-1K [7]. DeiT-T+GrafT outperforms DeiT-T [34] by +3.9% and even surpasses PVT-T [35] (a pyramid structure).

Model	Structure	Params ↓ (M)	FLOPs ↓ (G)	Acc. ↑ (%)
DeiT-T [34]	Homoge.	5.7	1.3	72.2
DeiT-T+GrafT	Homoge.	7.9	1.2	**(+3.9) 76.1**
CrossViT-9 [4]	Homoge.	8.6	1.8	73.9
PVT-T [35]	Pyramid	13.2	1.9	75.1

Since L-MSA is more complex than $\Theta(HW)$, the complexity of Transformer+GrafT is equal to the complexity of the pure backbone Transformer. For example,

$$\Omega(\text{Swin}) = \Omega(\text{Swin+GrafT}) = 12HWC^2 + 2M^2HWC \tag{7}$$

where H,W is the width and height of feature map and M is the size of window.

3. Experiments

3.1. ImageNet-1K Classification

We integrate GrafT in various models to show that it is generally applicable. We observe consistent gains by attaching GrafT on models that are hybrid or pure Transformers, have homogeneous or pyramid structures, and exploit various self-attention methods. We train our models on the ImageNet-1K with the standard settings in [8, 8, 9, 25–27]. Additional details are incorporated in the appendix.

Homogeneous or pyramid structures: Table 1 shows the models that has homogeneous structures (i.e., with a constant spatial resolution, or #tokens) . DeiT-T+GrafT achieves +3.9% boost over DeiT-T. Even though DeiT underperforms CrossViT (-1.7%), DeiT-T+GrafT outperforms it by +2.2% as it receives high-level semantics from GrafT.

When implementing DeiT-T+GrafT, we replace global attention in the backbone with window local attention without shifting to adhere to our proposed design (refer Figure 2-(b)). We confirm that this slightly-modified DeiT (prior to applying GrafT) underperforms the original DeiT by 2%. Thus, overall, GrafT delivers +5.9% performance gain (from 70.2% to 76.1%).

Table 2 contains models that adopt a pyramid structure. GrafT consistently boosts performance in MobViT(+1.9% in -XXS, +1.4% in -XS, +0.8% in -S), MobViTv2(+1.4% in -v0.5, +1.0% in -v1.0), Swin(+1.4% in -T), CSWin(+0.6% in -XT*, +0.5% in -T), and MViTv2-T(+0.7%) with minimal increase in parameters and FLOPs and the ∼25% decrease in throughput under the conditions of a batch size of 128. Models with SOTA accuracies at each scale are highlighted in bold. These results show that GrafT generalizes well.

Table 2. Performance of GrafT with pyramid architectures on ImageNet-1K [7]. Graft shows consistent gains across various architectures (Pure/Hybrid), model sizes and attention mechanisms.

Model	Type	Params ↓ (M)	FLOPs ↓ (G)	Acc. ↑ (%)	Thrput (FPS)
MobViT-XXS [26]	Hybrid	1.27	0.42	69.0	3771
MobViT-XXS+GrafT	Hybrid	1.43	0.44	(+1.9) 70.9	2646
MobViTv2-0.5 [27]	Hybrid	1.37	0.48	70.2	3996
MobViTv2-0.5+GrafT	Hybrid	1.53	0.49	**(+1.4) 71.6**	2791
MobileFormer-52 [5]	Hybrid	3.5	52M	68.7	-
MobViT-XS [26]	Hybrid	2.32	1.05	74.8	2134
MobViT-XS+GrafT	Hybrid	2.65	1.10	**(+1.4) 76.2**	1592
MobViT-S [26]	Hybrid	5.58	1.99	78.4	1691
MobViT-S+GrafT	Hybrid	6.38	2.13	**(+0.8) 79.2**	1274
MobViTv2-1.0 [27]	Hybrid	4.90	1.84	78.1	2023
MobViTv2-1.0+GrafT	Hybrid	5.52	1.90	(+1.0) 79.1	1391
ViL-Tiny-RPB [38]	Transformer	7	1.3	76.7	-
CSWin-XT* [8]	Transformer	6	1.2	77.4	2675
CSWin-XT+GrafT	Transformer	8	1.3	(+0.6) 78.0	2240
PVT-M [35]	Transformer	44	6.7	81.2	-
PoolFormer-S36 [36]	Transformer	31	5.2	81.4	-
$T2T_t$-14 [37]	Transformer	22	6.1	81.7	-
TNT-S [11]	Transformer	24	5.2	81.5	-
ViL-S-RPB [38]	Transformer	25	4.9	82.4	-
RegionViT-S [3]	Transformer	31	5.3	82.6	-
Swin-T [25]	Transformer	29	4.5	81.3	1357
Swin-T+GrafT	Transformer	34	5.1	(+1.4) 82.7	969
CSWin-T [8]	Transformer	23	4.3	82.7	1168
CSWin-T+GrafT	Transformer	29	4.7	**(+0.5) 83.2**	905
MViTv2-T [20]	Hybrid	24	4.7	82.3	695
MViTv2-T+GrafT	Hybrid	28	5.6	(+0.7) 83.0	488

Hybrid or pure Transformer: One research direction follows pure Transformers, whereas another follows hybrid models (i.e., CNNs+Transformers). GrafT performs well in both pure Transformers (eg: +1.4% in Swin-T+GrafT) and hybrid models. In particular, its efficiency is heightened when applied in powerful, light-weight hybrid models (eg: +1.9% MobViT-XXS+GrafT).

Various self-attention methods: As shown in Figure 2-(b), GrafT inherits the self-attention operation from the backbone that it is applied in. Therefore, we explore whether GrafT works with different self-attention mechanisms. For instance, DeiT-T and MViTv2-T (w/ global MSA) gain +3.9%, +0.7% whereas Swin-T (w/ shifted-window local MSA) obtains +1.4%. CSWin-T utilizes cross-shaped MSA and gains +0.5%. In MobViT-XXS (w/ inter-path MSA) and MobViTv2-0.5 (w/ separable MSA), GrafT gives +1.9% and +1.4% boosts. We observe that GrafT provides consistent gains by inheriting the specific type of self-attention in each backbone.

3.2. Object Detection and Segmentation

We benchmark GrafT on object detection and segmentation, showing its capabilities as a general-purpose model. We consider various backbones such as pyramid structures, hybrid architectures an pure Transformers, each having different self-attention mechanisms.

Experiments on object detection and instance segmentation: We run single shot and two-stage object detection on the COCO 2017 [23] by following a standard settings in MobViT [24, 26] and Swin [25]. The detailed settings are in the appendix. In Table 3, we show the benefit of GrafT in single-shot object detection. In MobViT-XS, it gains +1.6% mAP while only increasing parameters by 12% and FLOPs by 5%. It even outperforms MobileNetv3 by 4.4% with a 60% model-size. GrafT helps the mobile-size Transformers to detect various sizes of objects by providing multi-scale features as shown in Fig 3. The results of MobViTv2 are placed in the supplementary.

Table 3. Performance of GrafT with mobile backbones on a single shot object detection task on the COCO 2017 [23]. GrafT consistently improves the detection performance of MobViT [26].

Model	Type	Params ↓ (M)	FLOPs ↓ (G)	Acc. ↑ (%)
MobViT-XXS [26]	Hybrid	1.7	0.90	19.9
MobViT-XXS+GrafT	Hybrid	1.9	0.91	**(+0.7) 20.6**
MobileNetv1 [16]	CNN	5.1	1.3	22.2
MobileNetv2 [28]	CNN	4.3	0.8	22.1
MobileNetv3 [14]	CNN	5.0	0.6	22.0
MobileViT-XS [26]	Hybrid	2.7	1.89	24.8
MobileViT-XS+GrafT	Hybrid	3.1	1.98	**(+1.6) 26.4**
MobileViT-S [26]	Hybrid	5.7	3.48	27.7
MobileViT-S+GrafT	Hybrid	6.5	3.65	**(+1.1) 28.8**

Table 4. Performance of GrafT on two-stage object detection and instance segmentation on COCO 2017 [23]. GrafT outperforms Swin [25]. Here, 1× (SS) corresponds to 12 epochs with single scale, 3× (MS) corresponds to 36 epochs with multi-scale.

Model	Params↓	FLOPs↓	1x (SS)		3x (MS)	
			AP^b↑	AP^m↑	AP^b↑	AP^m↑
PVT-S [35]	44	245	40.4	37.8	43.0	39.9
Swin-T [25]	48	264	42.2	39.1	46.0	41.6
Swin-T+GrafT	53	275	43.3 **(+1.1)**	39.9 **(+0.8)**	47.0 **(+1.0)**	42.5 **(+0.9)**
RegionViT-S [3]	50	171	42.5	39.5	46.3	42.3
VIL-S-RPB [38]	45	277	–	–	47.1	42.7
CSWin-T [8]	42	279	46.7	42.2	49.0	43.6

Table 5. Performance of GrafT on semantic segmentation on ADE20K [39]. GrafT outperforms Swin [25].

Model	Params ↓ (M)	FLOPs ↓ (G)	mIOU ↑ (SS)	mIOU ↑ (MS)
Swin-T [25]	59	945	44.5	45.8
Swin-T + Graft	66	955	(+1.0) 45.5	(+1.3) 47.1

Table 4 presents two-stage object detection and instance segmentation settings. Swin-T+GrafT shows consistent gains in both 1× and 3× schedules. Even though Swin-T underperforms RegionViT-S, our Swin-T+GrafT outperforms it, showing an overall gain of +1.1 AP^b +0.8 AP^m in 1x(SS), +1.0 AP^b, +0.9 AP^m in 3x(MS). This implies that GrafT is applicable as a general-purpose model.

Experiments on semantic segmentation: For semantic segmentation, we train Swin-T on ADE20K [39] while following a standard training procedure similar to Swin [25] We provide implementation details in the appendix. Table 5

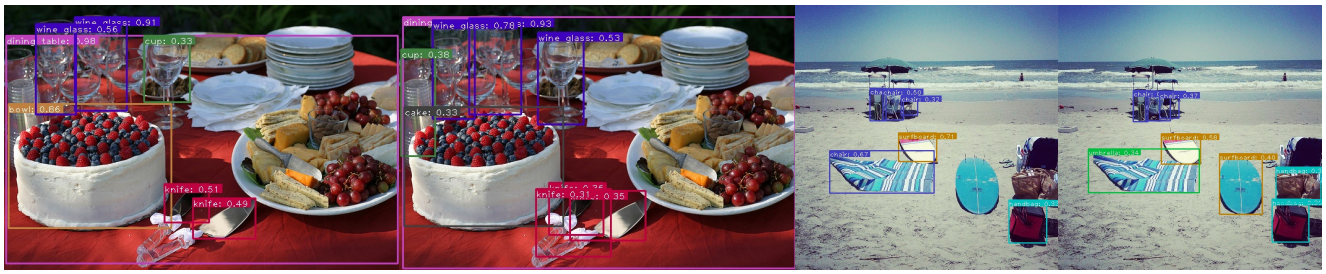

Figure 3. Object detection results of MobViT-XS [26] and MobViT-XS+GrafT on the COCO 2017 [23] validation set. In the left figure, GrafT correctly detects cake, cup, and wine glass correctly in contrast to the baseline. In the right figures, GrafT correctly detects surfboard and handbag. It shows to be capturing multi-scale information better.

Table 6. Different upsampling (US) and downsampling (DS) approaches considered in GrafT, evaluated on ImageNet-1K [7]. We fix the upsampling method when ablating downsampling methods, and vice-versa. Learn. W-Bilinear interpolation and avg. pooling are the best methods.

Fixed sampling	Varied sampling	Params ↓ (M)	FLOPs ↓ (G)	Thr. ↑ (im/s)	Acc. ↑ (%)
Linear proj.	Nearest nei. (US)	8.4	1.3	2890	74.8
	Cross att. (US)	8.7	1.3	2392	74.4
	Learn. bilin. (US)	8.5	1.3	2260	75.2
Learned bilin.	Linear proj. (DS)	8.5	1.3	2260	75.2
	Cross att. (DS)	7.9	1.2	2834	75.4
	Avg. pool. (DS)	7.9	1.2	3143	76.1

shows that Swin+GrafT outperform Swin in both single-scale (by +1.0%) and multi-scale (by +1.3%) settings. The multi-level semantics from GrafT has enabled our model to capture better pixel-level details.

3.3. Ablations on ImageNet-1K

We conduct the following experiments to better understand the design decisions of GrafT. Here, we consider Swin-T and Deit-T backbones.

Downsampling & upsampling in GrafT:. Table 6 shows the performance of DeiT-T+GrafT with different horizontal downsampling and upsampling approaches. We fix downsampling to be a linear projection when ablating upsampling approaches. Note that, in Cross attention, the finer-level features (in backbone) are used as as query while coarser-level features (in GrafT) acts as key/value. We also explore Nearest neighbor interpolation as the simplest upsampling method. Learnable Bilinear (i.e., Window-based Bilinear) interpolation uses anti-aliasing weights, and is applied in each local region separately. The results show that Learnable W-Bilinear interpolation achieves the highest accuracy with a reasonable complexity and speed.

We fix upsampling to be Learnable W-Bilinear when ablating downsampling approaches. Linear projection basically concatenates neighboring tokens and applies a linear layer (similar to Patch Merging in Swin). Cross attention first creates a coarser-level feature by average pooling a fine feature in the backbone (as query), using the backbone feature as key/value to perform cross attention. Average pooling simply creates a coarser feature by pooling. The results show that the simple average pooling achieves a better trade-off. Therefore, we adopt learnable W-Bilinear interpolation as upsampling and average pooling as downsampling in GrafT, by default.

#Scales in GrafT: In GrafT, we explore a horizontal pyramid structure where multi-scale low-resolution features are created. Here, we try to understand whether such multiple high-level semantics are useful, going beyond a single-scale. In Swin, we consider a maximum of $8\times$ horizontal downsampling (in 3 scales), similar to its original vertical downsampling. Table 7a shows that higher the #scales, the better. Swin-T+GrafT with 3-scales gives +1.4% with a little overhead compared to single-scale. Here, at the first stage of Swin (where inpiut resolution is 56×56), GrafT creates additional features of (28×28), (14×14), (7×7) resolutions to deliver multi-scale global information, even at the start of the network.

#Branches in GrafT: Table 7b shows that the accuracy consistently increases with #Grafts in DeiT. Therefore, we attach GrafT to entire layers after the first layer ($11\times$). It is not attached on the first layer to encode enough representations before being used to create coarser features through GrafT.

Supplementary Ablation The examination of upsampling elements and the replacement of GrafT with existing convolution modules have been incorporated in the appendix.

4. Related Work

Vision Transformers: Convolution neural networks (CNNs) have been widely adopted as it have shown promising performance [6, 12, 16–19, 28, 30, 32, 33] on small-scale dataset such as ImageNet-1K [7]. Inductive biases such as translation invariance and locality from CNNs are the key reasons to be trained well from scratch in small-scale

Table 7. Considering (a) the different number of scales within a GrafT at each layer, and (b) the different number of GrafTs along vertical layers, evaluated on ImageNet-1K [7].

(a)

Model	#Scales	Params ↓ (M)	FLOPs ↓ (G)	Acc. ↑ (%)
Swin-T [25]	0	29	4.5	81.3
Swin-T + GrafT	1	33.5	5.0	(+1.0) 82.3
	2	33.9	5.1	(+1.2) 82.5
	3	34.0	5.1	**(+1.4) 82.7**

(b)

Model	#Grafts	Params ↓ (M)	FLOPs ↓ (G)	Acc. ↑ (%)
DeiT-T [34]	0	5.7	1.3	72.2
DeiT-T + GrafT	4	6.5	1.2	(+2.2) 74.4
	8	7.3	1.2	(+3.4) 75.6
	11	7.9	1.2	**(+3.9) 76.1**

dataset. Recently, Transformers (*e.g.,* ViT [9] or DeiT [34]) achieved comparable results to CNNs. The first type is a pure Transformer with a homogeneous structure like ViT where the number of tokens and channels do not change over the vertical layers. T2T [37] proposes a progressive tokenization method where spatial structures are preserved. CrossViT [4] creates two branches to formulate both local and global information and exchange information. The second type is a pure Transformer with a pyramid structure such as PiT [13] and PVT [35] where vertical layers are divided into multiple stages, and the number of tokens is progressively decreased while the channel size increases over stages. Swin [25] introduces shifted-window self-attention where self-attention is performed in each window and shifting window mechanism exchanges information among windows in a pyramid structure. RegionViT [3] creates two branches to formulate local tokens and global tokens like CrossViT and assign each global token to local tokens in the same region to exchange information. CSWin [8] introduces cross-shaped window self-attention where half of the channels are used to create vertical stripes as local regions, and the other half is used to create horizontal stripes as local regions. The third type is a light-weight hybrid Transformer with a pyramid structure. Researchers have designed a hybrid Transformer where CNNs are combined with a Transformer [5, 21, 26, 27] to compete with the well-studied light-weight CNNs [12, 14, 16, 28]. For example, MobViT [26] places light-weight MobileNet blocks in the early stages to capture local features and exploits Transformer blocks in the late stages to capture global features. This process successfully incorporates spatial inductive biases through CNNs in Transformers. As a result, the model size is substantially reduced while stability and performance are improved.

In this paper, we propose GrafT, a simple and cost-efficient add-on that provides rich global information to backbone Transformers where there is a lack of communication between local regions because of local self-attention.

Exploiting multi-scale global tokens: In pyramid structure Transformers (e.g., Swin [25], CSWin [8] MViTv2 [20], iFormer [29], and CMT [10]), the scale of features varies across different stages, and high-level semantics are introduced at the final layers. In contrast, GrafT enables the generation of multi-scale features at each layer, capturing objects of different sizes/scales even at the initial layer, thereby enhancing the efficiency of representations.

Some Transformers such as CrossViT [4], TNT [11], RegionViT [3] keep two branches to encode low-level semantics and high-level semantics from the early stage. However, having two separate branches is detrimental to throughput and requires careful design of choosing which layers to exchange local and global information and the right size ratio of low-resolution and high-resolution features as mentioned in CrossViT [4]. ViL [38] creates global tokens through random initialization but they do not contain good inductive bias of multi-scale local tokens. GrafT is unique in the sense that it delivers multiple high-level semantics by exploiting horizontal pyramid structure and uses simple element-wise addition to fuse global information to the backbone Transformer. It is applicable to Transformers with both homogeneous structure and pyramid structure and improves the performance of Transformers without increasing the computation complexity due to the light-weight components as described in 2.2. We provide additional related work and the differences from the prior work in the appendix.

5. Conclusion

In this paper, we introduced GrafT: an add-on component that can easily be adopted in hybrid and pure Transformers, homogeneous and pyramid structures, and various self-attention methods, enabling multi-scale feature fusion in arbitrary depths of a model. The proposed GrafT branches are designed to be efficient, relying on the backbone to perform heavy computations. In fact, it gives consistent gains at a minimal computation burden. We also observe its effectiveness across multiple backbones and various benchmarks, including classification, detection, and segmentation. In the current work, GrafT is applied to three well-known Transformers: DeiT, Swin, and CSWin and three well-known hybrid Transformers: MobViT, MobViTv2, and MViTv2. In particular, GrafT largely benefits mobile-size models because global information is crucial to understand the scenes at that accuracy level. Going forward, we hope that GrafT becomes a generally used component for introducing multi-scale features in Transformers.

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3286–3295, 2019. 1

[2] Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-Performance Large-Scale Image Recognition Without Normalization. *arXiv*, 2021. 1

[3] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022. 1, 6, 8

[4] Chun-Fu (Richard) Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 357–366, October 2021. 1, 2, 5, 8

[5] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *arXiv preprint arXiv:2108.05895*, 2021. 6, 8

[6] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017. 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5, 6, 7, 8

[8] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12124–12134, June 2022. 2, 5, 6, 8

[9] 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. *CoRR*, abs/2010.11929, 2020. 1, 2, 3, 5, 8

[10] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 8

[11] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing XU, and Yunhe Wang. Transformer in transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15908–15919. Curran Associates, Inc., 2021. 1, 6, 8

[12] 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. 1, 7, 8

[13] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11936–11945, October 2021. 8

[14] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 6, 8

[15] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2, 3, 6, 7, 8

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 7

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 7

[20] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2, 6, 8

[21] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022. 8

[22] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3, 4

[23] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 6, 7

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 6

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021. 1, 2, 3, 4, 5, 6, 8

[26] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022. 2, 5, 6, 7, 8

[27] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 2, 5, 6, 8

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 6, 7, 8

[29] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *Advances in Neural Information Processing Systems*, 35:23495–23509, 2022. 8

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 7

[33] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1, 7

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021. 1, 2, 5, 8

[35] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 568–578, October 2021. 5, 6, 8

[36] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10819–10829, June 2022. 6

[37] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, October 2021. 6, 8

[38] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2998–3008, October 2021. 6, 8

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 6