Swin Transformer V2: Scaling Up Capacity and Resolution

Ze Liu* Han Hu*† Yutong Lin Zhuliang Yao Zhenda Xie Yixuan Wei Jia Ning
Yue Cao Zheng Zhang Li Dong Furu Wei Baining Guo
Microsoft Research Asia
{v-zeliu1,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. Our techniques are generally applicable for scaling up vision models, which has not been widely explored as that of NLP language models, partly due to the following difficulties in training and applications: 1) vision models often face instability issues at scale and 2) many downstream vision tasks require high resolution images or windows and it is not clear how to effectively transfer models pre-trained at low resolutions to higher resolution ones. The GPU memory consumption is also a problem when the image resolution is high. To address these issues, we present several techniques, which are illustrated by using Swin Transformer as a case study: 1) a 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 will be available at https://*

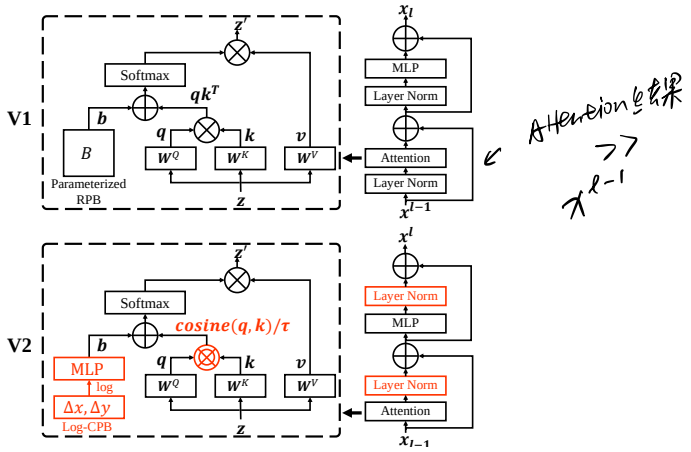

Figure 1. In order to better scale model capacity and window resolution, several adaptions are made on the original Swin Transformer architecture (V1): 1) a *post-norm* to replace the previous *pre-norm* configuration; 2) a *scaled cosine attention* to replace the original *dot product attention*; 3)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

github.com/microsoft/Swin-Transformer.

1. Introduction

Scaling up language models has been incredibly successful. It significantly improves a model's performance on language tasks [13, 16, 37, 38, 40, 41] and the model demonstrates amazing few-shot capabilities similar to that of human beings [6]. Since the BERT large model with 340 million parameters [13], 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

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

possess increasingly strong few-shot capabilities akin to human intelligence for a broad range of language tasks [6].

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 about 1-2 billion parameters very recently [11, 18, 28, 44, 65]. More importantly, unlike large language models, the existing large vision models are applied to the image classification task only [11, 44, 65].

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 post-norm, which moves the LN layer from the beginning of each residual unit to the back-end, 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 [7] 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 1,536×1,536, 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× 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% ([65]), +1.8/+1.4 ([60]), +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 Transformers serve the standard network since a pioneer work of [52]. The scaling of this architecture started with that, and the progress was speed up by the findings of effective self-supervised learning approaches such as masked or auto-regressive language modeling [13, 40], and was further encouraged by the findings of a scaling law [25]. Since then, the capacities of language models increase dramatically by more than 1,000 times within a few years, from BERT-340M to the Megatron-Turing-530B [6, 37, 38, 41] and to the sparse Switch-Transformer-1.6T [16]. With increased capabilities, the accuracy on various language benchmarks is also improved significantly. The significantly increased capabilities also encourage the paradigms of zero-shot or few-shot learning [6], which are closer to how human intelligence works.

Vision networks and scaling up CNNs for a long time are the standard computer vision networks [29, 30]. Since AlexNet [29], the architectures become deeper and

larger, which advance various vision tasks significantly, and largely propel the deep learning wave in computer vision, e.g., VGG [48], GoogleNet [49], and ResNet [19]. In recent two years, the CNN architectures are further scaled up to about 1 billion parameters [18, 28], however, the absolute performance is not that encouraging probably, perhaps due to the modeling power limited by the inductive bias in the CNN architectures. During the last year, Transformers started to take over one after another representative vision benchmarks including the image-level classification benchmark of ImageNet-1K [15], the region-level benchmark of COCO object detection [35], the pixel-level semantic segmentation benchmark of ADE20K [35,67], the video action classification benchmark of Kinetics-400 [1] and etc. Numerous vision Transformer variants were proposed to improve the accuracy at relatively small scale [9,14,23,31,50, 55,58,61,63,64,66]. However, only a few works attempted to scale the vision Transformers by leveraging a huge labelled image dataset [11, 44, 65], i.e., JFT-3B. The scaled models are also applied to the image classification problem only [11, 44, 65].

Transferring across window / kernel resolution For CNNs, previous works usually fix kernel size during pre-training and fine-tuning. The global vision Transformers such as ViT compute attention globally, with the equivalent attention window size linearly proportional to increased input image resolutions. For local vision Transformer architectures such as Swin Transformer [35], the window size can be either made fixed or varied during fine-tuning. Allowing varied window size is more convenient, e.g., to be divisible by the whole feature maps, and can also help achieve better accuracy. To deal with varied window size between pre-training and fine-tuning, a previous common practice is to use bi-cubic interpolation [15, 35]. In this paper, we present a log-spaced continuous position bias approach (Log-CPB), which can more smoothly transfer model weights pre-trained at low-resolution to deal with higher-resolution ones.

Study on bias terms In NLP, while absolute position embedding is used in the original Transformer, the relative position bias approach is later proved beneficial [4, 41]. In computer vision, the relative position bias approach is even more commonly used [21,35,61], probably because the spatial relationship of visual signals plays a more important role for vision modeling. A common practice is to directly learn the bias values as model weights, while with a few works studying the bias terms specially [27,56].

Continuous convolution and variants Our Log-CPB approach is also related to early works on continuous convolution and the variants [20, 34, 46, 54], which leverage a meta network to process irregular data points. Our Log-CPB approach is inspired by these works, while addressing a different problem of transferring relative position biases in vision Transformers across arbitrary window sizes. We additionally propose log-spaced coordinates to ease the extrapolation issue in transferring between large size variations.

3. Swin Transformer V2

3.1. A Brief Review of Swin Transformer

Swin Transformer is a general-purpose computer vision backbone, and it achieves strong performance on recognition tasks of various granularity, including the region-level object detection, the pixel-level semantic segmentation and the image-level image classification. The main idea of Swin Transformer is to introduce several important visual signal priors into the vanilla Transformer encoder architecture, including hierarchy, locality and translation invariance, which unions the strength of both: the basic Transformer unit possesses strong modeling capability, and the visual signal priors make it friendly to a variety of vision tasks.

Normalization configuration It is widely known that the normalization techniques [2, 24, 51, 57] are crucial in training deeper architectures as well as stabilizing the training process. The original Swin Transformer inherits the common practice in language Transformers and the vanilla ViT to leverage a pre-normalization configuration, as shown in Figure 1, without extensive study. In the following subsections, we will examine this design.

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

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

where $B \in \mathbb{R}^{M^2 \times M^2}$ is the relative position bias term to 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 accounts for relative spatial configurations of visual elements, and is shown critical in various vision tasks, particularly for the dense recognition tasks such as object detection.

In Swin Transformer, the relative position along each axis lies in 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 values 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 a bi-cubic interpolation approach.

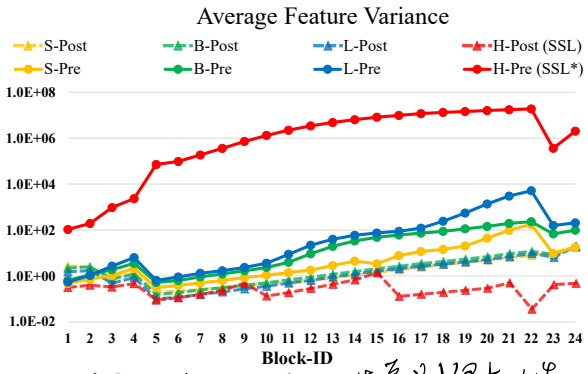

Figure 2. Signal Propagation Plot [5, 62] for various model sizes. The H-size models are trained at a self-supervised learning stage and other sizes are trained by the classification task. * indicates that we use a 40-epoch model before it crashes.

Issues in scaling up model capacity and window resolution We observe two issues in scaling the capacity and window resolution of 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 to large size, the activation values at deeper layers grow dramatically. The discrepancy between layers with the highest and the lowest amplitudes has reached an extreme of 10^4. When we further scale it up to a huge size (658 million parameters), it cannot accomplish the training, as shown in Figure 3.

- *Degraded performance when transferring the models across window resolutions.* As shown by the first row of Table 1, when we directly test the accuracy of a pre-trained ImageNet-1K model (256×256 images with 8×8 window size) on a larger image resolution and window size by the bi-cubic interpolation approach, the accuracy significantly drops. It may worth re-examine the relative position bias approach in the original Swin Transformer.

In the following subsections, we present techniques to address the above issues, including *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 described in Section 3.1, the original Swin Transformer (as well as most vision Transformers) adopts pre-normalization at the beginning of each block, inheriting from the vanilla ViT. It is observed with dramatically increased activation values at deeper layers when we scale up

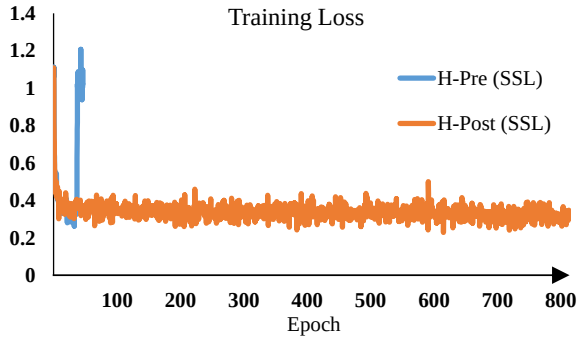

Figure 3. SwinV1-H versus SwinV2-H in training [59].

the model capacity. In fact, in the pre-normalization configuration, the output activation values of each residual block are directly merged back to the main branch, and the amplitudes of the main branch will be larger and larger at deeper layers. Large amplitude discrepancy in different layers may cause a training instability issue.

Post normalization To ease this problem, we propose to use a *post normalization* approach instead, as illustrated in Figure 1. In this approach, the output of each residual block is normalized before merged back to the main branch, and the amplitudes of the main branch will not be accumulated when layers go deeper. As shown in Figure 2, the activation amplitudes by this approach become much milder than in the original pre-normalization configuration.

In our largest model training, we additionally introduce a layer normalization unit on the main branch every 6 Transformer blocks, to further stabilize training and the amplitudes.

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

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

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, to make the relative position bias

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.

smoothly transferable across window resolution.

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 varied window sizes. In inference, the bias value at each relative position can be precomputed and stored as model parameters, such that it is the same convenient at inference than the original parameterized bias approach.

Log-spaced coordinates When transferred across largely varied window sizes, there will be a large portion of relative coordinate range requiring extrapolation. To ease this issue, we propose to use the 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 log-spaced coordinates, when we transfer relative position biases across window resolution, the required extrapolation ratio will be much less 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 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 size.

3.4. Other Implementation

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]. Regular optimizer implementations for the data-parallel mode broadcast model parameters and optimization states to every GPU or a master node. This is very unfriendly for large models, for example, a model of 3 billion parameters will consume 48G GPU memory when an AdamW optimizer and fp32 weights/states are used. By a ZeRO optimizer, the model parameters and the corresponding optimization states will be divided and distributed to multiple GPUs, and thus the memory consumption is significantly reduced. We adopt the DeepSpeed framework and use the ZeRO stage-1 option in our experiments. This optimization has little affect on training speed.

- *Activation check-pointing* [7]. The feature maps in Transformer layers also consume a lot of GPU memory, which can constitute a bottleneck when the image and window resolution is high. This optimization will reduce training speed by at most 30%.

- *Sequential self-attention computation*. To train large-scale models on very large resolutions, e.g., 1,536×1,536 images with a 32×32 window size, even

after employing the above two optimization strategies, it is still unaffordable for regular GPUs (40GB memory). We find the self-attention modules constitute a bottleneck in this case. To ease this issue, we implement the self-attention computation sequentially, instead of using the previous batch computation approach. This optimization is applied on layers in the first two stages, and has little affect on the overall training speed.

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

Joining with a self-supervised approach Larger model is more data hungry. To address the data hungry issue, previous large vision models usually either leverage huge labelled data such as JFT-3B [11, 44, 65] or self-supervised pre-training [18]. In this work, we combine both strategies: on the one hand, we moderately enlarge the ImageNet-22K datasets by 5 times to reach 70 million images with noisy labels; while this data scale is still far behind that of JFT-3B, we additionally employ a self-supervised learning approach [59] to better exploit this data. By combining the two strategies, we train a strong Swin Transformer model of 3 billion parameters, and achieve the state-of-the-art accuracy on several representative vision benchmarks.

3.5. Model configurations

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

- SwinV2-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- SwinV2-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$
- SwinV2-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$
- SwinV2-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

with C the channel number of hidden layers 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: $C = 352$, layer numbers = $\{2, 2, 18, 2\}$
- SwinV2-G: $C = 512$, layer numbers = $\{2, 2, 42, 2\}$

For SwinV2-H and SwinV2-G, we further introduce a layer normalization unit on the main branch every 6 layers. To save experimental time, we only employ SwinV2-G for the large-scale experiments on various vision tasks. SwinV2-H is employed for our another parallel study on self-supervised learning [59].

Add LN on main branch every 6 layers

4. Experiments

4.1. Tasks and Datasets

We conduct experiments on ImageNet-1K image classification (V1 and V2) [12, 43], COCO object detection [33], and ADE20K semantic segmentation [68]. For the 3B model experiments, we also report its accuracy on Kinetics-400 video action recognition [26].

- *Image classification*. ImageNet-1K V1 and V2 val are employed [12, 43] for evaluation. ImageNet-22K [12] which has 14M images and 22K categories is optionally employed for pre-training. A privately collected ImageNet-22K-ext dataset with 70M images with a duplicate removal process for IN-1K V1/V2 images [39] is used for pre-training our largest model.
- *Object detection*. COCO [33] is used for evaluation. For our largest model experiments, we employ the Object 365 v2 dataset [47] for detection pre-training after the image classifcation pre-traiing and before fine-tuning on COCO.
- *Semantic segmentation*. ADE20K [68] 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 A smaller 192×192 image resolution is adopted in pre-training to save the training cost. We employ a 2-step pre-training approach. Firstly, the model is pre-trained using a self-supervised approach [59] on the ImageNet-22K-ext dataset for 20 epochs. Secondly, the model is further pre-trained for 30 epochs using the classification task on this dataset. The pre-training and fine-tuning settings will be detailed in Appendix.

In the following paragraphs, we report the accuracy of SwinV2-G on representative vision benchmarks. Note 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 previous largest/best vision models on ImageNet-1K V1 and V2 classification. SwinV2-G is the largest among all previous dense vision models. It achieves 84.0% top-1 accuracy on ImageNet V2 benchmark, which is +0.7% higher than previous best one (83.3%). Nevertheless, 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 resolution than previous works, while performs strong.

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 shrinked gain by SwinV2-L than that of SwinV2-B may imply more labelled data, stronger regularization, or advanced self-supervised learning approaches are required if beyond this size.

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 number (61.3/53.0 by [60]). This indicates 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 bring gains, 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 ADE20K semantic segmentation benchmark. It achieves 59.9 mIoU on ADE20K val set, which is +1.5 higher than the previous best number (58.4 by [3]). This indicates 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, which is +1.4% higher than previous best number [45]. This indicates scaling up vision model is beneficial for video recognition tasks also. In this scenario, using a larger window size at test time can also additionally bring gains (+0.2%), probably attributed to the effective Log-spaced CPB approach.

4.3. Ablation Study

Ablation on post-norm and scaled cosine attention Table 6 ablates the performance of applying the proposed post-norm and scaled cosine attention approaches to the original Swin Transformer approaches. Both techniques improve the accuracy at all of 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.

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 ablates the performance of 3 approaches by scaling window resolutions from the 256×256 in pre-training to larger sizes in 3 down-stream vision tasks of ImageNet-1K image classification, COCO object detection, and ADE20K semantic segmentation, respectively. It can be seen: 1) different approaches have similar accuracy in pre-training (81.7%-81.8%); 2) when transferred to down-sream tasks, the two continuous position bias (CPB) approaches perform consistently better than the parameterzied position bias approach used in the original Swin Transformer. The log-spaced version is marginally better compared to the linear-spaced approach; 3) the larger the resolution change between pre-training and fine-tuning, the larger the benefit by the proposed log-spaced CPB approach.

In Table 1, we also report the accuracy on the targeted window resolutions without fine-tuning (see the first number for each column in the ImageNet-1K experiments). It can be seen that the recognition accuracy maintains not bad even when the window size is enlarged from 8 to 24 (78.9% versus 81.8%), while that of the original approach degrades significantly 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 can even outperforms that at the original accuracy by +0.4%. This indicates that we may improve the accuracy through test-time window adjustment, which is also observed by 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 *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

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 [65]	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 [11]	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 [10]	1200(-)	ms(-)	60.3	-	60.6	-
SoftTeacher [60]	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*
Focal-L [61]	640(40)	640(40)	55.4*
CSwin-L [14]	640(40)	640(40)	55.7*
MaskFormer [8]	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.

sets new recordson 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 onADE20K semantic segmentation, and 86.8% top-1 accuracy on Kinetics-400 video action classification. 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.

Method	train I(W) size	test I(W) size	views	top-1
ViViT [1]	-(-)	-(-)	4×3	84.8
SwinV1-L [36]	480×480×16 (12×12×8)	480×480×16 (12×12×8)	10×5	84.9
TokenLearner [45]	256×256×64 (8×8×64)	256×256×64 (8×8×64)	4×3	85.4
Video-SwinV2-G	320×320×8 (20×20×8)	320×320×8 (20×20×8)	1×1	83.2
		384×384×8 (24×24×8)	1×1	83.4
		384×384×8 (24×24×8)	4×5	**86.8**

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

Backbone	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

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

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 7. Ablation on Log-CPB using different model sizes.

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 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, 8

[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, 8

[4] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR, 2020. 3

[5] 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

[6] 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

[7] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost, 2016. 2, 5

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv*, 2021. 8

[9] 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

[10] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions, 2021. 8

[11] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes, 2021. 2, 3, 6, 8

[12] 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

[13] 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

[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, 8

[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. 8

[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, 3, 6

[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. 2, 3

[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. 8

[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, 3

[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. 8

[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, 8

[36] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021. 2, 8

[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

[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, 6, 8

[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. 2, 3

[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. 3

[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

[53] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks, 2021. 8

[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 trans-

formers 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, 6, 7

[60] 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, 8

[61] 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, 8

[62] 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

[63] 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

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

[65] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers, 2021. 2, 3, 6, 8

[66] 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

[67] 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

[68] 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