

MaxViT: Multi-Axis Vision Transformer

Zhengzhong Tu^{1,2}, Hossein Talebi¹, Han Zhang¹, Feng Yang¹,
Peyman Milanfar¹, Alan Bovik², and Yinxiao Li¹

¹ Google Research

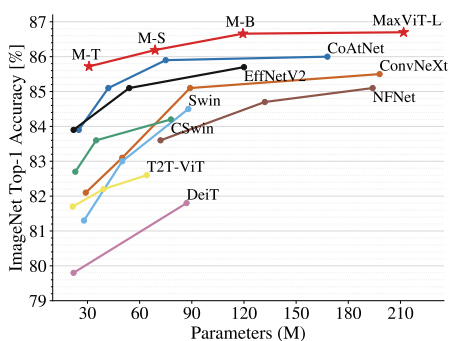
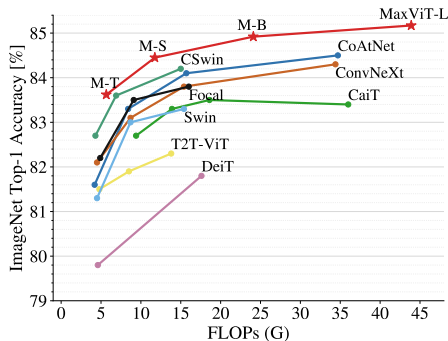
² University of Texas at Austin

Abstract. Transformers have recently gained significant attention in the computer vision community. However, the lack of scalability of self-attention mechanisms with respect to image size has limited their wide adoption in state-of-the-art vision backbones. In this paper we introduce an efficient and scalable attention model we call multi-axis attention, which consists of two aspects: blocked local and dilated global attention. These design choices allow global-local spatial interactions on arbitrary input resolutions with only linear complexity. We also present a new architectural element by effectively blending our proposed attention model with convolutions, and accordingly propose a simple hierarchical vision backbone, dubbed MaxViT, by simply repeating the basic building block over multiple stages. Notably, MaxViT is able to “see” globally throughout the entire network, even in earlier, high-resolution stages. We demonstrate the effectiveness of our model on a broad spectrum of vision tasks. On image classification, MaxViT achieves state-of-the-art performance under various settings: without extra data, MaxViT attains 86.5% ImageNet-1K top-1 accuracy; with ImageNet-21K pre-training, our model achieves 88.7% top-1 accuracy. For downstream tasks, MaxViT as a backbone delivers favorable performance on object detection as well as visual aesthetic assessment. We also show that our proposed model expresses strong generative modeling capability on ImageNet, demonstrating the superior potential of MaxViT blocks as a universal vision module. The source code and trained models will be available at <https://github.com/google-research/maxvit>. pytorch版本在timm

Keywords: Transformer, Image classification, Multi-axis attention.

1 Introduction

Convolutional Neural Networks (ConvNets) have been the dominant architectural design choice for computer vision [29, 48, 75, 76] since AlexNet [48]. ConvNets continue to excel on numerous vision problems by going deeper [75], wider [74, 76], adding dense connections [37], efficient separable convolutions [35, 70], 枯燥的 艺术的 美的 评估 convolutions [9], using encoder-decoder frameworks [67], and even introducing modern micro-design components [57]. Meanwhile, as inspired by the evolution of self-attention models like Transformers [85] in natural language processing [20, 49, 63, 100], numerous researchers have started to introduce attention



(a) Accuracy *vs.* FLOPs performance scaling curve under ImageNet-1K training setting at input resolution 224×224 .

(b) Accuracy *vs.* Parameters scaling curve under ImageNet-1K fine-tuning setting allowing for higher sizes (384/512).

Fig. 1: Performance comparison of MaxViT with state-of-the-art vision Transformers on ImageNet-1K. Our model shows superior performance in terms of both accuracy *vs.* computation and accuracy *vs.* parameters tradeoff.

mechanisms into vision [6, 88]. The Vision Transformer (ViT) [22] is perhaps the first fully Transformer-based architecture for vision, whereby image patches are simply regarded as sequences of words and a transformer encoder is applied on these visual tokens. When pre-trained on large-scale datasets [73], ViT can achieve compelling results on image recognition.

However, it has been observed that without extensive pre-training [22, 81] ViT underperforms on image recognition. This is due to the strong model capacity of Transformers, that is 充满了 with less inductive bias, which leads to overfitting. To properly regularize the model capacity and improve its scalability, numerous subsequent efforts have studied sparse Transformer models 量身定制 for vision tasks such as local attention [16, 50, 56, 99]. These methods typically reintroduce hierarchical architectures to compensate for the loss of non-locality. The Swin Transformer [56] is one such successful attempt to modify Transformers by applying self-attention on shifted non-overlapping windows. For the first time, this approach outperformed ConvNets on the ImageNet benchmark with a pure vision Transformer. Despite having more flexibility and generalizability than the full attention used in ViT, window-based attention has been observed to have limited model capacity due to the loss of non-locality, and 今后/此后 not good unfavorably on larger data regimes such as ImageNet-21K and JFT [19]. However, acquiring global interactions via full-attention at early or high-resolution stages in a hierarchical network is computationally heavy, as the attention operator requires quadratic complexity. How to efficiently incorporate global and local interactions to balance the model capacity and generalizability under a computation budget still remains challenging.

In this paper, we present a new type of Transformer module, called multi-axis self-attention (Max-SA), that capably serves as a basic architecture component

which can perform both local and global spatial interactions in a single block. Compared to full self-attention, **Max-SA enjoys greater flexibility and efficiency, i.e., naturally adaptive to different input lengths with linear complexity;** in contrast to (shifted) window/local attention, **Max-SA allows for stronger model capacity by proposing a global receptive field.** Moreover, with merely linear complexity, Max-SA can be used as a general stand-alone attention module in any layer of a network, even in earlier, high-resolution stages.

To demonstrate its effectiveness and universality, we further design a simple but effective vision backbone called **Multi-axis Vision Transformer (MaxViT)** by hierarchically stacking repeated blocks composed of Max-SA and convolutions. While our proposed model belongs to the category of hybrid vision Transformers, MaxViT distinguishes from previous approaches [19, 94] in that we strive for simplicity, by designing a basic block unifying convolution, local, and global attention, then simply repeating it. Our experiments shows that the MaxViT significantly improves upon state-of-the-art (SOTA) performance under all data regimes for a broad range of visual tasks including classification, object detection and segmentation, image aesthetics assessment, and image generation. Specifically, as Figure 1 shows, MaxViT outperforms all recent Transformer-based models in regards to both accuracy *vs.* FLOPs and accuracy *vs.* parameter curves. Our contributions are:

- A generic strong Transformer backbone, **MaxViT**, that can capture both local and global spatial interactions throughout every stage of the network.
- A novel stand-alone multi-axis attention module composed of blocked local and dilated global attention, enjoying global perception in linear complexity.
- We demonstrate large amounts of design choices including number of layers, layouts, the use of MBConv, *etc.* with extensive ablation studies, that eventually converge towards our final modular design, the MaxViT-Block.
- Our extensive experiments show that MaxViT achieves SOTA results under various data regimes for a broad range of tasks including image classification, object detection, image aesthetic assessment, and image generation.

形象审美评估

2 Related work

Convolutional networks. Since AlexNet [48], convolutional neural networks (ConvNets) have been used ^{作为事实上的解决方案} as *de facto* solutions to almost all vision tasks [8, 13, 29, 37, 51, 78, 89, 90, 104] before the “Roaring 20s” [57]. Phenomenal architectural improvements have been made in the past decade: residual [29] and dense connections [37], fully-convolutional networks [58], encoder-decoder schemes [67], feature pyramids [52], increased depths and widths [75], spatial- and channel-wise attention models [36, 91], non-local interactions [88], to name a few. A remarkable recent work ConvNeXt [57] has re-introduced core designs of vision Transformers and shown that a ‘modernized’ pure ConvNet can achieve performance comparable to Transformers on broad vision tasks.

Transformers in vision. Transformers were originally proposed for natural language processing [85]. The debut of the Vision Transformer (ViT) [22] in 2020

showed that pure Transformer-based architectures are also effective solutions for vision problems. The elegantly novel view of ViT that treats image patches as visual words has stimulated ^{爆炸|激烈的|热情的} explosive research interest in visual Transformers. To account for locality and 2D nature of images, the Swin Transformer aggregates attention in shifted windows in a hierarchical architecture [56]. More recent works have been focused on improving model and data efficiency, including sparse attention [1, 21, 64, 86, 96, 99], improved locality [27, 101], pyramidal designs [24, 87, 97], improved training strategies [3, 81, 82, 105], *etc.* We refer readers to dedicated surveys [44, 44] of vision Transformers for a ^{综合的} comprehensive review.

Hybrid models. Pure Transformer-based vision models have been observed to generalize poorly due to relatively less inductive bias [19, 22, 81]. Vision Transformers also exhibit substandard optimizability [94]. An ^{有趣的} intriguingly simple improvement is to adopt a hybrid design of Transformer and convolution layers such as using a few convolutions to replace the coarse patchify stem [19, 94]. A broad range of works fall into this category, either explicitly hybridized [4, 19, 23, 24, 93, 94, 98] or in an implicit fashion [16, 56].

Transformer for GANs. Transformers have also proven effective in generative adversarial networks (GANs) [26]. TransGAN [40] built a pure Transformer GAN with a careful design of local attention and upsampling layers, demonstrating effectiveness on small scale datasets [18, 47]. GANformer [38] explored efficient global attention mechanisms to improve on StyleGAN [42] generator. HiT [103] presents an efficient Transformer generator based on local-global attention that can scale up to 1K high-resolution image generation.

3 Method

Inspired by the sparse approaches presented in [83, 103], we introduce a new type of attention module, dubbed blocked multi-axis self-attention (Max-SA), by decomposing the fully dense attention mechanisms into two sparse forms – window attention and grid attention – which reduces the quadratic complexity of vanilla attention to linear, without any loss of non-locality. Our sequential design offers greater simplicity and flexibility, while performing even better than previous methods – each individual module can be used either standalone or combined in any order (Tables 7-9), whereas parallel designs [83, 103] offer no such benefits. Because of the flexibility and scalability of Max-SA, we are able to build a novel vision backbone, which we call MaxViT, by simply stacking alternative layers of Max-SA with MBConv [35] in a hierarchical architecture, as shown in Figure 2. MaxViT benefits from global and local receptive fields throughout the entire network, from shallow to deep stages, demonstrating superior performance in regards to both model capacity and generalization abilities.

3.1 Attention

Self-attention allows for spatial mixing of entire spatial (or sequence) locations while also benefiting from content-dependent weights based on normalized pairwise similarity. The standard self-attention defined in [22, 85] is location-unaware,

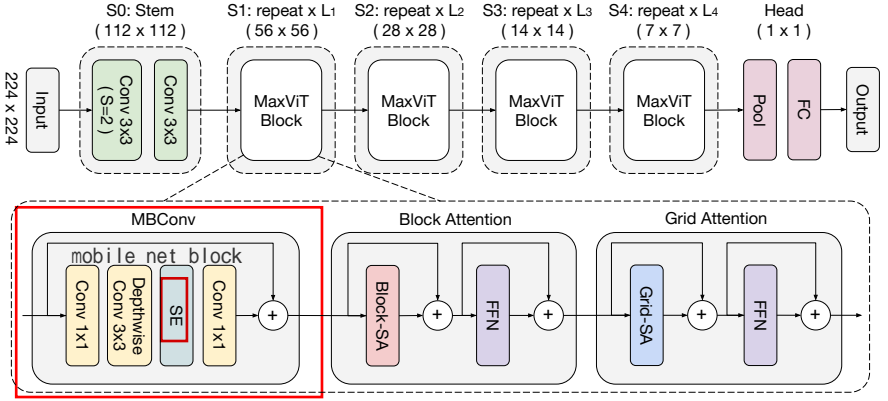


Fig. 2: **MaxViT architecture.** We follow a typical hierarchical design of ConvNet practices (e.g., ResNet) but instead build a new type of basic building block that unifies MBConv, block, and grid attention layers. Normalization and activation layers are omitted for simplicity.

i.e., non-translation equivariant, an important inductive bias imbued in ConvNets. **Relative self-attention** [19, 40, 56, 71] has been proposed to improve on vanilla attention by introducing a relative learned bias added to the attention weights, which has been shown to consistently outperform original attention on many vision tasks [19, 40, 56]. **In this work, we mainly adopt the pre-normalized relative self-attention defined in [19] as the key operator in MaxViT.**

3.2 Multi-axis Attention

Global interaction is one of the key advantages of self-attention as compared to local convolution. However, directly applying attention along the entire space is computationally infeasible as the attention operator requires quadratic complexity. To tackle this problem, we present a multi-axis approach to decompose the full-size attention into two sparse forms – local and global – by simply decomposing the spatial axes. Let $X \in \mathbb{R}^{H \times W \times C}$ be an input feature map. Instead of applying attention on the flattened spatial dimension HW , we block the feature into a tensor of shape $(\frac{H}{P} \times \frac{W}{P}, P \times P, C)$, ^{P*P个两两做} representing partitioning into non-overlapping windows, each of size $P \times P$. Applying self-attention on the local spatial dimension *i.e.*, $P \times P$, is equivalent to attending within a small window [56]. We will use this **block attention** to conduct local interactions.

Despite ^{绕过} ~~bypassing~~ the ^{众所周知的} ~~notoriously~~ heavy computation of full self-attention, local-attention models have been observed to underfit on huge-scale datasets [19, 22]. Inspired by block attention, we present a surprisingly simple but effective way to gain sparse global attention, which we call **grid attention**. Instead of partitioning feature maps using fixed window size, we grid the tensor into the shape $(G \times G, \frac{H}{G} \times \frac{W}{G}, C)$ using a fixed $G \times G$ uniform grid, resulting in windows

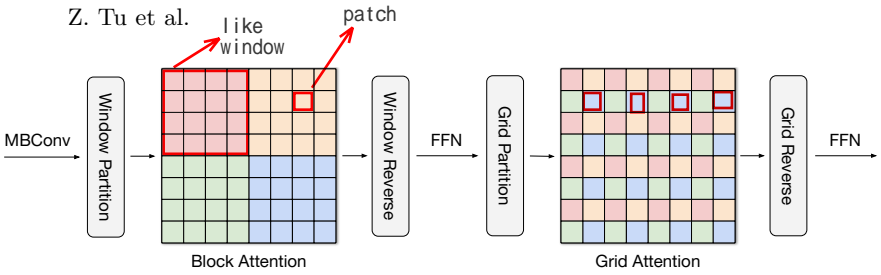


Fig. 3: **Multi-axis self-attention (Max-SA)** (best viewed in color). An illustration of the multi-axis approach for computing self-attention (window/grid size is 4×4). The block-attention module performs self-attention within windows, while the grid-attention module attends globally to pixels in a sparse, uniform grid overlaid on the entire 2D space, with both having linear complexity against input size, as we use fixed attention footage. The same colors are spatially mixed by the self-attention operation.

having adaptive size $\frac{H}{G} \times \frac{W}{G}$. Employing self-attention on the decomposed grid axis *i.e.*, $G \times G$, corresponds to dilated, global spatial mixing of tokens. By using the same *fixed* window and grid sizes (we use $P = G = 7$ following Swin [56]), we can fully balance the computation between local and global operations, both having only linear complexity with respect to spatial size or sequence length. Note that our proposed Max-SA module can be a drop-in replacement of the Swin attention module [56] with exactly the same number of parameters and FLOPs. Yet it enjoys *global interaction* capability without requiring masking, padding, or cyclic-shifting, making it more implementation friendly, preferable to the shifted window scheme [56]. For instance, the multi-axis attention can be easily implemented with einops [66] without modifying the original attention operation (see Appendix). It is worth mentioning that our proposed multi-axis attention (Max-SA) is fundamentally different from the axial-attention models [33, 86]. Please see Appendix for a detailed comparison.

MaxViT block. We sequentially stack the two types of attentions to gain both local and global interactions in a single block, as shown in Figure 3. Note that we also adopt typical designs in Transformers [22, 56], including LayerNorm [2], Feedforward networks (FFNs) [22, 56], and skip-connections. We also add a MB-Conv block [35] with squeeze-and-excitation (SE) module [36] prior to the multi-axis attention, as we have observed that using MBConv together with attention further increases the generalization as well as the trainability of the network [94]. Using MBConv layers prior to attention offers another advantage, in that depth-wise convolutions can be regarded as conditional position encoding (CPE) [17], making our model free of explicit positional encoding layers. Note that our proposed stand-alone multi-axis attention may be used together or in isolation for different purposes – block attention for local interaction, and grid attention for global mixing. These elements can be easily plugged into many vision architectures, especially on high-resolution tasks that can benefit by global interactions with affordable computation.

Table 1: **MaxViT architecture variants.** B and C denotes number of blocks and number of channels for each stage. We set each attention head to 32 for all attention layers. For MBConv, we always use expansion rate 4 and shrinkage rate 0.25 in SE [36], following [19, 79, 80]. We use two Conv layers in the stem.

Stage	Size	MaxViT-T	MaxViT-S	MaxViT-B	MaxViT-L	MaxViT-XL
S0: Conv-stem	$1/2$	B=2 C=64	B=2 C=64	B=2 C=64	B=2 C=128	B=2 C=192
S1: MaxViT-Block	$1/4$	B=2 C=64	B=2 C=96	B=2 C=96	B=2 C=128	B=2 C=192
S2: MaxViT-Block	$1/8$	B=2 C=128	B=2 C=192	B=6 C=192	B=6 C=256	B=6 C=384
S3: MaxViT-Block	$1/16$	B=5 C=256	B=5 C=384	B=14 C=384	B=14 C=512	B=14 C=768
S4: MaxViT-Block	$1/32$	B=2 C=512	B=2 C=768	B=2 C=768	B=2 C=1024	B=2 C=1536

3.3 Architecture Variants

We designed a series of extremely simple architectural variants to explore the effectiveness of our proposed MaxViT block, as shown in Figure 2. We use a hierarchical backbone similar to common ConvNet practices [19, 29, 57, 80] where the input is first downsampled using Conv3x3 layers in stem stage (S0). The body of the network contains four stages (S1-S4), with each stage having half the resolution of the previous one with a doubled number of channels (hidden dimension). In our network, we employ *identical* MaxViT blocks throughout the entire backbone. We apply downsampling in the Depthwise Conv3x3 layer of the first MBConv block in each stage. The expansion and shrink rates for inverted bottleneck [35] and squeeze-excitation (SE) [36] are 4 and 0.25 by default. We set the attention head size to be 32 for all attention blocks. We scale up the model by increasing block numbers per stage B and the channel dimension C . We summarize the architectural configurations of the MaxViT variants in Table 1.

4 Experiments

We validated the efficacy of our proposed model on various vision tasks: ImageNet classification [48], image object detection and instance segmentation [53], image aesthetics/quality assessment [61], and unconditional image generation [26]. More experimental details can be found in the Appendix.

4.1 Image Classification on ImageNet-1K

ImageNet-1K. We show in Table 2 the performance comparisons on ImageNet-1K classification. Under the basic 224×224 setting, MaxViT outperformed the most recent strong hybrid model CoAtNet by a large margin across the entire FLOPs spectrum, as shown in Figure 1a. The MaxViT-L model sets a new performance record of 85.17% at 224×224 training without extra training strategies, outperforming CoAtNet-3 by 0.67%. In regards to throughput-accuracy trade-offs at 224^2 , MaxViT-S obtains 84.45% top-1 accuracy, 0.25% higher than CSWin-B and 0.35% higher than CoAtNet-2 with comparable throughput.

Table 2: **Performance comparison under ImageNet-1K setting.** Throughput is measured on a single V100 GPU with batch size 16, following [56, 57, 80].

	Model	Eval size	Params	FLOPs	Throughput (image/s)	IN-1K top-1 acc.
ConvNets	•EffNet-B6 [79]	528	43M	19.0G	96.9	84.0
	•EffNet-B7 [79]	600	66M	37.0G	55.1	84.3
	•RegNetY-16 [62]	224	84M	16.0G	334.7	82.9
	•NFNet-F0 [5]	256	72M	12.4G	533.3	83.6
	•NFNet-F1 [5]	320	132M	35.5G	228.5	84.7
	•EffNetV2-S [80]	384	24M	8.8G	666.6	83.9
	•EffNetV2-M [80]	480	55M	24.0G	280.7	85.1
	•ConvNeXt-S [57]	224	50M	8.7G	447.1	83.1
	•ConvNeXt-B [57]	224	89M	15.4G	292.1	83.8
	•ConvNeXt-L [57]	224	198M	34.4G	146.8	84.3
ViTs	◦ViT-B/32 [22]	384	86M	55.4G	85.9	77.9
	◦ViT-B/16 [22]	384	307M	190.7G	27.3	76.5
	◦DeiT-B [81]	384	86M	55.4G	85.9	83.1
	◦CaiT-M24 [82]	224	186M	36.0G	-	83.4
	◦CaiT-M24 [82]	384	186M	116.1G	-	84.5
	◦DeepViT-L [105]	224	55M	12.5G	-	83.1
	◦T2T-ViT-24 [101]	224	64M	15.0G	-	82.6
	◦Swin-S [56]	224	50M	8.7G	436.9	83.0
	◦Swin-B [56]	384	88M	47.0G	84.7	84.5
	◦CSwin-B [21]	224	78M	15.0G	250	84.2
	◦CSwin-B [21]	384	78M	47.0G	-	85.4
	◦Focal-S [99]	224	51M	9.1G	-	83.5
	◦Focal-B [99]	224	90M	16.0G	-	83.8
Hybrid	◊CvT-21 [93]	384	32M	24.9G	-	83.3
	◊CoAtNet-2 [19]	224	75M	15.7G	247.7	84.1
	◊CoAtNet-3 [19]	224	168M	34.7G	163.3	84.5
	◊CoAtNet-3 [19]	384	168M	107.4G	48.5	85.8
	◊CoAtNet-3 [19]	512	168M	203.1G	22.4	86.0
	◊MaxViT-T	224	31M	5.6G	349.6	83.62
	◊MaxViT-S	224	69M	11.7G	242.5	84.45
	◊MaxViT-B	224	120M	23.4G	133.6	84.95
	◊MaxViT-L	224	212M	43.9G	99.4	85.17
	◊MaxViT-T	384	31M	17.7G	121.9	85.24
	◊MaxViT-S	384	69M	36.1G	82.7	85.74
	◊MaxViT-B	384	120M	74.2G	45.8	86.34
	◊MaxViT-L	384	212M	133.1G	34.3	86.40
	◊MaxViT-T	512	31M	33.7G	63.8	85.72
	◊MaxViT-S	512	69M	67.6G	43.3	86.19
	◊MaxViT-B	512	120M	138.5G	24.0	86.66
	◊MaxViT-L	512	212M	245.4G	17.8	86.70

When fine-tuned at higher resolutions (384/512), MaxViT continues to deliver high performance compared to strong ConvNet and Transformer competitors: (1) at 384², MaxViT-B attains 86.34% top-1 accuracy, outperforming EfficientNetV2-L by 0.64%; (2) when fine-tuned at 512², our MaxViT-L (212M)

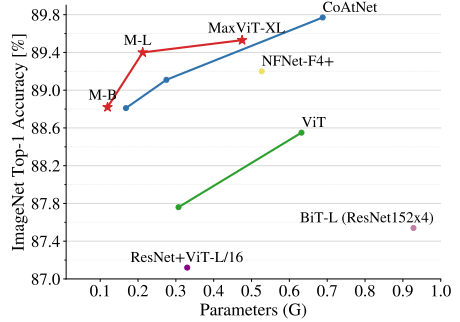
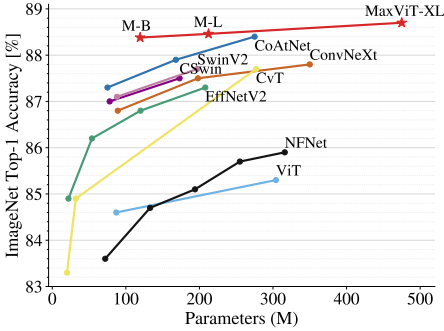
Table 3: **Performance comparison for large-scale data regimes:** ImageNet-21K and JFT pretrained models.

	Model	Eval size	Params	FLOPs	IN-1K top-1 acc.	
					21K→1K	JFT→1K
ConvNets	•BiT-R-101x3 [46]	384	388M	204.6G	84.4	-
	•BiT-R-152x4 [46]	480	937M	840.5G	85.4	-
	•EffNetV2-L [80]	480	121M	53.0G	86.8	-
	•EffNetV2-XL [80]	512	208M	94.0G	87.3	-
	•ConvNeXt-L [57]	384	198M	101.0G	87.5	-
	•ConvNeXt-XL [57]	384	350M	179.0G	87.8	-
	•NFNet-F4+ [5]	512	527M	367G	-	89.20
ViTs	◦ViT-B/16 [22]	384	87M	55.5G	84.0	-
	◦ViT-L/16 [22]	384	305M	191.1G	85.2	-
	◦ViT-L/16 [22]	512	305M	364G	-	87.76
	◦ViT-H/14 [22]	518	632M	1021G	-	88.55
	◦HaloNet-H4 [84]	512	85M	-	85.8	-
	◦SwinV2-B [56]	384	88M	-	87.1	-
	◦SwinV2-L [56]	384	197M	-	87.7	-
Hybrid	◊CvT-W24 [93]	384	277M	193.2G	87.7	-
	◊R+ViT-L/16 [22]	384	330M	-	-	87.12
	◊CoAtNet-3 [19]	384	168M	107.4G	87.6	88.52
	◊CoAtNet-3 [19]	512	168M	214G	87.9	88.81
	◊CoAtNet-4 [19]	512	275M	360.9G	88.1	89.11
	◊CoAtNet-5 [19]	512	688M	812G	-	89.77
	◊MaxViT-B	384	119M	74.2G	88.24	88.69
	◊MaxViT-L	384	212M	128.7G	88.32	89.12
	◊MaxViT-XL	384	475M	293.7G	88.51	89.36
	◊MaxViT-B	512	119M	138.3G	88.38	88.82
	◊MaxViT-L	512	212M	245.2G	88.46	89.41
	◊MaxViT-XL	512	475M	535.2G	88.70	89.53

achieves top-1 accuracy 86.7% , setting new SOTA performance on ImageNet-1K under the normal training setting. As Figure 1 shows, MaxViT scales much better than SOTA vision Transformers on the ImageNet-1K trained model scale.

ImageNet-21K. Table 3 shows the results of models pre-trained on ImageNet-21K. Remarkably, the MaxViT-B model achieves 88.38% accuracy, outperforming the previous best model CoAtNet-4 by 0.28% using only 43% of parameter count and 38% of FLOPs, demonstrating greater parameter and computing efficiency. Figure 4a visualizes the model size comparison – MaxViT scales significantly better than previous attention-based models of similar complexities, across the board. Additionally, the MaxViT-XL model achieves new SOTA performance, an accuracy of 88.70% when fine-tuned at resolution 512×512 .

JFT-300M. We also trained our model on a larger-scale proprietary dataset JFT-300M which contains ~ 300 million weakly labeled images. As shown in Table 3 and Figure 4b, our model is also scalable to massive scale training data – MaxViT-XL achieves a high accuracy of 89.53% with 475 million parameters, outperforming previous models under comparable model sizes. Due to resource



(a) Accuracy *vs.* Params performances for ImageNet-21K pre-trained models.

(b) Accuracy *vs.* Params scaling curve for JFT-300M pre-trained models.

Fig. 4: **Performance comparison on large-scale pre-trained models.** MaxViT shows superior scaling performance under both ImageNet-21K and JFT-300M pre-trained settings.

limitations, we leave experiments on billion-parameter-scale models on planet-scale datasets (*e.g.*, JFT-3B [102]) as future work.

4.2 Object Detection and Instance Segmentation

Setting. We evaluated the MaxViT architectures on the COCO2017 [53] object bounding box detection and instance segmentation tasks with a two-stage framework [65]. On the object detection task, a feature-pyramid architecture [52] was employed to boost different levels of objectiveness. In the instance segmentation task, a well-known Cascade Mask-RCNN framework [28] was employed. The dataset contains 118K training and 5K validation samples. For all the compared models, the backbones are first pretrained using ImageNet-1K. The pretrained models are then used to finetune on the detection and segmentation tasks.

Results on COCO. As shown in Table 4, AP , AP_{50} , and AP_{75} are reported for comparison. The parameters and FLOPs are also reported as a reference for model complexity. The MaxViT backbone models, used in object detection and segmentation tasks, outperform all other backbones by large margins, including Swin, ConvNeXt, and UViT at various model sizes with respect to both accuracy and efficiency. Note that MaxViT-S outperforms other base-level models (*e.g.*, Swin-B, UViT-B), with about 40% less computational cost.

4.3 Image Aesthetic Assessment.

Setting. We train and evaluate the MaxViT model on the AVA benchmark [61] which contains 255K images with aesthetics scores rated by amateur photographers. Similar to [77], we split the dataset into 80%/20% training and test sets. We followed [77] and used the normalized Earth Mover’s Distance as our

Table 4: **Comparison of two-stage object detection and instance segmentation on COCO2017.** All models are pretrained on ImageNet-1K.

Backbone	Resolution	AP	AP ₅₀	AP ₇₅	AP ^m	AP ₅₀ ^m	AP ₇₅ ^m	FLOPs	Pars.
•ResNet-50 [29]	1280×800	46.3	64.3	50.5	40.1	61.7	43.4	739G	82M
•X101-32 [95]	1280×800	48.1	66.5	52.4	41.6	63.9	45.2	819G	101M
•X101-64 [95]	1280×800	48.3	66.4	52.3	41.7	64.0	45.1	972G	140M
•ConvNeXt-T [57]	1280×800	50.4	69.1	54.8	43.7	66.5	47.3	741G	-
•ConvNeXt-S [57]	1280×800	51.9	70.8	56.5	45.0	68.4	49.1	827G	-
•ConvNeXt-B [57]	1280×800	52.7	71.3	57.2	45.6	68.9	49.5	964G	-
◦Swin-T [56]	1280×800	50.4	69.2	54.7	43.7	66.6	47.3	745G	86M
◦Swin-S [56]	1280×800	51.9	70.7	56.3	45.0	68.2	48.8	838G	107M
◦Swin-B [56]	1280×800	51.9	70.5	56.4	45.0	68.1	48.9	982G	145M
◦UViT-T [14]	896×896	51.1	70.4	56.2	43.6	67.7	47.2	613G	47M
◦UViT-S [14]	896×896	51.4	70.8	56.2	44.1	68.2	48.0	744G	54M
◦UViT-B [14]	896×896	52.5	72.0	57.6	44.3	68.7	48.3	975G	74M
◦As-ViT-L [15]	1024×1024	52.7	72.3	57.9	45.2	69.7	49.8	1094G	139M
◊MaxViT-T	896×896	52.1	71.9	56.8	44.6	69.1	48.4	475G	69M
◊MaxViT-S	896×896	53.1	72.5	58.1	45.4	69.8	49.5	595G	107M
◊MaxViT-B	896×896	53.4	72.9	58.1	45.7	70.3	50.0	856G	157M

training loss. We trained MaxViT at three different input resolutions: 224^2 , 384^2 and 512^2 , initialized with ImageNet-1K pre-trained weights.

Results on AVA. To evaluate and compare our model against existing methods, we present a summary of our results in Table 5. For similar input resolutions, the proposed MaxViT-T model outperforms existing image aesthetic assessment methods. As the input resolution increases, the performance improves, benefiting from its strong non-local capacity. Also, MaxViT shows better linear correlation compared to the SOTA method [43] which uses multi-resolution inputs.

4.4 Image Generation

Setting. We evaluate the generative ability of MaxViT blocks to generate images of 128x128 resolution on ImageNet-1K. We choose the unconditional image generation to focus on the performance of different generators in GANs. We use the Inception Score (IS) [69] and the Fréchet Inception Distance (FID) [32] as quantitative evaluation metrics. 50,000 samples were randomly generated to calculate the FID and IS scores. We compared MaxViT against HiT [103], a SOTA generative Transformer model, which uses attention at low resolutions (e.g., 32, 64), and using implicit neural functions at high resolutions (e.g., 128). By contrast, MaxViT uses the proposed MaxViT block at every resolution. Note that we use an inverse block order (**GA-BA-Conv**) as we found it to perform better (see Table 8). Since Batch Normalization [39, 103] achieves better results on image generation, we replaced all Layer Norm with Batch Norm under this setting.

Table 5: **Image aesthetic assessment results on the AVA benchmark [61]**. PLCC and SRCC represent the Pearson’s linear and Spearman’s rank correlation coefficients.

Model	Res.	Pars.	PLCC \uparrow	SRCC \uparrow
•NIMA [77]	224	56M	0.636	0.612
•EffNet-B0 [79]	224	5.3M	0.642	0.620
•AFDC [10]	224	44.5M	0.671	0.649
◦ViT-S/32 [43]	384	22M	0.665	0.656
◦ViT-B/32 [43]	384	88M	0.664	0.664
◦MUSIQ [43]	224~512	27M	0.720	0.706
◊MaxViT-T	224	31M	0.707	0.685
◊MaxViT-T	384	31M	0.736	0.699
◊MaxViT-T	512	31M	0.745	0.708

Table 6: **Comparison of image generation on ImageNet**. \ddagger used a pre-trained ImageNet classifier.

Model	FID \downarrow	IS \uparrow
•GAN [26]	54.17	14.01
•PacGAN2 [54]	57.51	13.50
•MGAN [34]	50.90	14.44
•LogoGAN [68] \ddagger	38.41	18.86
•SS-GAN [12]	43.87	-
•SC GAN [55]	40.30	15.82
•ConvNet- R_1 [103]	37.18	19.55
◦HiT [103] (32.9M)	30.83	21.64
◊MaxViT (18.6M)	30.77	22.58

Results on ImageNet-1K. The results are shown in Table 6. Our MaxViT achieved better FID and IS with significantly lower number of parameters. These results demonstrate the effectiveness of MaxViT blocks for generation tasks. More details of the generative experiment can be found in Appendix.

4.5 Ablation Studies.

In this section, we ablate important design choices in MaxViT on ImageNet-1K image classification. We use the MaxViT-T model trained for 300 epochs by default and report top-1 accuracy on ImageNet-1K. Except for the ablated design choice, we used the same training configurations, unless stated otherwise.

Global grid-attention. One of our main contributions is the grid-attention module, which allows for sparse global interactions at linear time, enabling our model to capture global information at all stages. We conducted two ablations to understand its gain: 1) completely removed global attention at each stage; 2) replaced grid attention with block attention to retain the same parameter count and FLOPs. As Table 7 shows, enabling global attention at earlier stages can further boost performance over using only local attention or convolutions.

MBConv layer. We also ablated the usage of MBConv layers in MaxViT by removing all MBConv in each stage. Note that we should also consider the reduction of parameter count and FLOPs when removing the MBConv layers. Plus, Stage 3 has 5 blocks whereas other stages have only 2. As Table 9 shows, the usage of MBConv layers in MaxViT significantly boosts performance.

Block order study. We present three different modules to build the MaxViT block – MBConv, block-, and grid-attention – which captures spatial interactions from local to global. To investigate the most effective way to combine them, we evaluated the MaxViT-T model using all 6 permutations. We always apply downsampling in the first layer, which might cause a minor model size difference. We can observe from Table 8 that placing MBConv before attention layers is

Table 7: **Effects of global grid-attention.** Ablate-S1 means we remove grid-attention in stage 1 while Replace-S1 means replacing grid-attention with block-attention.

Model	Pars.	FLOPs	Top-1 Acc.
MaxViT-T	30.9M	5.6G	83.62
Ablate-S1	30.8M	5.3G	83.36(-0.26)
Ablate-S2	30.5M	5.3G	83.38(-0.24)
Ablate-S3	26.9M	4.9G	83.00(-0.62)
Replace-S1	30.9M	5.6G	83.49(-0.13)
Replace-S2	30.9M	5.6G	83.41(-0.22)
Replace-S3	30.9M	5.6G	83.40(-0.23)

Table 9: **Ablation of MBConv.** Ablate-S1 means we delete MBConv layers in stage 1. Note that the network will also be smaller if we ablate MBConv layers in some stage.

Model	Pars.	FLOPs	Top-1 acc.
MaxViT-T	30.9M	5.6G	83.62
Ablate-S1	30.8M	5.2G	83.24(-0.38)
Ablate-S2	30.5M	5.4G	83.02(-0.60)
Ablate-S3	27.6M	5.1G	82.65(-0.97)
Ablate-S4	25.7M	5.4G	83.09(-0.53)

Table 8: **Block order study.** C, BA, GA represent MBConv, block-, and grid-attention respectively.

Model	Pars.	FLOPs	Top-1 acc.
C-BA-GA	30.9M	5.6G	83.62
C-GA-BA	30.9M	5.6G	83.54(-0.08)
BA-C-GA	31.1M	5.3G	83.07(-0.55)
BA-GA-C	31.1M	5.3G	83.02(-0.60)
GA-C-BA	31.1M	5.3G	83.08(-0.54)
GA-BA-C	31.1M	5.3G	83.03(-0.59)

GAN experiments			
Model	Pars.	FID↓	IS↑
GA-BA-C	18.6M	30.77	22.68
C-BA-GA	18.6M	31.40	21.49(-1.19)

Table 10: **Sequential vs. parallel.** We compared our model with modified parallel multi-axis scheme *Paral-**.

Model	Pars.	FLOPs	Top-1 acc.
MaxViT-T	30.9M	5.6G	83.62
<i>Paral-T</i>	34.5M	6.2G	82.64(-0.98)
MaxViT-S	68.9M	11.7G	84.45
<i>Paral-S</i>	76.9M	13.0G	83.45(-1.00)
MaxViT-B	119.4M	24.2G	84.95
<i>Paral-B</i>	133.4M	26.9G	83.70(-1.25)
MaxViT-L	211.8M	43.9G	85.17
<i>Paral-L</i>	236.6M	48.8G	83.54(-1.63)

almost always better than other combinations. The reason might be that it is more suitable to get local features/patterns in early layers, then aggregate them globally, which is aligned with existing hybrid models [19, 94], which puts Conv layers in front of attention. In generative experiments (Section 4.4), however, we found the best order to be from global to local: **GA-BA-C**. We hypothesize that it may be advantageous for generation tasks to first obtain the overall structures correct with global processing blocks (*i.e.*, grid-attention layers), then fill in finer details using local processing blocks (*i.e.*, MBConv).

Sequential vs. parallel. In our approach, we sequentially stack the multi-axis attention modules following [56, 86], while there also exist other models that adopt a parallel design [83, 103]. In this ablation, we compare our sequential Max-SA against parallel branches containing block- and grid-attention respectively. Note that we use an input projection to double the channels, then split the heads to feed the two branches in order to remain similar complexity to MaxViT, and an output projection that reduces the concatenated branches. We did rough parameter tuning and found that an initial learning rate of 10^{-3} per-

forms significantly better than 3×10^{-3} for parallel models. We use all the same parameters except the learning rate. As Table 10 shows, our sequential approach remarkably outperforms parallel counterparts with fewer parameters and computation. The reason may be that the parallel designs learn complementary cues with less interactions between them, whereas our sequential stack is able to learn more powerful fusions between local and global layers.

Vertical layout. We further examine our vertical layout design, *i.e.*, the number of blocks each stage. We compared our design against the choice of Swin/ConvNeXt [56, 57]. We change MaxViT-T and -S to blocks $B = (2, 2, 6, 2)$, and MaxViT-B, -L to have blocks $B = (2, 2, 18, 2)$ strictly following the stage ratio of Swin [56]. It may be seen from Figure 5 that our layout performed comparably to Swin for small models, but scales significantly better for larger models.

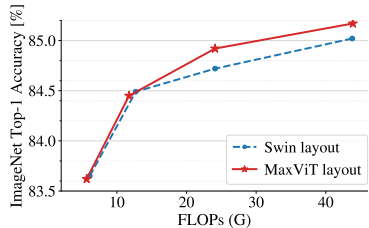


Fig. 5: Vertical layout ablation. Our model scales better than Swin layout [56].

5 Discussion and Conclusion

While recent works in the 2020s have arguably shown that ConvNets and vision Transformers can achieve similar performance on image recognition, our work presents a unified design that takes advantages of the best of both worlds – efficient convolution and sparse attention – and demonstrates that a model built on top, namely MaxViT, can achieve state-of-the-art performance on a variety of vision tasks, and more importantly, scale extremely well to massive scale data sizes. Even though we present our model in the context of vision tasks, the proposed multi-axis approach can easily extend to language modeling to capture both local and global dependencies in linear time. We also look forward to studying other forms of sparse attention in higher-dimensional or multi-modal signals such as videos, point clouds, and vision-languages.

Societal impact. Investigating the performance and scalability of large model designs would consume considerable computing resources. These efforts can contribute to increased carbon emissions, which could hence raise environmental concerns. However, the proposed model offers strong modular candidates that expand the network’s design space for future efforts on automated architectural design. If trained improperly, the proposed model may express bias and fairness issues. The proposed generative model can be abused to generate misleading media and fake news. These issues demand caution in future related research.

Acknowledgment. We thank Xianzhi Du and Wuyang Chen for extensive help on experiments. We also thank Hanxiao Liu, Zihang Dai, Anurag Arnab, Huiwen Chang, Junjie Ke, Mauricio Delbracio, Sungjoon Choi, and Irene Zhu for valuable discussions and help.

Appendix

In this Appendix we provide the following material:

- Sec. A describes the detailed architectures of MaxViT for image classification (Sec. A.1), object detection and segmentation (Sec. A.2), image aesthetics assessment (Sec. A.3), and image generation (Sec. A.4).
- Sec. B presents complete training settings and hyperparameters for image classification (Sec. B.1), object detection and segmentation (Sec. B.2), image aesthetics assessment (Sec. B.3), and image generation (Sec. B.4).
- Sec. C demonstrates comprehensive experimental results, including image classification on ImageNet-1K (Table 13), ImageNet-21K and JFT (Table 14), as well as more image generation visualizations on ImageNet-1K (Figure 8).

A Model Details

A.1 Backbone Details

MBConv MaxViT leverages the MBConv block [70, 79] as the main convolution operator. We also adopt a pre-activation structure [19, 30] to promote homogeneity between MBConv and Transformer blocks. Specifically, assume \mathbf{x} to be the input feature, the MBConv block without downsampling is formulated as:

$$\mathbf{x} \leftarrow \mathbf{x} + \text{Proj}(\text{SE}(\text{DWConv}(\text{Conv}(\text{Norm}(\mathbf{x})))), \quad (1)$$

where Norm is BatchNorm [39], Conv is the expansion Conv1x1 followed by BatchNorm and GELU [31] activation, a typical choice for Transformer-based models. DWConv is the Depthwise Conv3x3 followed by BatchNorm and GELU . SE is the Squeeze-Excitation layer [36], while Proj is the shrink Conv1x1 to down-project the number of channels. Note that for the first MBConv block in every stage, the downsampling is done by applying stride-2 Depthwise Conv3x3 while the shortcut branch should also apply pooling and channel projection:

$$\mathbf{x} \leftarrow \text{Proj}(\text{Pool2D}(\mathbf{x})) + \text{Proj}(\text{SE}(\text{DWConv}\downarrow(\text{Conv}(\text{Norm}(\mathbf{x}))))). \quad (2)$$

Relative Attention Relative attention has been explored in several previous studies for both NLP [71, 92] and vision [19, 40, 56, 84]. Here to simplify the presentation, we present our model using only a single head of the multi-head self-attention. In the actual implementation, we always use multi-head attention with the same head dimension. The relative attention can be defined as:

$$\text{RelAttention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d} + B)V, \quad (3)$$

where $Q, K, V \in \mathbb{R}^{(H \times W) \times C}$ are the query, key, and value matrices and d is the hidden dimension. The attention weights are co-decided by a learned static

location-aware matrix B and the scaled input-adaptive attention QK^T/\sqrt{d} . Considering the differences in 2D coordinates, the relative position bias B is parameterized by a matrix $\hat{B} \in \mathbb{R}^{(2H-1)(2W-1)}$. Following typical practices [19, 56], when fine-tuned at a higher resolution *e.g.*, $H' \times W'$, we use bilinear interpolation to map the relative positional bias from $\mathbb{R}^{(2H-1)(2W-1)}$ to $\mathbb{R}^{(2H'-1)(2W'-1)}$. This relative attention benefits from input-adaptivity, translation equivariance, and global interactions, which is a preferred choice over the vanilla self-attention on 2D vision tasks. In our model, all the attention operators use this relative attention defined in Eq. 3 by default.

Multi-Axis Attention We assume the relative attention operator in Eq. 3 follows the convention for 1D input sequences *i.e.*, always regards the *second last dimension* of an input $(..., L, C)$ as the *spatial axis* where L, C represent sequence length and channels. The proposed Multi-Axis Attention can be implemented without modification to the self-attention operation. To start with, we first define the $\text{Block}(\cdot)$ operator with parameter P as partitioning the input image/feature $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into non-overlapping blocks with each block having size $P \times P$. Note that after window partition, the block dimensions are gathered onto the spatial dimension (*i.e.*, -2 axis):

$$\text{Block} : (H, W, C) \rightarrow \left(\frac{H}{P} \times P, \frac{W}{P} \times P, C\right) \rightarrow \left(\frac{HW}{P^2}, P^2, C\right). \quad (4)$$

We denote the $\text{Unblock}(\cdot)$ operation as the reverse of the above block partition procedure. Similarly, we define the $\text{Grid}(\cdot)$ operation with parameter G as dividing the input feature into a uniform $G \times G$ grid, with each lattice having *adaptive size* $\frac{H}{G} \times \frac{W}{G}$. Unlike the **block** operator, we need to apply an extra **Transpose** to place the grid dimension in the assumed spatial axis (*i.e.*, -2 axis):

$$\text{Grid} : (H, W, C) \rightarrow \left(G \times \frac{H}{G}, G \times \frac{W}{G}, C\right) \rightarrow \underbrace{\left(G^2, \frac{HW}{G^2}, C\right)}_{\text{swappaxes(axis1=-2,axis2=-3)}} \rightarrow \left(\frac{HW}{G^2}, G^2, C\right) \quad (5)$$

with its inverse operation $\text{Ungrid}(\cdot)$ that reverses the gridded input back to the normal 2D feature space.

To this end, we are ready to explain the multi-axis attention module. Given an input tensor $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, the local Block Attention can be expressed as:

$$\begin{aligned} \mathbf{x} &\leftarrow \mathbf{x} + \text{Unblock}(\text{RelAttention}(\text{Block}(\text{LN}(\mathbf{x})))) \\ \mathbf{x} &\leftarrow \mathbf{x} + \text{MLP}(\text{LN}(\mathbf{x})) \end{aligned} \quad (6)$$

while the global, dilated Grid Attention module is formulated as:

$$\begin{aligned} \mathbf{x} &\leftarrow \mathbf{x} + \text{Ungrid}(\text{RelAttention}(\text{Grid}(\text{LN}(\mathbf{x})))) \\ \mathbf{x} &\leftarrow \mathbf{x} + \text{MLP}(\text{LN}(\mathbf{x})) \end{aligned} \quad (7)$$

where we omit the QKV input format in the RelAttention operation for simplicity. LN denotes the Layer Normalization [2], where MLP is a standard MLP network [22, 56] consisting of two linear layers: $\mathbf{x} \leftarrow W_2 \text{GELU}(W_1 \mathbf{x})$.

Comparison to Axial attention

It should be noted that our proposed multi-axis attention (Max-SA) module is completely different from the axial attention proposed in [33, 86]. As shown in Figure 6(a), Axial attention proposes to first apply column-wise attention then row-wise, which achieves a global receptive field with $\mathcal{O}(N\sqrt{N})$ complexity (assuming N equals to the number of pixels). On the contrary, our proposed Max-SA shown in Figure 6(b) first employs local attention, then sparse global attention, enjoying global receptive fields with only $\mathcal{O}(N)$ linear complexity. Moreover, we deem the proposed Max-SA a more natural approach for vision since the design of attended regions account for the 2D structure of images, *e.g.*, mixing tokens in a spatially-local small window.

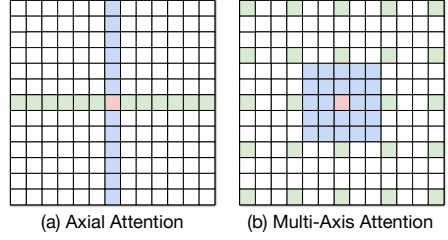


Fig. 6: Comparison of Axial attention and our proposed Multi-Axis attention.

MaxViT Block We demonstrate in Algo. 1 an `einops`-style pseudocode of the MaxViT block which contains MBConv, block attention, and grid attention.

Algo. 1 Pseudocode of MaxViT Block

```
# input: features (b, h, w, c). Assume h==w; x/output: features (b, h, w, c).
# p/g: block/grid size. Use 7 by default.

def RelSelfAttn(x): return x # A self-attn function applied on the -2 axis

# Window/grid partition function
from einops import rearrange
def block(x,p):
    return rearrange(x, "b(hy)(wx)c->b(hw)(yx)c", h=x.shape[1]//p, w=x.shape[2]//p, y=p, x=p)

def unblock(x,g,p):
    return rearrange(x, "b(hw)(yx)c->b(hy)(wx)c", h=g, w=g, y=p, x=p)

x = MBConv(input) # MBConv layer

x = block(x,p) # window partition
x = RelSelfAttn(x) # Apply window-attention
x = unblock(x,x.shape[1]//p,p) # reverse

x = block(x,x.shape[1]//g) # grid partition
x = swapaxes(x,-2,-3) # move grid-axis to -2
x = RelSelfAttn(x) # Apply grid-attention
x = swapaxes(x,-2,-3) # reverse swapaxes
output = unblock(x,g,x.shape[1]//g) # reverse
```

Classification Head Instead of using the `[cls]` token [22], we simply apply global average pooling to the output of the last stage (S4) to obtain the feature representation, followed by the final classification head.

Table 11: **Detailed architectural specifications** for MaxViT families.

	dsp. rate (out size)	MaxViT-T		MaxViT-S	
stem	2× (112 × 112)	3×3, 64, stride 2 3×3, 64, stride 1		3×3, 64, stride 2 3×3, 64, stride 1	
S1	4× (56 × 56)	MBCConv, 64, E 4, R 4 Rel-MSA, P 7×7, H 2 Rel-MSA, G 7×7, H 2	× 2	MBCConv, 96, E 4, R 4 Rel-MSA, P 7×7, H 3 Rel-MSA, G 7×7, H 3	× 2
S2	8× (28 × 28)	MBCConv, 128, E 4, R 4 Rel-MSA, P 7×7, H 4 Rel-MSA, G 7×7, H 4	× 2	MBCConv, 192, E 4, R 4 Rel-MSA, P 7×7, H 6 Rel-MSA, G 7×7, H 6	× 2
S3	16× (14 × 14)	MBCConv, 256, E 4, R 4 Rel-MSA, P 7×7, H 8 Rel-MSA, G 7×7, H 8	× 5	MBCConv, 384, E 4, R 4 Rel-MSA, P 7×7, H 12 Rel-MSA, G 7×7, H 12	× 5
S4	32× (7 × 7)	MBCConv, 512, E 4, R 4 Rel-MSA, P 7×7, H 16 Rel-MSA, G 7×7, H 16	× 2	MBCConv, 768, E 4, R 4 Rel-MSA, P 7×7, H 24 Rel-MSA, G 7×7, H 24	× 2
	dsp. rate (out size)	MaxViT-B		MaxViT-L	
stem	2× (112 × 112)	3×3, 64, stride 2 3×3, 64, stride 1		3×3, 128, stride 2 3×3, 128, stride 1	
S1	4× (56 × 56)	MBCConv, 96, E 4, R 4 Rel-MSA, P 7×7, H 3 Rel-MSA, G 7×7, H 3	× 2	MBCConv, 128, E 4, R 4 Rel-MSA, P 7×7, H 4 Rel-MSA, G 7×7, H 4	× 2
S2	8× (28 × 28)	MBCConv, 192, E 4, R 4 Rel-MSA, P 7×7, H 6 Rel-MSA, G 7×7, H 6	× 6	MBCConv, 256, E 4, R 4 Rel-MSA, P 7×7, H 8 Rel-MSA, G 7×7, H 8	× 6
S3	16× (14 × 14)	MBCConv, 384, E 4, R 4 Rel-MSA, P 7×7, H 12 Rel-MSA, G 7×7, H 12	× 14	MBCConv, 512, E 4, R 4 Rel-MSA, P 7×7, H 16 Rel-MSA, G 7×7, H 16	× 14
S4	32× (7 × 7)	MBCConv, 768, E 4, R 4 Rel-MSA, P 7×7, H 24 Rel-MSA, G 7×7, H 24	× 2	MBCConv, 1024, E 4, R 4 Rel-MSA, P 7×7, H 32 Rel-MSA, G 7×7, H 32	× 2

Architectural Specifications Finally, we present detailed architectural specifications for the MaxViT model family (T/S/B/L) in Table 11.

A.2 Detection and Segmentation Models

We follow the settings of the cascaded Faster-RCNN [65] and Mask-RCNN [28], but replace the feature extraction backbone with our MaxViT backbone. We also applied FPN [52] in the feature map generation, where the S2, S3, S4 (multi-scale features of targeted resolution 1/8, 1/16, 1/32 in MaxViT, respectively) are used. Then the generated feature maps are fed into the detection head. For fair comparison, we follow the original implementation without adopting any system-level strategies to further boost the final performance, such as the HTC framework [7], instaboost [25], *etc.* used in Swin [56]. We show the results of MaxViT-T/S/B on these two tasks to compare it against recent strong models at similar model complexity.

A.3 Image Aesthetics Model

This task requires incorporating both local and global information of an image to accurately predict human perceptual preference. To this end, the model needs to

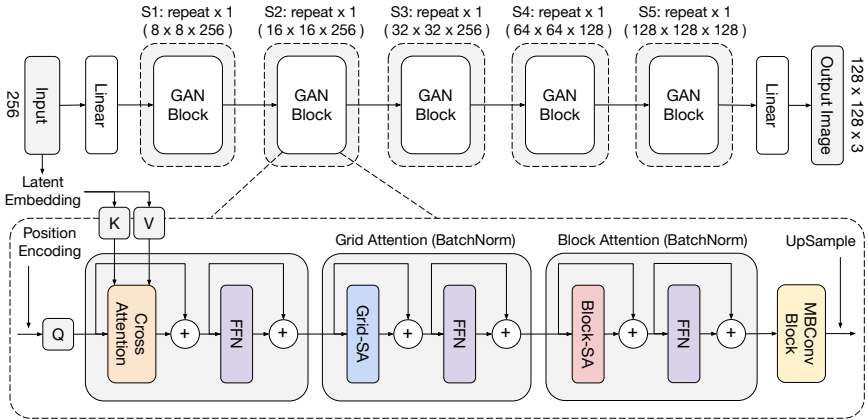


Fig. 7: **Generator architecture using the MaxViT block for the GAN experiment.** In every stage, we first use the cross-attention module to let the features attend to the latent embedding projected from the input code, which are then fed into the proposed MaxViT block consisting of grid attention, block attention, and MBConv layer. Note that unlike the main model in Sec. A.1, the order of applying the three layers are reversed: from global to local.

have the capacity to learn pixel-level quality aspects such as sharpness, noisiness and contrast as well as semantic-level aspects such as composition and depth-of-field. We follow [77] and use the normalized Earth Mover’s Distance as our training loss. Given the ground truth and predicted probability mass functions \mathbf{p} and $\hat{\mathbf{p}}$ representing the histogram of scores, the normalized Earth Mover’s Distance can be expressed as:

$$\text{EMD}(\mathbf{p}, \hat{\mathbf{p}}) = \left(\frac{1}{N} \sum_{k=1}^N |\text{CDF}_{\mathbf{p}}(k) - \text{CDF}_{\hat{\mathbf{p}}}(k)|^r \right)^{1/r} \quad (8)$$

where $\text{CDF}_{\mathbf{p}}(k)$ is the cumulative distribution function as $\sum_{i=1}^k \mathbf{p}_i$, and $N = 10$ represents the number score bins. In our experiments we set $r = 2$. We remove the classification head used in MaxViT, and instead append a fully-connected layer with 10 neurons followed by **softmax**.

A.4 GAN Model

The above image recognition tasks can validate the power of our proposed MaxViT block used in downsampling (contracting) models. For this GAN experiment, we would like to demonstrate its effectiveness in upsampling (expanding) architectures. The MaxViT-GAN model for image generation is illustrated in Figure 7. For unconditional image generation, MaxViT-GAN first takes a latent code $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as input, then progressively generates an image of target

resolution through a hierarchically upsampling structure. We start by linearly projecting the input to a feature with spatial dimension 8×8 . During the generation, the feature will go through five stages consisting of identical GAN blocks with gradually increased spatial resolution, similar to the design of our main model. Similar to [103], we apply a cross-attention layer before the MaxViT block as a memory-efficient form of self-modulation in every stage, which has been shown to stabilize GAN training and also improve mode coverage [11, 103]. We use pixel shuffle [72] for upsampling in the end of each stage.

B Experimental Settings

B.1 ImageNet Classification

We provide ImageNet-1K experimental settings of MaxViT models for both pre-training and fine-tuning in Table 12. All the MaxViT variants used similar hyperparameters except that we mainly customize the stochastic depth rate to regularize each model separately.

Table 12: **Detailed hyperparameters used in ImageNet-1K experiments.** Multiple values separated by ‘/’ are for each model size respectively.

Hyperparameter	ImageNet-1K		ImageNet-21K		JFT-300M	
	Pre-training	Fine-tuning	Pre-training	Fine-tuning	Pre-training	Fine-tuning
	(MaxViT-T/S/B/L)		(MaxViT-B/L/XL)		(MaxViT-B/L/XL)	
Stochastic depth	0.2/0.3/0.4/0.6		0.3/0.4/0.6	0.4/0.5/0.9	0.0/0.0/0.0	0.1/0.2/0.2
Center crop	True	False	True	False	True	False
RandAugment	2, 15	2, 15	2, 5	2, 15	2, 5	2, 15
Mixup alpha	0.8	0.8	None	None	None	None
Loss type	Softmax	Softmax	Sigmoid	Softmax	Sigmoid	Softmax
Label smoothing	0.1	0.1	0.0001	0.1	0	0.1
Train epochs	300	30	90	30	14	30
Train batch size	4096	512	4096	512	4096	512
Optimizer type	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Peak learning rate	3e-3	5e-5	1e-3	5e-5	1e-3	5e-5
Min learning rate	1e-5	5e-5	1e-5	5e-5	1e-5	5e-5
Warm-up	10K steps	None	5 epochs	None	20K steps	None
LR decay schedule	Cosine	None	Linear	None	Linear	None
Weight decay rate	0.05	1e-8	0.01	1e-8	0.01	1e-8
Gradient clip	1.0	1.0	1.0	1.0	1.0	1.0
EMA decay rate	None	0.9999	None	0.9999	None	0.9999

B.2 Coco Detection and Segmentation

We evaluated MaxViT on the COCO2017 [53] object bounding box detection and instance segmentation tasks. The dataset contains 118K training and 5K validation samples. All the MaxViT backbones used are pretrained on ImageNet-1k at resolution 224×224 . These pretrained checkpoints are then used as the warm-up weights for fine-tuning the detection and segmentation tasks. For both

tasks, the input images are resized to 896×896 . The training is conducted with a batch size of 256, using the AdamW [59] optimizer with learning rate of 1e-3, 3e-3, 3e-3, and stochastic depth of 0.8, 0.3, 0.3 for MaxViT-T/S/B, respectively.

B.3 Image Aesthetics Assessment

We trained and evaluated the MaxViT model on the AVA benchmark [61]. This dataset consists of 255K images rated by armature photographers through photography contests. Each image is rated by an average of 200 human raters, assigning a score from 1 to 10 to images. The higher the score, the better the visual aesthetic quality of the image. Each image in the dataset has a histogram of scores associated with it, which we use as the ground truth label. Similar to [43, 77], we split the dataset into train and test sets, such that 20% of the data is used for testing. We train MaxViT for three different input resolutions: 224×224 , 384×384 and 512×512 . We initialized the model with ImageNet-1K 224×224 pre-trained weights. The weight and bias momentums are set to 0.9, and a dropout rate of 0.75 is applied on the last layer of the baseline network. We use an initial learning rate of 1e-3, exponentially decayed with decay factor 0.9 every 10 epochs. We set the stochastic depth rate to 0.5.

B.4 Image Generation

We use a ResNet-based discriminator following [42]. To train the model, we also used the standard non-saturating logistic GAN loss with R_1 gradient penalty [60] applied to the discriminator with the gradient penalty weight set to 10. We employ the Adam [45] optimizer with a learning rate of 1e-4 for both generator and discriminator. The model is trained on TPU for one million steps with batch size 256. Notably, we do not employ extra GAN training tricks such as pixel norm, noise injection, progressive growing, *etc.* on which recent state-of-the-art models are heavily relied to attain good results [41, 42]. The overall objectives of the GAN training are defined as:

$$\mathcal{L}_G = -\mathbb{E}_{z \sim P_z} [\log(D(G(z)))], \quad (9)$$

$$\mathcal{L}_D = -\mathbb{E}_{x \sim P_x} [\log(D(x))] - \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] + \gamma \mathbb{E}_{x \sim P_x} [\|\nabla_x D(x)\|_2^2], \quad (10)$$

where γ denotes the R_1 gradient penalty weight.

C Complete Experimental Results

We provide complete experiment comparisons for ImageNet-1K, Image-21K, and JFT datasets in Table 13 and Table 14, respectively. We also provide more visual results for unconditional image generation on ImageNet-1K in Figure 8.

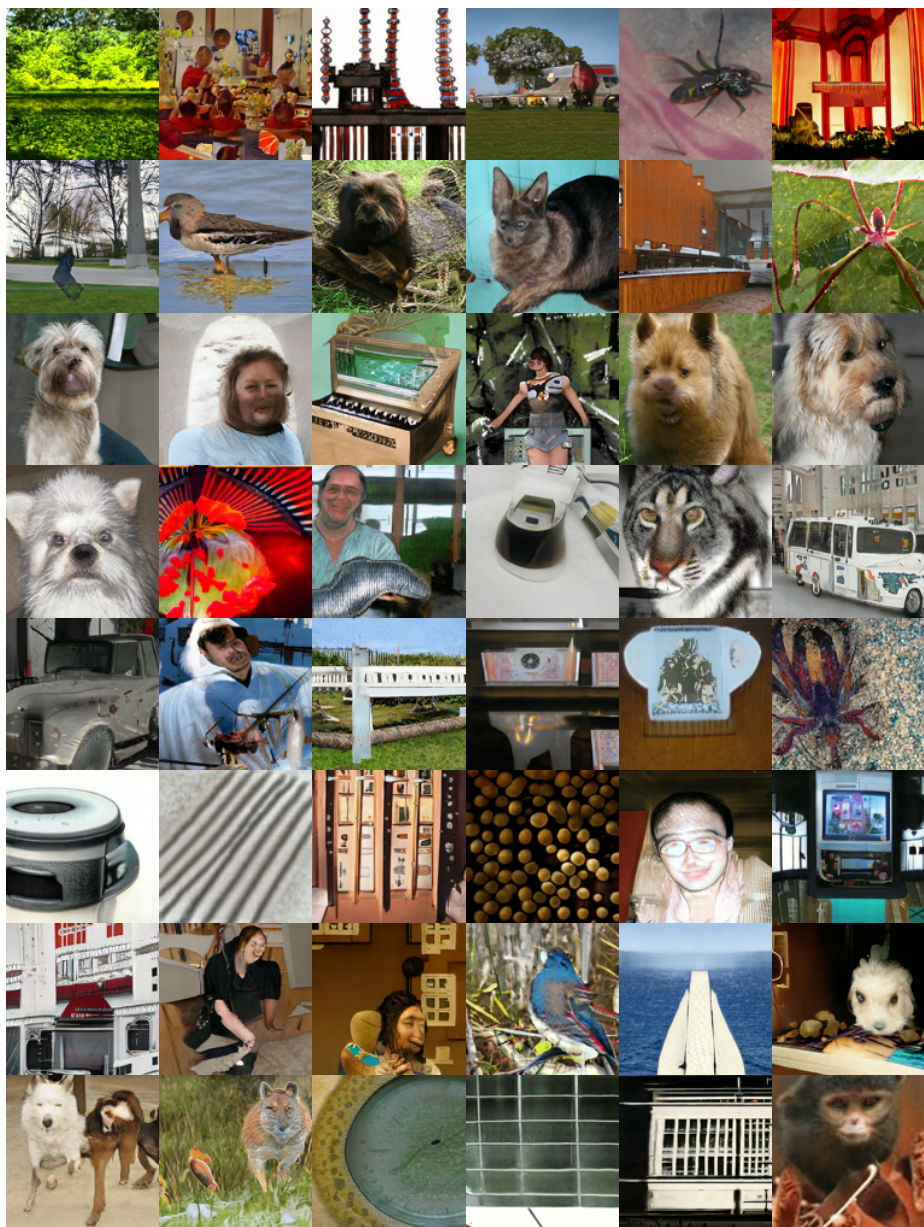


Fig. 8: Unconditional generation results on ImageNet-1k 128×128 .

Table 13: Complete performance comparison under ImageNet-1K only setting.

	Model	Eval size	Params	FLOPs	throughput (img/s)	ImageNet top-1 acc.
ConvNets	•EffNet-B3 [79]	300	12M	1.8G	732.1	81.6
	•EffNet-B4 [79]	380	19M	4.2G	349.4	82.9
	•EffNet-B5 [79]	456	30M	9.9G	169.1	83.6
	•EffNet-B6 [79]	528	43M	19.0G	96.9	84.0
	•EffNet-B7 [79]	600	66M	37.0G	55.1	84.3
	•RegNetY-8GF [62]	224	39M	8.0G	591.6	81.7
	•RegNetY-16GF [62]	224	84M	16.0G	334.7	82.9
	•NFNet-F0 [5]	256	72M	12.4G	533.3	83.6
	•NFNet-F1 [5]	320	132M	35.5G	228.5	84.7
	•NFNet-F2 [5]	352	194M	62.6G	129.0	85.1
	•NFNet-F3 [5]	416	255M	114.7G	78.8	85.7
	•NFNet-F4 [5]	512	316M	215.2G	51.7	85.9
	•NFNet-F5 [5]	544	377M	289.8G	-	86.0
	•EffNetV2-S [80]	384	24M	8.8G	666.6	83.9
	•EffNetV2-M [80]	380	55M	24.0G	280.7	85.1
	•EffNetV2-L [80]	480	121M	53.0G	163.2	85.7
	•ConvNeXt-T [57]	224	29M	4.5G	774.7	82.1
	•ConvNeXt-S [57]	224	50M	8.7G	447.1	83.1
	•ConvNeXt-B [57]	224	89M	15.4G	292.1	83.8
	•ConvNeXt-L [57]	384	198M	101.0G	50.4	85.5
ViTs	◊ViT-B/32 [22]	384	86M	55.4G	85.9	77.9
	◊ViT-B/16 [22]	384	307M	190.7G	27.3	76.5
	◊DeiT-S [81]	224	22M	4.6G	940.4	79.8
	◊DeiT-B [81]	224	86M	17.5G	292.3	81.8
	◊DeiT-B [81]	384	86M	55.4G	85.9	83.1
	◊CaiT-S36 [82]	224	68M	13.9G	-	83.3
	◊CaiT-M24 [82]	224	186M	36.0G	-	83.4
	◊CaiT-M24 [82]	384	186M	116.1G	-	84.5
	◊DeepViT-S [105]	224	27M	6.2G	-	82.3
	◊DeepViT-L [105]	224	55M	12.5G	-	83.1
	◊T2T-ViT-14 [101]	224	22M	6.1G	-	81.7
	◊T2T-ViT-19 [101]	224	39M	9.8G	-	82.2
	◊T2T-ViT-24 [101]	224	64M	15.0G	-	82.6
	◊Swin-T [56]	224	29M	4.5G	755.2	81.3
	◊Swin-S [56]	224	50M	8.7G	436.9	83.0
	◊Swin-B [56]	384	88M	47.0G	84.7	84.5
	◊CSwin-B [21]	224	78M	15.0G	250	84.2
	◊CSwin-B [21]	384	78M	47.0G	-	85.4
	◊Focal-S [99]	224	51M	9.1G	-	83.5
	◊Focal-B [99]	224	90M	16.0G	-	83.8
Hybrid	◊CvT-13 [93]	224	20M	4.5G	-	81.6
	◊CvT-21 [93]	224	32M	7.1G	-	82.5
	◊CvT-21 [93]	384	32M	24.9G	-	83.3
	◊CoAtNet-0 [19]	224	25M	4.2G	534.5	81.6
	◊CoAtNet-1 [19]	224	42M	8.4G	336.5	83.3
	◊CoAtNet-2 [19]	224	75M	15.7G	247.6	84.1
	◊CoAtNet-3 [19]	384	168M	107.4G	48.5	85.8
	◊CoAtNet-3 [19]	512	168M	203.1G	22.4	86.0
	◊MaxViT-T	224	31M	5.6G	349.6	83.62
	◊MaxViT-S	224	69M	11.7G	242.5	84.45
	◊MaxViT-B	224	120M	23.4G	133.6	84.95
	◊MaxViT-L	224	212M	43.9G	99.4	85.17
	◊MaxViT-T	384	31M	17.7G	121.9	85.24
	◊MaxViT-S	384	69M	36.1G	82.7	85.74
	◊MaxViT-B	384	120M	74.2G	45.8	86.34
	◊MaxViT-L	384	212M	133.1G	34.3	84.40
	◊MaxViT-T	512	31M	33.7G	63.8	85.72
	◊MaxViT-S	512	69M	67.6G	43.3	86.19
	◊MaxViT-B	512	120M	138.5G	24.0	86.66
	◊MaxViT-L	512	212M	245.4G	17.8	86.70

Table 14: Complete performance comparison for ImageNet-21K and JFT pre-trained models.

	Model	Eval size	Params	FLOPs	IN-1K top-1 acc.	
					21K→1K	JFT→1K
ConvNets	• BiT-R-101x3 [46]	384	388M	204.6G	84.4	
	• BiT-R-152x4 [46]	480	937M	840.5G	85.4	
	• EffNetV2-S [80]	384	24M	8.8G	85.0	
	• EffNetV2-M [80]	480	55M	24.0G	86.1	
	• EffNetV2-L [80]	480	121M	53.0G	86.8	
	• EffNetV2-XL [80]	512	208M	94.0G	87.3	
	• NFNet-F4+ [5]	512	527M	367G	-	89.20
	• ConvNeXt-B [57]	384	89M	45.1G	86.8	
	• ConvNeXt-L [57]	384	198M	101.0G	87.5	
	• ConvNeXt-XL [57]	384	350M	179.0G	87.8	
ViTs	◦ ViT-B/16 [22]	384	87M	55.5G	84.0	
	◦ ViT-L/16 [22]	384	305M	191.1G	85.2	
	◦ ViT-L/16 [22]	512	305M	364G	-	87.76
	◦ ViT-H/14 [22]	518	632M	1021G	-	88.55
	◦ HaloNet-H4 [84]	384	85M	-	85.6	
	◦ HaloNet-H4 [84]	512	85M	-	85.8	
	◦ Swin-B [56]	384	88M	47.0G	86.4	
	◦ Swin-L [56]	384	197M	103.9G	87.3	
	◦ SwinV2-B [56]	384	88M	-	87.1	
	◦ SwinV2-L [56]	384	197M	-	87.7	
	◦ CSwin-B [21]	384	78M	47.0G	87.0	
	◦ CSwin-L [21]	384	173M	96.8G	87.5	
Hybrid	◊ CvT-13 [93]	384	20M	16.0G	83.3	
	◊ CvT-21 [93]	384	32M	25.0G	84.9	
	◊ CvT-W24 [93]	384	277M	193.2G	87.7	
	◊ ResNet+ViT-L/16 [22]	384	330M	-	-	87.12
	◊ CoAtNet-2 [19]	384	75M	49.8G	87.1	
	◊ CoAtNet-3 [19]	384	168M	107.4G	87.6	
	◊ CoAtNet-4 [19]	384	275M	189.5G	87.9	
	◊ CoAtNet-2 [19]	512	75M	96.7G	87.3	
	◊ CoAtNet-3 [19]	512	168M	203.1G	87.9	88.81
	◊ CoAtNet-4 [19]	512	275M	360.9G	88.1	89.11
	◊ CoAtNet-5 [19]	512	688M	812G	-	89.77
	◊ MaxViT-B	384	119M	74.2G	88.24	88.69
	◊ MaxViT-L	384	212M	128.7G	88.32	89.12
	◊ MaxViT-XL	384	475M	293.7G	88.51	89.36
	◊ MaxViT-B	512	119M	138.3G	88.38	88.82
	◊ MaxViT-L	512	212M	245.2G	88.46	89.41
	◊ MaxViT-XL	512	475M	535.2G	88.70	89.53

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6836–6846 (2021) [4](#)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) [6](#), [16](#)
3. Bello, I., Fedus, W., Du, X., Cubuk, E.D., Srinivas, A., Lin, T.Y., Shlens, J., Zoph, B.: Revisiting resnets: Improved training and scaling strategies. Advances in Neural Information Processing Systems **34**, 22614–22627 (2021) [4](#)
4. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3286–3295 (2019) [4](#)
5. Brock, A., De, S., Smith, S.L., Simonyan, K.: High-performance large-scale image recognition without normalization. In: International Conference on Machine Learning. pp. 1059–1071. PMLR (2021) [8](#), [9](#), [23](#), [24](#)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020) [2](#)
7. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) [18](#)
8. Chen, L.H., Bampis, C.G., Li, Z., Norkin, A., Bovik, A.C.: Proxiqa: A proxy approach to perceptual optimization of learned image compression. IEEE Transactions on Image Processing **30**, 360–373 (2020) [3](#)
9. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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) [1](#)
10. Chen, Q., Zhang, W., Zhou, N., Lei, P., Xu, Y., Zheng, Y., Fan, J.: Adaptive fractional dilated convolution network for image aesthetics assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14114–14123 (2020) [12](#)
11. Chen, T., Lucic, M., Houlsby, N., Gelly, S.: On self modulation for generative adversarial networks. arXiv preprint arXiv:1810.01365 (2018) [20](#)
12. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12154–12163 (2019) [12](#)
13. Chen, W.T., Huang, Z.K., Tsai, C.C., Yang, H.H., Ding, J.J., Kuo, S.Y.: Learning multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: Toward a unified model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17653–17662 (2022) [3](#)
14. Chen, W., Du, X., Yang, F., Beyer, L., Zhai, X., Lin, T., Chen, H., Li, J., Song, X., Wang, Z., Zhou, D.: A simple single-scale vision transformer for object localization and instance segmentation. CoRR **abs/2112.09747** (2021), <https://arxiv.org/abs/2112.09747> [11](#)
15. Chen, W., Huang, W., Du, X., Song, X., Wang, Z., Zhou, D.: Auto-scaling vision transformers without training. arXiv preprint arXiv:2202.11921 (2022) [11](#)

16. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems* **34** (2021) [2](#), [4](#)
17. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882* (2021) [6](#)
18. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Gordon, G., Dunson, D., Dudík, M. (eds.) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 15, pp. 215–223. PMLR, Fort Lauderdale, FL, USA (11–13 Apr 2011), <https://proceedings.mlr.press/v15/coates11a.html> [4](#)
19. Dai, Z., Liu, H., Le, Q., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems* **34** (2021) [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [13](#), [15](#), [16](#), [23](#), [24](#)
20. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018) [1](#)
21. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652* (2021) [4](#), [8](#), [23](#), [24](#)
22. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020) [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [9](#), [16](#), [17](#), [23](#), [24](#)
23. d’Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: *International Conference on Machine Learning*. pp. 2286–2296. PMLR (2021) [4](#)
24. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 6824–6835 (2021) [4](#)
25. Fang, H.S., Sun, J., Wang, R., Gou, M., Li, Y.L., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 682–691 (2019) [18](#)
26. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems* **27** (2014) [4](#), [7](#), [12](#)
27. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. *Advances in Neural Information Processing Systems* **34** (2021) [4](#)
28. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. pp. 2980–2988 (2017). <https://doi.org/10.1109/ICCV.2017.322> [10](#), [18](#)
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016) [1](#), [3](#), [7](#), [11](#)
30. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: *European conference on computer vision*. pp. 630–645. Springer (2016) [15](#)
31. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016) [15](#)

32. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS. pp. 6629–6640 (2017) [11](#)
33. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180 (2019) [6](#), [17](#)
34. Hoang, Q., Nguyen, T.D., Le, T., Phung, D.: Mgan: Training generative adversarial nets with multiple generators. In: International conference on learning representations (2018) [12](#)
35. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) [1](#), [4](#), [6](#), [7](#)
36. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018) [3](#), [6](#), [7](#), [15](#)
37. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017) [1](#), [3](#)
38. Hudson, D.A., Zitnick, L.: Generative adversarial transformers. In: International Conference on Machine Learning. pp. 4487–4499. PMLR (2021) [4](#)
39. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015) [11](#), [15](#)
40. Jiang, Y., Chang, S., Wang, Z.: Transgan: Two pure transformers can make one strong gan, and that can scale up. Advances in Neural Information Processing Systems **34** (2021) [4](#), [5](#), [15](#)
41. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017) [21](#)
42. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020) [4](#), [21](#)
43. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5148–5157 (2021) [11](#), [12](#), [21](#)
44. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM Computing Surveys (CSUR) (2021) [4](#)
45. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [21](#)
46. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning. In: European conference on computer vision. pp. 491–507. Springer (2020) [9](#), [24](#)
47. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) [4](#)
48. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012) [1](#), [3](#), [7](#)
49. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019) [1](#)

50. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021) [2](#)
51. Li, Y., Jin, P., Yang, F., Liu, C., Yang, M.H., Milanfar, P.: Comisr: Compression-informed video super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2543–2552 (2021) [3](#)
52. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 936–944 (2017) [3](#), [10](#), [18](#)
53. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) [7](#), [10](#), [20](#)
54. Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. Advances in neural information processing systems **31** (2018) [12](#)
55. Liu, S., Wang, T., Bau, D., Zhu, J.Y., Torralba, A.: Diverse image generation via self-conditioned gans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14286–14295 (2020) [12](#)
56. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021) [2](#), [4](#), [5](#), [6](#), [8](#), [9](#), [11](#), [13](#), [14](#), [15](#), [16](#), [18](#), [23](#), [24](#)
57. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022) [1](#), [3](#), [7](#), [8](#), [9](#), [11](#), [14](#), [23](#), [24](#)
58. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) [3](#)
59. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) [21](#)
60. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018) [21](#)
61. Murray, N., Marchesotti, L., Perronnin, F.: Ava: A large-scale database for aesthetic visual analysis. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2408–2415. IEEE (2012) [7](#), [10](#), [12](#), [21](#)
62. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10428–10436 (2020) [8](#), [23](#)
63. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019) [1](#)
64. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34** (2021) [4](#)
65. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), <https://proceedings.neurips.cc/paper/2015/file/14bfa6bbb14875e45bba028a21ed38046-Paper.pdf> [10](#), [18](#)
66. Rogozhnikov, A.: Einops: Clear and reliable tensor manipulations with einstein-like notation. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=oapKSM2bcj6> [6](#)

67. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 1, 3
68. Sage, A., Agustsson, E., Timofte, R., Van Gool, L.: Logo synthesis and manipulation with clustered generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5879–5888 (2018) 12
69. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training GANs. In: NeurIPS (2016) 11
70. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018) 1, 15
71. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. arXiv preprint arXiv:1803.02155 (2018) 5, 15
72. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016) 20
73. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017) 2
74. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017) 1
75. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015) 1, 3
76. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016) 1
77. Talebi, H., Milanfar, P.: Nima: Neural image assessment. IEEE transactions on image processing 27(8), 3998–4011 (2018) 10, 12, 19, 21
78. Talebi, H., Milanfar, P.: Learning to resize images for computer vision tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 497–506 (2021) 3
79. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019) 7, 8, 12, 15, 23
80. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International Conference on Machine Learning. pp. 10096–10106. PMLR (2021) 7, 8, 9, 23, 24
81. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021) 2, 4, 8, 23
82. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 32–42 (2021) 4, 8, 23

83. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. arXiv preprint arXiv:2201.02973 (2022) [4](#), [13](#)
84. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12894–12904 (2021) [9](#), [15](#), [24](#)
85. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017) [1](#), [3](#), [4](#)
86. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: European Conference on Computer Vision. pp. 108–126. Springer (2020) [4](#), [6](#), [13](#), [17](#)
87. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021) [4](#)
88. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018) [2](#), [3](#)
89. Wang, Y., Ke, J., Talebi, H., Yim, J.G., Birkbeck, N., Adsumilli, B., Milanfar, P., Yang, F.: Rich features for perceptual quality assessment of ugc videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13435–13444 (2021) [3](#)
90. Whang, J., Delbracio, M., Talebi, H., Saharia, C., Dimakis, A.G., Milanfar, P.: Deblurring via stochastic refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16293–16303 (2022) [3](#)
91. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) [3](#)
92. Wu, F., Fan, A., Baevski, A., Dauphin, Y.N., Auli, M.: Pay less attention with lightweight and dynamic convolutions. arXiv preprint arXiv:1901.10430 (2019) [15](#)
93. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22–31 (2021) [4](#), [8](#), [9](#), [23](#), [24](#)
94. Xiao, T., Dollar, P., Singh, M., Mintun, E., Darrell, T., Girshick, R.: Early convolutions help transformers see better. *Advances in Neural Information Processing Systems* **34** (2021) [3](#), [4](#), [6](#), [13](#)
95. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) [11](#)
96. Xu, R., Tu, Z., Xiang, H., Shao, W., Zhou, B., Ma, J.: Cobevt: Cooperative bird’s eye view semantic segmentation with sparse transformers. arXiv preprint arXiv:2207.02202 (2022) [4](#)
97. Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.H., Ma, J.: V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. arXiv preprint arXiv:2203.10638 (2022) [4](#)
98. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9981–9990 (2021) [4](#)

99. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641 (2021) [2](#), [4](#), [8](#), [23](#)
100. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* **32** (2019) [1](#)
101. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 558–567 (2021) [4](#), [8](#), [23](#)
102. Zhai, X., Kolesnikov, A., Neil, H., Beyer, L.: Scaling vision transformers. arXiv preprint arXiv:2106.04560 (2021) [10](#)
103. Zhao, L., Zhang, Z., Chen, T., Metaxas, D., Zhang, H.: Improved transformer for high-resolution gans. *Advances in Neural Information Processing Systems* **34** (2021) [4](#), [11](#), [12](#), [13](#), [20](#)
104. Zhao, Z., Wu, Z., Zhuang, Y., Li, B., Jia, J.: Tracking objects as pixel-wise distributions. arXiv preprint arXiv:2207.05518 (2022) [3](#)
105. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886 (2021) [4](#), [8](#), [23](#)