Swin Transformer V2: Scaling Up Capacity and Resolution

Ze Liu[2*] Han Hu[1*†] Yutong Lin[3] Zhuliang Yao[4] Zhenda Xie[4] Yixuan Wei[4] Jia Ning[5]
Yue Cao[1] Zheng Zhang[1] Li Dong[1] Furu Wei[1] Baining Guo[1]
[1]Microsoft Research Asia [2]University of Science and Technology of China
[3]Xian Jiaotong University [4]Tsinghua University [5]Huazhong University of Science and Technology
{t-liuze,hanhu,t-yutonglin,t-zhuyao,t-zhxie,t-yixuanwei,v-jianing}@microsoft.com
{yuecao,zhez,lidong1,fuwei,bainguo}@microsoft.com

Abstract

We present techniques for scaling Swin Transformer [35] up to 3 billion parameters and making it capable of training with images of up to 1,536×1,536 resolution. By scaling up capacity and resolution, Swin Transformer sets new records on four representative vision benchmarks: ***84.0%*** *top-1 accuracy on ImageNet-V2 image classification,* ***63.1 / 54.4*** *box / mask mAP on COCO object detection,* ***59.9*** *mIoU on ADE20K semantic segmentation, and* ***86.8%*** *top-1 accuracy on Kinetics-400 video action classification.*

We tackle issues of training instability, and study how to effectively transfer models pre-trained at low resolutions to higher resolution ones. To this aim, several novel technologies are proposed: 1) a residual post normalization technique and a scaled cosine attention approach to improve the stability of large vision models; 2) a log-spaced continuous position bias technique to effectively transfer models pre-trained at low-resolution images and windows to their higher-resolution counterparts. In addition, we share our crucial implementation details that lead to significant savings of GPU memory consumption and thus make it feasible to train large vision models with regular GPUs. Using these techniques and self-supervised pre-training, we successfully train a strong 3 billion Swin Transformer model and effectively transfer it to various vision tasks involving high-resolution images or windows, achieving the state-of-the-art accuracy on a variety of benchmarks. Code is available at `https://github.com/microsoft/Swin-Transformer`.

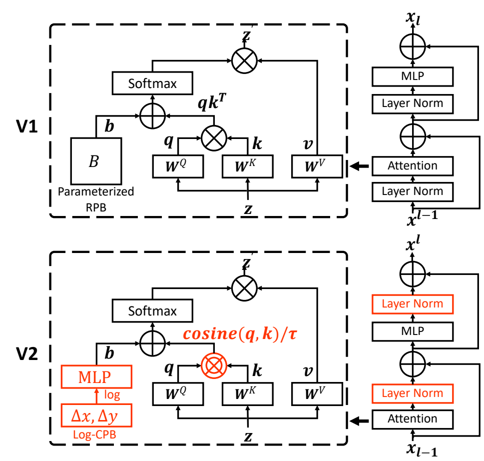

Figure 1. To better scale up model capacity and window resolution, several adaptions are made on the original Swin Transformer architecture (V1): 1) A *res-post-norm* to replace the previous *pre-norm* configuration; 2) A *scaled cosine attention* to replace the original *dot product attention*; 3) A *log-spaced continuous* relative position bias approach to replace the previous *parameterized* approach. Adaptions 1) and 2) make it easier for the model to scale up capacity. Adaption 3) makes the model to be transferred more effectively across window resolutions. The adapted architecture is named Swin Transformer V2.

1. Introduction

Scaling up language models has been incredibly successful. It significantly improves a model's performance on language tasks [12, 16, 37, 38, 40, 41] and the model demonstrates amazing few-shot capabilities similar to that of human beings [5]. Since the BERT large model with 340 million parameters [12], language models are quickly scaled up by more than 1,000 times in a few years, reaching 530 billion dense parameters [38] and 1.6 trillion sparse parameters [16]. These large language models are also found to possess increasingly strong few-shot capabilities akin to human intelligence for a broad range of language tasks [5].

On the other hand, the scaling up of vision models has been lagging behind. While it has long been recognized that larger vision models usually perform better on vision tasks [19, 48], the absolute model size was just able to reach

*Equal. †Project lead. Ze, Yutong, Zhuliang, Zhenda, Yixuan, Jia are long-term interns at MSRA.

about 1-2 billion parameters very recently [10,18,28,44,66]. More importantly, unlike large language models, the existing large vision models are applied to the image classification task only [10, 44, 66].

To successfully train large and general vision model, we need to address a few key issues. Firstly, our experiments with large vision models reveal an instability issue in training. We find that the discrepancy of activation amplitudes across layers becomes significantly greater in large models. A closer look at the original architecture reveals that this is caused by the output of the residual unit directly added back to the main branch. The result is that the activation values are accumulated layer by layer, and the amplitudes at deeper layers are thus significantly larger than those at early layers. To address this issue, we propose a new normalization configuration, called res-post-norm, which moves the LN layer from the beginning of each residual unit to the backend, as shown in Figure 1. We find this new configuration produces much milder activation values across the network layers. We also propose a scaled cosine attention to replace the previous dot product attention. The scaled cosine attention makes the computation irrelevant to amplitudes of block inputs, and the attention values are less likely to fall into extremes. In our experiments, the proposed two techniques not only make the training process more stable but also improve the accuracy especially for larger models.

Secondly, many downstream vision tasks such as object detection and semantic segmentation require high resolution input images or large attention windows. The window size variations between low-resolution pre-training and high-resolution fine-tuning can be quite large. The current common practice is to perform a bi-cubic interpolation of the position bias maps [15, 35]. This simple fix is somewhat ad-hoc and the result is usually sub-optimal. We introduce a log-spaced continuous position bias (Log-CPB), which generates bias values for arbitrary coordinate ranges by applying a small meta network on the log-spaced coordinate inputs. Since the meta network takes any coordinates, a pre-trained model will be able to freely transfer across window sizes by sharing weights of the meta network. A critical design of our approach is to transform the coordinates into the log-space so that the extrapolation ratio can be low even when the target window size is significantly larger than that of pre-training. The scaling up of model capacity and resolution also leads to prohibitively high GPU memory consumption with existing vision models. To resolve the memory issue, we incorporate several important techniques including zero-optimizer [42], activation check pointing [6] and a novel implementation of sequential self-attention computation. With these techniques, the GPU memory consumption of large models and resolutions is significantly reduced with only marginal effect on the training speed.

With the above techniques, we successfully trained a 3 billion Swin Transformer model and effectively transferred it to various vision tasks with image resolution as large as 1,536$\times$1,536, using Nvidia A100-40G GPUs. In our model pre-training, we also employ self-supervised pre-training to reduce the dependency on super-huge labeled data. With 40$\times$ less labelled data than that in previous practice (JFT-3B), the 3 billion model achieves the state-of-the-art accuracy on a broad range of vision benchmarks. Specifically, it obtains 84.0% top-1 accuracy on the ImageNet-V2 image classification validation set [43], 63.1 / 54.4 box / mask AP on the COCO test-dev set of object detection, 59.9 mIoU on ADE20K semantic segmentation, and 86.8% top-1 accuracy on Kinetics-400 video action classification, which are +NA%, +4.4/+3.3, +6.3 and +1.9 higher than the best numbers in the original Swin Transformers [35, 36], and surpass previous best records by +0.8% ([66]), +1.8/+1.4 ([61]), +1.5 ([3]) and +1.4% ([45]).

By scaling up both capacity and resolution of vision models with strong performance on general vision tasks, just like a good language model's performance on general NLP tasks, we aim to stimulate more research in this direction so that we can eventually close the capacity gap between vision and language models and facilitate the joint modeling of the two domains.

2. Related Works

Language networks and scaling up Transformer has served the standard network since the pioneer work of [52]. The exploration of scaling this architecture has since begun, and the progress has been accelerated by the invention of effective self-supervised learning approaches, such as masked or auto-regressive language modeling [12,40], and has been further encouraged by the discovery of a scaling law [25]. Since then, the capacity of language models has increased dramatically by more than 1,000 times in a few years, from BERT-340M to the Megatron-Turing-530B [5, 37, 38, 41] and sparse Switch-Transformer-1.6T [16]. With increased capacity, the accuracy of various language benchmarks has been significantly improved. The zero-shot or few-shot performance is also significantly improved [5], which is a foundation of human generic intelligence.

Vision networks and scaling up CNNs have long been the standard computer vision networks [29, 30]. Since AlexNet [29], architectures have become deeper and larger, which has greatly advanced various visual tasks and largely fueled the wave of deep learning in computer vision, such as VGG [48], GoogleNet [49] and ResNet citehe2015resnet. In the past two years, the CNN architectures have been further scaled up to about 1 billion parameters [18, 28], however, absolute performance may not be so encouraging, perhaps due to inductive biases in the CNN architecture limiting modeling power.

Last year, Transformers started taking over one representative visual benchmark after another, including ImageNet-1K image-level classification benchmarks [15], COCO region-level object detection benchmark [35], ADE20K pixel-level semantic segmentation benchmark [35, 68], Kinetics-400 video action classification benchmark [1], etc. Since these works, numerous vision Transformer variants have been proposed to improve the accuracy at relatively small scale [8, 14, 23, 31, 50, 55, 58, 62, 64, 65, 67]. Only a few works have attempted to scale up the vision Transformers [10,44,66]. However, they rely on a huge image dataset with classification labels, i.e., JFT-3B, and are only applied to image classification problems.

Transferring across window / kernel resolution For CNNs, previous works typically fixed kernel size during pre-training and fine-tuning. Global vision Transformers, such as ViT [15], compute attention globally, with the equivalent attention window size linearly proportional to the increased input image resolution. For local vision Transformer architectures, such as Swin Transformer [35], the window size can be either fixed or changed during fine-tuning. Allowing variable window sizes is more convenient in use, so as to be divisible by the probably variable entire feature map and to tune receptive fields for better accuracy. To handle the variable window sizes between pre-training and fine-tuning, bi-cubic interpolation was the previous common practice [15, 35]. In this paper, we propose a log-spaced continuous position bias approach (Log-CPB) that more smoothly transfers pre-trained model weights at low resolution to deal-with higher resolution windows.

Study on bias terms In NLP, the relative position bias method proved beneficial [41], compared to the absolute position embedding used in the original Transformer [52]. In computer vision, the relative positional bias method is more commonly used [21,35,62], probably because the spatial relationships of visual signals play a more important role in visual modeling. A common practice is to directly learn the bias values as model weights. There are also a few works particularly study how to set and learn the bias terms [27,56].

Continuous convolution and variants Our Log-CPB approach is also related to earlier works on continuous convolution and variants [20, 34, 46, 54], which utilize a meta network to handle irregular data points. Our Log-CPB approach is inspired by these efforts while solving a different problem of transferring relative position biases in vision Transformers across arbitrary window sizes. We also propose log-spaced coordinates to alleviate the difficulty of extrapolation when transferring between large size changes.

3. Swin Transformer V2

3.1. A Brief Review of Swin Transformer

Swin Transformer is a general-purpose computer vision backbone that has achieved strong performance in various granular recognition tasks such as region-level object detection, pixel-level semantic segmentation, and image-level image classification. The main idea of Swin Transformer is to introduce several important visual priors into the vanilla Transformer encoder, including hierarchy, locality, and translation invariance, which combines the strength of both: the basic Transformer unit has strong modeling capabilities, and the visual priors make it friendly to a variety of visual tasks.

Normalization configuration It is widely known that normalization technologies [2, 24, 51, 57] are crucial in stably training deeper architectures. The original Swin Transformer inherits the common practice in the language Transformers [40] and vanilla ViT [15] to utilize a pre-normalization configuration without extensive study, as shown in the figure 1. In the following subsections, we will examine this default normalization configuration[1].

Relative position bias is a key component in the original Swin Transformer which introduces an additional parametric bias term to encode the geometric relationship in self-attention calculation:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (1)$$

where $B \in \mathbb{R}^{M^2 \times M^2}$ is the relative position bias term for each head; $Q, K, V \in \mathbb{R}^{M^2 \times d}$ are the *query*, *key* and *value* matrices; d is the *query/key* dimension, and M^2 is the number of patches in a window. The relative position bias encodes relative spatial configurations of visual elements and is shown critical in a variety of visual tasks, especially for dense recognition tasks such as object detection.

In Swin Transformer, the relative positions along each axis are within the range of $[-M+1, M-1]$ and the relative position bias is parameterized as a bias matrix $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$, and the elements in B are taken from $\hat{B}$. When transferring across different window sizes, the learnt relative position bias matrix in pre-training is used to initialize the bias matrix of a different size in fine-tuning by bi-cubic interpolation.

[1]There have been a few alternative normalization configurations, such as post-normalization [52] and sandwich normalization [13]. Post-normalization harms training stability [60], and sandwich normalization sacrifices representation power due to too many normalization layers.

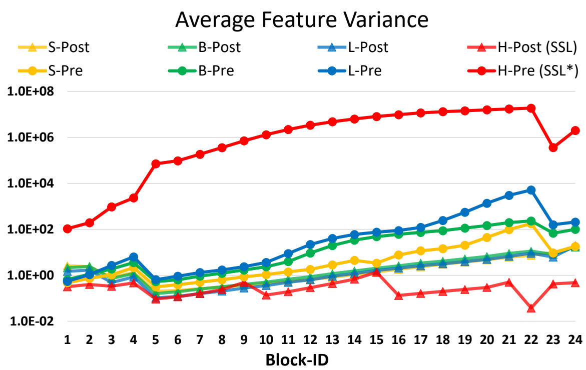

Figure 2. The Signal Propagation Plot [4, 63] for various model sizes. H-size models are trained at a self-supervised learning phase, and other sizes are trained by an image classification task. * indicates that we use a 40-epoch model before it crashes.

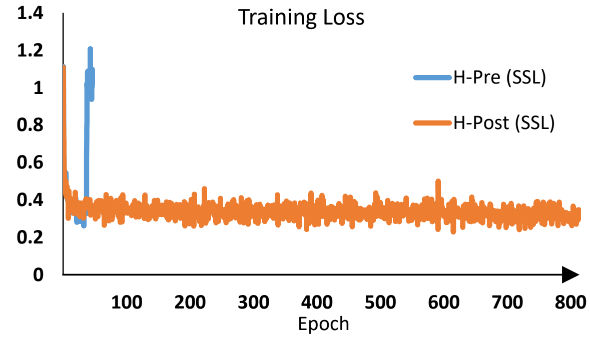

Figure 3. SwinV1-H versus SwinV2-H in training [59].

Issues in scaling up model capacity and window resolution We observe two issues when we scale up the capacity and window resolution of the Swin Transformer.

- *An instability issue when scaling up model capacity*. As shown in Figure 2, when we scale up the original Swin Transformer model from small size to large size, the activation values at deeper layers increase dramatically. The discrepancy between layers with the highest and the lowest amplitudes has reached an extreme value of 10^4. When we scale it up further to a huge size (658 million parameters), it cannot complete the training, as shown in Figure 3.
- *Degraded performance when transferring models across window resolutions*. As shown in the first row of Table 1, the accuracy decreases significantly when we directly test the accuracy of a pre-trained ImageNet-1K model (256×256 images with 8×8 window size) at larger image resolutions and window sizes through the bi-cubic interpolation approach. It may be worth re-examining the relative position bias approach in the original Swin Transformer.

In the following subsections, we present techniques to address these issues, including *residual post normalization* and *scaled cosine attention* to address the instability issue, and a *log-spaced continuous position bias* approach to address the issue in transferring across window resolutions.

3.2. Scaling Up Model Capacity

As mentioned in Section 3.1, the original Swin Transformer (and most vision Transformers) adopts a layer norm layer at the beginning of each block, inherited from vanilla ViT. When we scale up the model capacity, a significant increase in activation values is observed at deeper layers. In fact, in a pre-normalization configuration, the output activation values of each residual block are merged directly back to the main branch, and the amplitude of the main branch grows larger and larger at deeper layers. Large amplitude discrepancy in different layers causes training instability.

Post normalization To ease this problem, we propose to use a *residual post normalization* approach instead, as shown in Figure 1. In this approach, the output of each residual block is normalized before merging back into the main branch, and the amplitude of the main branch does not accumulate when the layer goes deeper. As shown in Figure 2, the activation amplitudes by this approach are much milder than in the original pre-normalization configuration.

In our largest model training, we introduce an additional layer normalization layer on the main branch every 6 Transformer blocks, to further stabilize training.

Scaled cosine attention In the original self-attention computation, the similarity terms of the pixel pairs are computed as a dot product of the *query* and *key* vectors. We find that when this approach is used in large visual models, the learnt attention maps of some blocks and heads are frequently dominated by a few pixel pairs, especially in the *res-post-norm* configuration. To ease this issue, we propose a *scaled cosine attention* approach that computes the attention logit of a pixel pair i and j by a scaled cosine function:

$$\mathrm{Sim}(\mathbf{q}_i, \mathbf{k}_j) = \cos(\mathbf{q}_i, \mathbf{k}_j)/\tau + B_{ij}, \tag{2}$$

where B_{ij} is the relative position bias between pixel i and j; τ is a learnable scalar, non-shared across heads and layers. τ is set larger than 0.01. The cosine function is naturally normalized, and thus can have milder attention values.

3.3. Scaling Up Window Resolution

In this subsection, we introduce a log-spaced continuous position bias approach, so that the relative position bias can be smoothly transferred across window resolutions.

method	ImageNet*	ImageNet†				COCO		ADE20k		
	W8, I256 top-1 acc	W12, I384 top-1 acc	W16, I512 top-1 acc	W20, I640 top-1 acc	W24, I768 top-1 acc	W16 AP^{box}	W32 AP^{box}	W16 mIoU	W20 mIoU	W32 mIoU
Parameterized position bias [35]	81.7	79.4/82.7	77.2/83.0	73.2/83.2	68.7/83.2	50.8	50.9	45.5	45.8	44.5
Linear-Spaced CPB	81.7 (+0.0)	82.0/82.9 (+2.6/+0.2)	81.2/83.3 (+4.0/+0.3)	79.8/83.6 (+6.6/+0.4)	77.6/83.7 (+8.9/+0.5)	50.9 (+0.1)	51.7 (+0.8)	47.0 (+1.5)	47.4 (+1.6)	47.2 (+2.7)
Log-Spaced CPB	81.8 (+0.1)	82.4/83.2 (+3.0/+0.5)	81.7/83.8 (+4.5/+0.8)	80.4/84.0 (+7.2/+0.8)	79.1/84.2 (+10.4/+1.0)	51.1 (+0.3)	51.8 (+0.9)	47.0 (+1.5)	47.7 (+1.9)	47.8 (+3.3)

Table 1. Comparison of different position bias computation approaches using Swin-T. * indicates the top-1 accuracy on ImageNet-1k trained from scratch. The models in * column will be used for testing on the ImageNet-1K image classification task using larger image/window resolutions, marked by †. For these results, we report both the results w.o./with fine-tuning. These models are also used for fine-tuning on COCO object detection and ADE20K semantic segmentation tasks.

Continuous relative position bias Instead of directly optimizing the parameterized biases, the *continuous* position bias approach adopts a small meta network on the relative coordinates:

$$B(\Delta x, \Delta y) = \mathcal{G}(\Delta x, \Delta y), \tag{3}$$

where $\mathcal{G}$ is a small network, e.g., a 2-layer MLP with a ReLU activation in between by default.

The meta network $\mathcal{G}$ generates bias values for arbitrary relative coordinates, and thus can be naturally transferred to fine-tuning tasks with arbitrarily varying window sizes. In inference, the bias values at each relative position can be pre-computed and stored as model parameters, such that the inference is the same as the original parameterized bias approach.

Log-spaced coordinates When transferring across largely varying window sizes, a large portion of the relative coordinate range needs to be extrapolated. To ease this issue, we propose using log-spaced coordinates instead of the original linear-spaced ones:

$$\begin{aligned} \widehat{\Delta x} &= \text{sign}(x) \cdot \log(1 + |\Delta x|), \\ \widehat{\Delta y} &= \text{sign}(y) \cdot \log(1 + |\Delta y|), \end{aligned} \tag{4}$$

where Δx, Δy and $\widehat{\Delta x}$, $\widehat{\Delta y}$ are the linear-scaled and log-spaced coordinates, respectively.

By using the log-spaced coordinates, when we transfer the relative position biases across window resolutions, the required extrapolation ratio will be much smaller than that of using the original linear-spaced coordinates. For an example of transferring from a pre-trained 8×8 window size to a fine-tuned 16×16 window size, using the original raw coordinates, the input coordinate range will be from $[-7, 7] \times [-7, 7]$ to $[-15, 15] \times [-15, 15]$. The extrapolation ratio is $\frac{8}{7} = 1.14\times$ of the original range. Using log-spaced coordinates, the input range will be from $[-2.079, 2.079] \times [-2.079, 2.079]$ to $[-2.773, 2.773] \times [-2.773, 2.773]$. The extrapolation ratio is $0.33\times$ of the original range, which is an about 4 times smaller extrapolation ratio than that using the original linear-spaced coordinates.

Table 1 compares the transferring performance of different position bias computation approaches. It can be seen that the log-spaced CPB (continuous position bias) approach performs best, particularly when transferred to larger window sizes.

3.4. Self-Supervised Pre-training

Larger models are more data hungry. To address the data hungry problem, previous large vision models typically utilize huge labelled data such as JFT-3B [10, 44, 66]. In this work, we exploit a self-supervised pre-training method, SimMIM [59], to alleviate the demands on labelled data. By this approach, we successfully trained a powerful Swin Transformer model of 3 billion parameters which achieves state-of-the-art (SOTA) on 4 representative visual benchmarks, by using only 70 million labelled images (1/40 of that in JFT-3B).

3.5. Implementation to Save GPU Memory

Another issue lies in the unaffordable GPU memory consumption with a regular implementation when both the capacity and resolution are large. To facility the memory issue, we adopt the following implementations:

- *Zero-Redundancy Optimizer (ZeRO)* [42]. In a general data-parallel implementation of optimizers, the model parameters and optimization states are broadcasted to every GPU. This implementation is very unfriendly on GPU memory consumption, for example, a model of 3 billion parameters will consume 48G GPU memory when an AdamW optimizer and fp32 weights/states are used. With a ZeRO optimizer, the model parameters and the corresponding optimization states will be split and distributed to multiple GPUs, which significantly reduces memory consumption. We adopt the DeepSpeed framework and use the ZeRO stage-1 option in our experiments. This optimization has little effect on training speed.

- *Activation check-pointing* [6]. Feature maps in the Transformer layers also consume a lot of GPU memory, which can create bottlenecks when image and window resolutions are high. The activation check-pointing technology can significantly reduce the memory consumption, while the training speed is up to 30% slower.
- *Sequential self-attention computation*. To train large models on very large resolutions, for example, an image of 1,536×1,536 resolution with a window size of 32×32, regular A100 GPUs (40GB memory) are still unaffordable, even with the above two optimization technologies. We found that in this case, the self-attention module constitutes a bottleneck. To alleviate this problem, we implement self-attention computation sequentially, instead of using the previous batch computation approach. This optimization is applied to the layers in the first two stages and has little impact on the overall training speed.

With these implementations, we managed to train a 3B model using the Nvidia A100-40G GPUs for COCO object detection with an input image resolution of 1,536×1,536, and Kinetics-400 action classification with an input resolution of $320 \times 320 \times 8$.

3.6. Model configurations

We maintain the stage, block, and channel settings of the original Swin Transformer for 4 configurations of Swin Transformer V2:

- SwinV2-T: C = 96, #. block = $\{2, 2, 6, 2\}$
- SwinV2-S/B/L: C=96/128/192, #.block=$\{2, 2, 18, 2\}$

with C the number of channels in the first stage.

We further scale up Swin Transformer V2 to its huge size and giant size, with 658 million parameters and 3 billion parameters, respectively:

- SwinV2-H: $C = 352$, #. block = $\{2, 2, 18, 2\}$
- SwinV2-G: $C = 512$, #. block = $\{2, 2, 42, 4\}$

For SwinV2-H and SwinV2-G, we add an additional layer normalization layer on the main branch every 6 layers. To save experimental time, we only employ SwinV2-G for large-scale experiments. SwinV2-H is employed for another parallel study about self-supervised learning [59].

4. Experiments

4.1. Tasks and Datasets

We conduct experiments on ImageNet-1K image classification (V1 and V2) [11, 43], COCO object detection [33], and ADE20K semantic segmentation [69]. For the 3B model experiments, we also report the accuracy on Kinetics-400 video action recognition [26].

- *Image classification*. ImageNet-1K V1 and V2 val are used [11,43] for evaluation. ImageNet-22K [11] which has 14M images and 22K categories is optionally employed for pre-training. For the pre-training our largest model SwinV2-G, a privately collected ImageNet-22K-ext dataset with 70 million images is used. For this dataset, a duplicate removal process [39] is conducted to exclude overlapping images with ImageNet-1K V1 and V2 validation sets.
- *Object detection*. COCO [33] is used for evaluation. For our largest model experiments, we employ an additional detection pre-training phase using Object 365 v2 dataset [47], in-between the image classification pre-training phase and the COCO fine-tuning phase.
- *Semantic segmentation*. ADE20K [69] is used.
- *Video action classification*. Kinetics-400 (K400) [26] is used in evaluation.

The pre-training and fine-tuning settings will be detailed in Appendix.

4.2. Scaling Up Experiments

We first present the results on various representative visual benchmarks by scaling up models to 3 billion parameters and to high image/window resolutions.

Settings for SwinV2-G experiments We adopt a smaller 192×192 image resolution in pre-training to save on training costs. We take a 2-step pre-training approach. First, the model is pre-trained using a self-supervised method [59] on the ImageNet-22K-ext dataset by 20 epochs. Second, the model is further pre-trained by 30 epochs using the image classification task on this dataset. Detailed pre-training and fine-tuning setups are described in the appendix.

In the following paragraphs, we report the accuracy of SwinV2-G on representative vision benchmarks. Note that since our main goal is to explore how to feasibly scale up model capacity and window resolution, and whether the vision tasks can benefit from significantly larger capacity, we did not particularly align complexities or pre-training data in comparisons.

ImageNet-1K image classification results Table 2 compares the SwinV2-G model with previously largest/best vision models on ImageNet-1K V1 and V2 classification. SwinV2-G is the largest dense vision model to present. It achieves a top-1 accuracy of 84.0% on the ImageNet V2 benchmark, which is +0.7% higher than previous best one

Method	param	pre-train images	pre-train length (#im)	pre-train im size	pre-train time	fine-tune im size	ImageNet-1K-V1 top-1 acc	ImaegNet-1K-V2 top-1 acc
SwinV1-B	88M	IN-22K-14M	1.3B	224^2	$<30^\dagger$	384^2	86.4	76.58
SwinV1-L	197M	IN-22K-14M	1.3B	224^2	$<10^\dagger$	384^2	87.3	77.46
ViT-G [66]	1.8B	JFT-3B	164B	224^2	>30k	518^2	90.45	83.33
V-MoE [44]	14.7B*	JFT-3B	-	224^2	16.8k	518^2	90.35	-
CoAtNet-7 [10]	2.44B	JFT-3B	-	224^2	20.1k	512^2	**90.88**	-
SwinV2-B	88M	IN-22K-14M	1.3B	192^2	$<30^\dagger$	384^2	87.1	78.08
SwinV2-L	197M	IN-22K-14M	1.3B	192^2	$<20^\dagger$	384^2	87.7	78.31
SwinV2-G	3.0B	IN-22K-ext-70M	3.5B	192^2	$<0.5k^\dagger$	640^2	90.17	**84.00**

Table 2. Comparison with previous largest vision models on ImageNet-1K V1 and V2 classification. * indicates the sparse model; the "pre-train time" column is measured by the TPUv3 core days with numbers copied from the original papers. † That of SwinV2-G is estimated according to training iterations and FLOPs.

Method	train I(W) size	test I(W) size	mini-val (AP) box	mini-val (AP) mask	test-dev (AP) box	test-dev (AP) mask
CopyPaste [17]	1280(-)	1280(-)	57.0	48.9	57.3	49.1
SwinV1-L [35]	800(7)	ms(7)	58.0	50.4	58.7	51.1
YOLOR [53]	1280(-)	1280(-)	-	-	57.3	-
CBNet [32]	1400(7)	ms(7)	59.6	51.8	60.1	52.3
DyHead [9]	1200(-)	ms(-)	60.3	-	60.6	-
SoftTeacher [61]	1280(12)	ms(12)	60.7	52.5	61.3	53.0
SwinV2-L (HTC++)	1536(32)	1100(32)	58.8	51.1	-	-
		1100 (48)	58.9	51.2	-	-
		ms (48)	60.2	52.1	60.8	52.7
SwinV2-G (HTC++)	1536(32)	1100(32)	61.7	53.3	-	-
		1100 (48)	61.9	53.4	-	-
		ms (48)	**62.5**	**53.7**	**63.1**	**54.4**

Table 3. Comparison with previous best results on COCO object detection and instance segmentation. I(W) indicates the image and window size. ms indicate multi-scale testing is employed.

Method	train I(W) size	test I(W) size	mIoU
SwinV1-L [35]	640(7)	640(7)	53.5*
MaskFormer [7]	640(7)	640(7)	55.6*
FaPN [22]	640(7)	640(7)	56.7*
BEiT [3]	640(40)	640(40)	58.4*
SwinV2-L (UperNet)	640(40)	640(40)	55.9*
SwinV2-G (UperNet)	640(40)	640(40)	59.1
		896 (56)	59.3
		896 (56)	**59.9***

Table 4. Comparison with previous best results on ADE20K semantic segmentation. * indicates multi-scale testing is used.

(83.3%). Our accuracy on ImageNet-1K V1 is marginally lower (90.17% vs 90.88%). The performance difference might come from different degrees of dataset over-tuning [43]. Also note we employ much less training iterations and lower image resolutions than those in previous efforts, while performing very well.

We also compare the SwinV2-B and SwinV2-L to the original SwinV1-B and SwinV1-L, respectively, where a +0.8% and +0.4% gains are observed. The shrunken gains by SwinV2-L than that of SwinV2-B may imply that if exceeding this size, more labeled data, stronger regularization, or advanced self-supervised learning methods are required.

Method	train I(W) size	test I(W) size	views	top-1
ViViT [1]	-(-)	-(-)	4×3	84.8
SwinV1-L [36]	$480(12)^2\times16(8)$	$480(12)^2\times16(8)$	10×5	84.9
TokenLearner [45]	$256(8)^2\times64(64)$	$256(8)^2\times64(64)$	4×3	85.4
Video-SwinV2-G	$320(20)^2\times8(8)$	$320(20)^2\times8(8)$	1×1	83.2
		$384(24)^2\times8(8)$	1×1	83.4
		$384(24)^2\times8(8)$	4×5	**86.8**

Table 5. Comparison with previous best results on Kinetics-400 video action classification.

COCO object detection results Table 3 compares the SwinV2-G model with previous best results on COCO object detection and instance segmentation. It achieves 63.1/54.4 box/max AP on COCO test-dev, which is +1.8/1.4 higher than previous best numberw (61.3/53.0 by [61]). This suggests that scaling up vision model is beneficial for the dense vision recognition task of object detection. Our approach can use a different window size at test to additionally benefit, probably attributed to the effective Log-spaced CPB approach.

ADE20K semantic segmentation results Table 4 compares the SwinV2-G model with previous best results on the ADE20K semantic segmentation benchmark. It achieves 59.9 mIoU on ADE20K val set, +1.5 higher than the previous best number (58.4 by [3]). This suggests scaling up vision model is beneficial for pixel-level vision recognition tasks. Using a larger window size at test time can additionally bring +0.2 gains, probably attributed to the effective Log-spaced CPB approach.

Kinetics-400 video action classification results Table 5 compares the SwinV2-G model with previous best results on the Kinetics-400 action classification benchmark. It achieves 86.8% top-1 accuracy, +1.4% higher than previous

Backbone	res-post-norm	scaled cosine attention	ImageNet top-1 acc
Swin-T			81.5
	✓		81.6
	✓	✓	**81.7**
Swin-S			83.2
	✓		83.3
	✓	✓	**83.6**
Swin-B			83.6
	✓		83.8
	✓	✓	**84.1**
ViT-B			82.2
	✓	✓	**82.6**

Table 6. Ablation on res-post-norm and cosine attention.

Backbone	pre-norm	sandwich [13]	post-norm [52]	our
Swin-S	83.2	82.6	83.3	**83.6**
Swin-B	83.6	-	83.6	**84.1**

Table 7. Comparison with other normalization methods. The post-norm method diverges at the default learning rate, and we use 1/4 of the default learning rate for this method. Sandwich performs worse than ours, probably because it sacrifices expressiveness.

best number [45]. This suggests that scaling up vision models also benefits video recognition tasks. In this scenario, using a larger window size at test time can also bring additional benefits of +0.2%, probably attributed to the effective Log-spaced CPB approach.

4.3. Ablation Study

Ablation on res-post-norm and scaled cosine attention Table 6 ablates the performance of applying the proposed res-post-norm and scaled cosine attention approaches to Swin Transformer. Both techniques improve the accuracy at all the tiny, small and base size, and the overall improvements are +0.2%, +0.4% and +0.5% respectively, indicating the techniques are more beneficial for larger models. It also turns out to benefit ViT architecture (+0.4%). The proposed normalization approach also performs better than some other normalization methods, as shown in Table 7.

More importantly, the combination of post-norm and scaled cosine attention stabilize the training. As shown in Figure 2, while the activation values at deeper layers for the original Swin Transformer are almost exploded at large (L) size, those of the new version have much milder behavior. On a huge size model, the self-supervised pre-training [59] diverges using the original Swin Transformer, while it trains well by a Swin Transformer V2 model.

Scaling up window resolution by different approaches Table 1 and 8 ablate the performance of 3 approaches by scaling window resolutions from 256×256 in pre-training to larger sizes in 3 down-stream vision tasks of ImageNet-1K image classification, COCO object detection,

Backbone	L-CPB	ImageNet* W8, I256	ImageNet† W12, I384	ImageNet† W16, I512
SwinV2-S		83.7	81.8/84.5	79.4/84.9
	✓	83.7	84.1/84.8	82.9/85.4
SwinV2-B		84.1	82.9/85.0	81.0/85.3
	✓	84.2	84.5/85.1	83.8/85.6

Table 8. Ablation on Log-CPB using different model sizes.

and ADE20K semantic segmentation, respectively. It can be seen that: 1) Different approaches have similar accuracy in pre-training (81.7%-81.8%); 2) When transferred to down-stream tasks, the two continuous position bias (CPB) approaches perform consistently better than the parameterized position bias approach used in Swin Transformer V1. Compared to the linear-spaced approach, the log-spaced version is marginally better; 3) The larger the change in resolutions between pre-training and fine-tuning, the larger the benefit of the proposed log-spaced CPB approach.

In Table 1 and 8, we also report the accuracy using targeted window resolutions without fine-tuning (see the first number in each column in the ImageNet-1K experiments). The recognition accuracy remains not bad even when the window size is enlarged from 8 to 24 (78.9% versus 81.8%), while the top-1 accuracy of the original approach significantly degrades from 81.7% to 68.7%. Also note that without fine-tuning, using a window size of 12 that the pre-trained model has never seen before can even be +0.4% higher that the original accuracy. This suggests that we can improve accuracy through test-time window adjustment, as also observed in Table 3, 4 and 5.

5. Conclusion

We have presented techniques for scaling Swin Transformer up to 3 billion parameters and making it capable of training with images of up to 1,536×1,536 resolution, including the *res-post-norm* and *scaled cosine attention* to make the model easier to be scaled up in capacity, as well a log-spaced continuous relative position bias approach which lets the model more effectively transferred across window resolutions. The adapted architecture is named Swin Transformer V2, and by scaling up capacity and resolution, it sets new records on 4 representative vision benchmarks. By these strong results, we hope to stimulate more research in this direction so that we can eventually close the capacity gap between vision and language models and facilitate the joint modeling of the two domains.

Acknowledgement

We thank many colleagues at Microsoft for their help, in particular, Eric Chang, Lidong Zhou, Jing Tao, Aaron Zhang, Edward Cui, Bin Xiao, Lu Yuan, Peng Cheng, Fan Yang for useful discussion and the help on GPU resources and datasets.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021. 3, 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 3

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers, 2021. 2, 7

[4] Andrew Brock, Soham De, and Samuel L Smith. Characterizing signal propagation to close the performance gap in unnormalized resnets. *arXiv preprint arXiv:2101.08692*, 2021. 4

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 1, 2

[6] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost, 2016. 2, 6

[7] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv*, 2021. 7

[8] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers, 2021. 3

[9] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions, 2021. 7

[10] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes, 2021. 2, 3, 5, 7

[11] 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. 6

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

[13] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. *arXiv preprint arXiv:2105.13290*, 2021. 3, 8

[14] 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, 2021. 3

[15] 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. In *International Conference on Learning Representations*, 2021. 2, 3

[16] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021. 1, 2

[17] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. *arXiv preprint arXiv:2012.07177*, 2020. 7

[18] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild, 2021. 2

[19] 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

[20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 3

[21] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3464–3473, October 2019. 3

[22] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction, 2021. 7

[23] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer, 2021. 3

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 3

[25] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. 2

[26] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 6

[27] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training, 2021. 3

[28] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019. 2

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[31] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers, 2021. 3

[32] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. Cbnetv2: A composite backbone network architecture for object detection, 2021. 7

[33] 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. 6

[34] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis, 2020. 3

[35] 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, 2021. 1, 2, 3, 5, 7

[36] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021. 2, 7

[37] Microsoft. Turing-nlg: A 17-billion-parameter language model by microsoft, 2020. 1, 2

[38] Microsoft. Using deepspeed and megatron to train megatron-turing nlg 530b, the world's largest and most powerful generative language model, 2021. 1, 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 6

[40] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1, 2, 3

[41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 1, 2, 3

[42] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models, 2020. 2, 5

[43] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019. 2, 6, 7

[44] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts, 2021. 2, 3, 5, 7

[45] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos?, 2021. 2, 7, 8

[46] Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*, 2017. 3

[47] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015. 1, 2

[49] 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. 2

[50] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 3

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. 3

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2, 3, 8

[53] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks, 2021. 7

[54] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 3

[55] 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, 2021. 3

[56] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer, 2021. 3

[57] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[58] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021. 3

[59] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Tech report*, 2022. 4, 5, 6, 8

[60] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. 2020. 3

[61] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher, 2021. 2, 7

[62] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers, 2021. 3

[63] Zhuliang Yao, Yue Cao, Yutong Lin, Ze Liu, Zheng Zhang, and Han Hu. Leveraging batch normalization for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 413–422, 2021. 4

[64] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet, 2021. 3

[65] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021. 3

[66] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers, 2021. 2, 3, 5, 7

[67] 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, 2021. 3

[68] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 3

[69] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018. 6