This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ParCNetV2: Oversized Kernel with Enhanced Attention*

Ruihan Xu[1,2], Haokui Zhang[2,3,‡], Wenze Hu[2], Shiliang Zhang[1,‡], Xiaoyu Wang[2]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]Intellifusion [3]Harbin Institute of Technology (Shenzhen)

Abstract

*Transformers have shown great potential in various computer vision tasks. By borrowing design concepts from transformers, many studies revolutionized CNNs and showed remarkable results. This paper falls in this line of studies. Specifically, we propose a new convolutional neural network, **ParCNetV2**, that extends the research line of ParCNetV1 by bridging the gap between CNN and ViT. It introduces two key designs: 1) **O**versized **C**onvolution (**OC**) with twice the size of the input, and 2) **B**ifurcate **G**ate **U**nit (**BGU**) to ensure that the model is input adaptive. Fusing OC and BGU in a unified CNN, ParCNetV2 is capable of flexibly extracting global features like ViT, while maintaining lower latency and better accuracy. Extensive experiments demonstrate the superiority of our method over other convolutional neural networks and hybrid models that combine CNNs and transformers. The code are publicly available at https://github.com/XuRuihan/ParCNetV2.*

1. Introduction

Transformers have shown great potential in computer vision recently. ViT [14] and its variants [52, 62, 55, 37] have been adopted to various vision tasks such as object detection [3, 15], semantic segmentation [67], and multi-modal tasks such as visual question answering [29] and text-to-image synthesis [42]. Despite the great performance of vision transformers, they do not win CNNs in all aspects. For example, the computational complexity of self-attention modules, one of the critical designs in transformers, is quadratic ($\mathcal{O}(N^2C)$) to the resolution of inputs [53]. This property restricts its adoption in real applications such as defect inspection, which finds small defects in high-resolution images [65]. Moreover, transformers are arguably more data-hungry than CNNs [14, 52, 21, 17], making them difficult to deploy to long-tail applications without large-scale data. Lastly, CNNs have been intensively studied in the past several decades [30]. There are lots of off-the-shelf dedicated features already developed in existing deployment hardware (CPU, GPU, FPGA, ASIC, *etc.*). Some acceleration and deployment techniques are designed mainly around convolution operations, such as operator fusion [45] and multi-level tiling [66, 6].

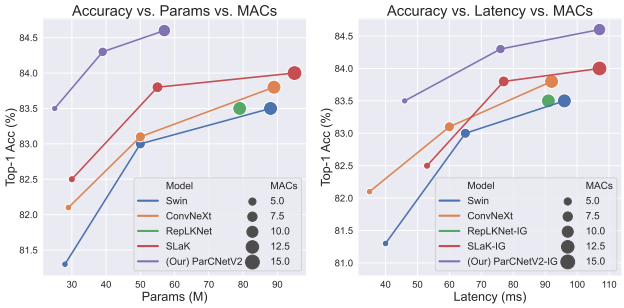
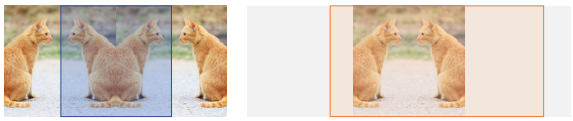


Figure 1. Comparison between ParCNetV2 with the prevailing transformer (Swin), CNN (ConvNeXt), and large kernel CNNs (RepLKNet & SLaK) when trained from scratch on ImageNet-1K. Left: performance curve of model size vs. top-1 accuracy. Right: performance curve of inference latency vs. top-1 accuracy. **IG** represents using the *implicit gemm* acceleration algorithm.

Thus pushing the envelope of CNNs is still important and valuable. Recent works have improved CNNs from multiple perspectives. A straightforward approach is to take the benefits from both CNNs and transformers by mixing their building blocks [18, 49, 39, 7, 34]. While bringing together merits from the two parties, those approaches still keep the ViT blocks and has the quadratic complexity problem. Another line of research is to design purely convolutional architectures. For example, with larger convolution kernels, ConvNeXt [38], RepLKNet [12], and ParCNetV1 [64] successfully improved the performance of CNNs by encoding broader spatial contexts.

Specifically, ParCNetV1 introduced **p**osition-**a**ware ci**r**cular **c**onvolutions (ParC) to CNNs. It uses depth-wise circular 1D convolutions of input feature map size ($C \times H \times 1$ and $C \times 1 \times W$) to achieve global receptive fields. To avoid spatial over-smoothing caused by global kernels, Par-

*Work was done when R. Xu was an intern at Intellifusion. ‡ denotes corresponding author.



a) Circular convolution b) Oversized convolution

Figure 2. **Comparison between circular convolution and oversized convolution.** We only show horizontal convolution for illustration purposes. a) Circular convolution in ParCNetV1 inevitably distorts context information at the boundary of images. b) Oversized convolution resolves the distortion while maintaining the global receptive field over the whole image.

CNetV1 augmented the feature input with absolute position encoding to ensure the feature output is still location sensitive. It also brought attention mechanisms into the framework by adopting squeeze-and-excitation block [27]. These modifications lead to the superior performance of ParCNetV1, especially on mobile devices.

Despite improved model efficiency and accuracy, ParCNetV1 still suffers from some design drawbacks. Firstly, as mentioned in [64] and shown in Fig 2, the circular padding introduces spatial distortion by performing convolutions crossing image borders. Secondly, the attention design is relatively weak compared with transformers which may limit the framework performance. Thirdly, it is not feasible to apply global convolution to all blocks in CNNs, especially those shallow blocks due to expensive computational costs and over-smoothing effects.

To address these issues, we propose a pure convolutional neural network architecture called ParCNetV2. It is composed of three essential improvements over ParCNetV1.

First, we push the kernel size to the extreme by doubling the circular convolution kernel and removing the absolute positional encoding. As shown in Fig. 2, through large size (equal to the size of the input) padding, the convolution operation avoids feature distortion around image borders. By using constant paddings, the oversized kernel implicitly encodes spatial locations when it convolves with the feature maps [28]. It enables us to discard the positional encoding module without hurting network performance. We explain why $2\times$ is the extreme in Sec.3.1.

Second, the original ParC block uses a limited attention mechanism inserted at the end of the channel mixing phase. We propose a more flexible bifurcate gate unit (BGU) at both the token mixing phase (spatial BGU) and channel mixing phase (channel BGU) in our newly designed block. Compared to the squeeze-and-excitation block, the BGU is stronger while more compact and general to combine with various structures, leading to spatial attention and channel attention. The enhanced attention mechanism also simplifies our ParC V2 block, as both phases adopt the consistent BGU structure.

Last, in contrast to ParCNetV1 which applies large kernel convolutions only on later-stage CNN blocks, we unify the block design by mixing large kernel convolutions with local depth-wise convolutions in all the blocks. Both types of convolutions are operated on the input feature map channels. This progressive design combines local features and global features in one convolution step, unlike many other works that stack the two sequentially [18, 60, 64] or as two separate branches [7, 39, 9]. To this end, the resulting redesigned ParC V2 structure is capable of performing local convolutions, global convolutions, token channel mixing, and BGU-based attention all in one block.

To summarize, the main contributions of this paper are as follows:

- We propose oversized convolutions for the effective modeling of long-range feature interactions in CNNs. Compared to ParCNetV1, it enables homogeneous convolution across all spatial locations, while removes the need for extra position encoding.
- We propose two bifurcate gate units (spatial BGU and channel BGU), which are compact and powerful attention modules. They boost the performance of ParCNetV2 and could be easily integrated into other network structures.
- We bring oversized convolution to shallow layers of CNNs and unify the local-global convolution design across blocks.

Extensive experiments are conducted to demonstrate that ParCNetV2 outperforms all other CNNs given a similar amount of parameters and computation budgets as shown in Fig. 1. It also beats state-of-the-art ViTs and CNN-ViT hybrids, which indicates that convolution networks are as strong as transformers in extracting features.

2. Related Works

Convolution Networks. Before transformers were introduced to vision tasks, convolutional neural networks had dominated vision architectures in a variety of computer vision tasks [22, 44, 5, 20, 32]. ResNet [22] introduced residual connections to eliminate network degradation, enabling very deep convolutional networks. It has been a strong baseline in various vision tasks. Inception [51] focuses on the multi-branch structure and utilizes kernel decomposition on small kernels. MobileNets [26, 46, 25] introduced depth separable convolution to build a lightweight convolution model. After the appearance of vision transformers, researchers improved pure convolution networks with ideas from transformers. RepLKNet [12] increased kernel size to as large as 31×31, which can extract long-range dependencies in contrast to commonly used 3×3 kernels. ConvNeXt [38] reviewed the design of the vision transformers

and gradually modernized a standard ResNet toward a transformer. ParCNet [64] proposed a pure convolution network with position-aware circular convolution, which achieved better performance than popular light-weight CNNs and vision transformers.

Vision Transformers. ViT [14] is the first transformer network in computer vision. It cropped images into 16×16 patches as input tokens and used positional encoding to learn spatial information. However, the vanilla ViT was hard to train and huge datasets are required such as JFT-300M [50]. DeiT [52] exploited knowledge distillation to train ViT models and achieved competitive accuracy with less pretraining data. To further enhance the model architecture, some researchers attempted to optimize ViTs with ideas from CNNs. T2T-ViT [62] introduced a token-to-token process to progressively tokenize images to tokens and structurally aggregate tokens. PVT [55] inserted convolution into each stage of ViT to reduce the number of tokens and build hierarchical multi-stage structures. Swin transformer [37] computed self-attention among shifted local windows, which has become the new baseline of many vision tasks. CSWin[13] adopted cross-attention to enlarge the receptive field of local attention. PiT [23] jointly used pooling layers and depth-wise convolution layers to achieve channel multiplication and spatial reduction. Yu *et al*. [61] pointed out that the general architecture of the transformers, MetaFormer, is more essential to the model's performance instead of the specific token mixer module. Some other works focus on efficient transformers [16, 7, 34].

Hybrid Convolution Networks and Vision Transformers. In addition to ViTs, another popular line of research is to combine elements of ViTs and CNNs to absorb the strengths of both architectures. LeViT [18] proposed a hybrid neural network for fast inference and significantly outperformed existing CNNs and ViTs concerning the speed/accuracy trade-off. BoTNet [49] replaces the standard convolutions with multi-head attention in the final three bottleneck blocks of ResNet. CvT [58] introduced depth-wise and point-wise convolution in front of the self-attention unit, which introduced shift, scale, and distortion invariance while maintaining the merits of transformers. Some other works focused on improving efficiency with hybrid models. CMT [19] combined a convolutional inverted feed-forward network with a lightweight multi-head self-attention way and took advantage of transformers to capture long-range dependencies and CNN to model local features. MobileViT [39] proposed a lightweight model and a fast training strategy for mobile devices. CabViT [63] enhanced the interactions of tokens across blocks, which encourages more information flows to the lower levels.

Although many works have successfully combined transformers and CNNs for vision tasks, they are not as much focused as our work on the systematic design of the global receptive field, advanced attention mechanism, and unified local-global balance across the whole network. We invent a newly evolved version of these designs and demonstrate the potential of pure CNNs compared with transformers and hybrid architectures.

3. Methods

An overview of the ParCNetV2 architecture is presented in Fig. 3. Compared with the original ParCNet (Fig. 3a), we first substitute the position-aware circular convolution with oversized convolution to encode long-range dependencies along with position information (Fig. 3b). Then we introduce bifurcate gate units as a stronger attention mechanism (Fig. 3c). Finally, we propose a uniform block that balances local and global convolutions to build full ParCNetV2 (Fig. 3d). The following sections describe the details of these components.

3.1. Oversized convolution

In ParCNetV1, the model is divided into two branches, alternating the order of vertical and horizontal convolution. However, we find that changing the order does not affect the output (proof in supplementary), thus we keep only one branch for simplicity. To further enhance the model's capacity and incorporate long-range spatial context, we introduce an oversized depth-wise convolution with a kernel size approximately twice the input feature size (ParC-O-H and ParC-O-W), as illustrated in Fig. 3b. In this section, we provide details about the oversized convolution and discuss its effectiveness, efficiency, and adaptability.

Formulation: We denote the input feature map as $X \in \mathcal{R}^{C\times H\times W}$, where C, H, and W represent the number of channels, height, and width of X, respectively. The kernel weight for vertical and horizontal oversized convolution is $k^h \in \mathcal{R}^{C\times(2H-1)\times 1}$ and $k^w \in \mathcal{R}^{C\times 1\times(2W-1)}$. We let index 0 denote the center point of k^h and k^w. As shown in Fig. 4, we choose this size because it naturally covers the global receptive field at each position, and keeps the output size the same as the input without requiring any post-processing. In contrast, smaller kernels can not simultaneously preserve position cues and provide a global receptive field, while larger kernels need post-processing to adjust the output size.

To compute the output of the oversized convolution $Z_{i,j}$ at location (i, j), we use the following equations:

$$Y_{i,j} = \sum_{s=-(H-1)}^{H-1} k_s^h X_{i+s,j}, \tag{1}$$

$$Z_{i,j} = \sum_{t=-(W-1)}^{W-1} k_t^w Y_{i,j+t}, \tag{2}$$

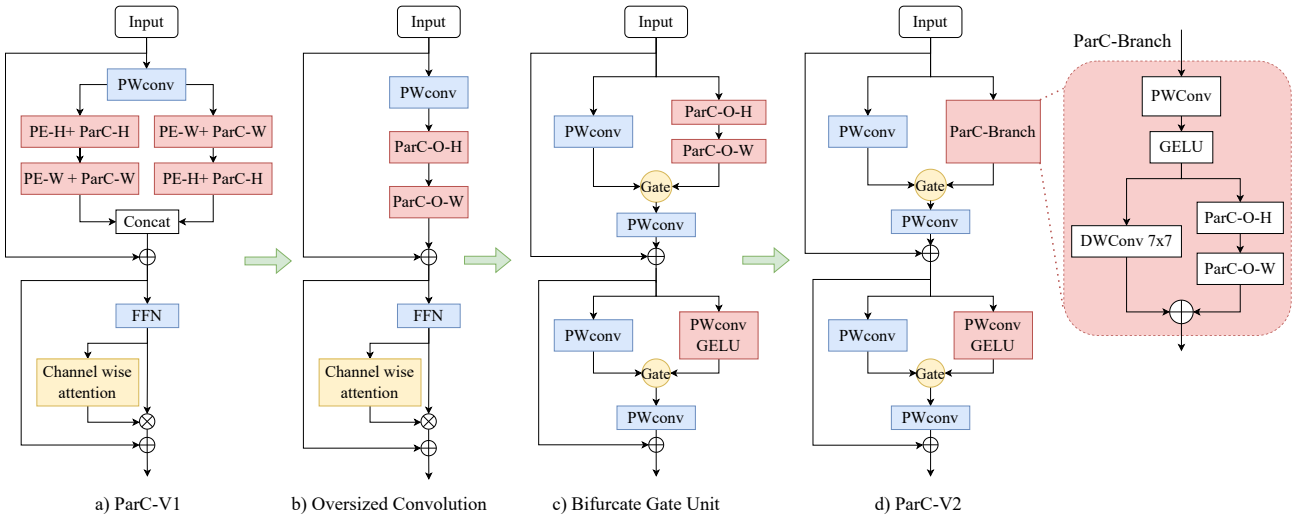


Figure 3. **The transitions from the original ParC V1 to ParC V2 block.** Compared with ParCNetV1, we first introduce oversized convolutions to further enhance capacity while simplifying architecture; then we design a bifurcate gate unit to improve efficiency and strengthen attention; finally, we propose a uniform local-global block and construct the whole network with this uniform block.

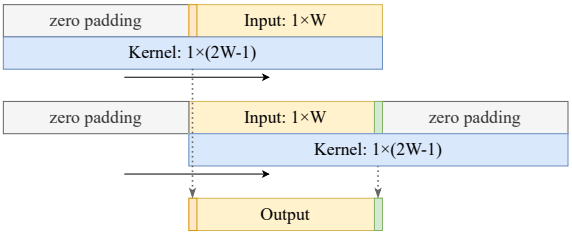


Figure 4. **Illustration of the oversized convolution.** Kernels are almost twice the size of input feature maps, and zero-padding is applied to keep the output resolution the same as the input.

where Eq. (1) denotes ParC-O-H, and Eq. (2) denotes ParC-O-W. Zero-padding means that $X_{i,j} = 0$ and $Y_{i,j} = 0$, if $i \notin [0, H-1]$ or $j \notin [0, W-1]$.

The padding operation is designed to work with oversized convolution, which encodes not only global dependency but also position information. For the horizontal convolution, we apply $W-1$ pixels zero padding to both left and right sides, where W is the width of the input feature. Similar operations are performed for vertical convolution. This schema keeps the output feature size the same as the input feature, and implicitly encodes position cues by zeroing out partial convolution kernel parameters according to spatial locations.

Effectiveness: The oversized convolution brings two advantages. First, it encodes position information by embedding it into each location using zero-padding, eliminating the need for position embeddings. As shown in Fig. 4, each position in the output is transformed by different parameters across the input features, and thus embeds position information in the model weights. It is similar to relative position embeddings [47], while the oversized convolution encodes both spatial context and position information in kernel weights. As a result, position embeddings are no longer required and therefore abandoned to make the network more concise.

Second, it improves model capacity with limited computational complexity. For instance, the largest oversized kernel in ParCNetV2-Tiny is extended to 111×1 and 1×111 with input size 224×224. The capacity of the model will be significantly enhanced with such large convolution kernels. As far as we know, it has achieved the largest convolution kernel among prevailing vision CNNs. Other works on large kernel [43, 12, 36] use a spatially dense form of convolution, which requires massive computation. In contrast, our oversized convolution boosts performance with much less computation cost. It enables our model to achieve state-of-the-art performance, which indicates that it is an effective operation.

Efficiency: The complexity of the oversized convolution is proportional to $HW \times [(2H-1)+(2W-1)]$. Although it has less computation than the previous large kernel convolution networks [12, 36], the multi-fragment structure is poorly supported by the hardware, especially with PyTorch. This is because PyTorch is not optimized for multi-fragmentation, hence we implement a block-wise (inverse) *implicit gemm* algorithm following RepLKNet [12]. Fig 1 shows the comparison results. Compared to other recently proposed models, our ParCNetV2 offers a clear advantage in terms of both accuracy and inference speed. Furthermore, *even on Vanilla PyTorch, our ParCNetV2 achieves a superior trade-off between accuracy and speed*. Additional results can be found in the supplementary material.

Adaptability to multi-scale input: To deal with input im-

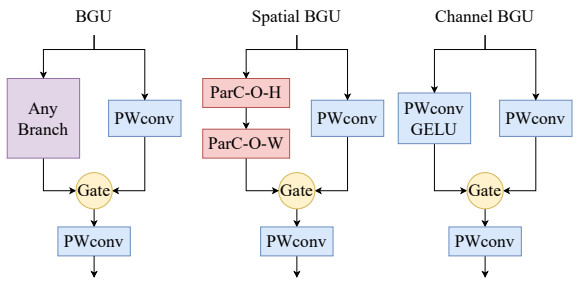


Figure 5. **Illustration of the Bifurcate gate unit (BGU).** We propose a general BGU which can be easily integrated into various network structures. For Spatial GPU, we insert our ParC branch and a point-wise convolution to extract spatial features. While in Channel BGU, we simply adopt a point-wise convolution to conduct channel mixing.

ages of different resolutions, each convolution kernel will be first zoomed with linear interpolation to $C \times (2H-1) \times 1$ and $C \times 1 \times (2W-1)$. In addition, this method keeps the model's global receptive field on any input size and learns to extract scale-invariant features.

3.2. Bifurcate Gate Unit

To make the model data-driven as ViT models, ParCNetV1 employed the squeeze-and-excitation block, which was demonstrated to boost the model performance on various tasks. In this work, the attention mechanism is reinvented with two major improvements: strengthened attention and better computation efficiency. Specifically, we propose the Bifurcate Gate Unit (BGU) structure inspired by gated linear unit (GLU) [10] which improves MLP through gating mechanism. BGU inherits high computation efficiency from GLU and accomplishes attention and feature extraction in a single unit. Different from GLU which inserts gate operation into two homologous features, the proposed BGU applies gate operation on two features from two branches. One branch adopts a point-wise convolution to serve as attention weights. The other transforms the features depending on the purpose of the module, *i.e.*, ParC branch to extract spatial information for spatial interaction, and point-wise convolution to perform channel mixing. Therefore, the BGU design is extended to spatial BGU and channel BGU modules, making it a general module as shown in Fig. 5. Finally, the outputs of the two branches are fused by an element-wise multiplication operation and an additional point-wise convolution. We introduce the details and discuss the difference from other attentions.

Spatial BGU: In the spatial BGU, we aim to extract representative spatial information including local and global dependencies. We adopt ParC branch as the feature transform branch, which consists of a point-wise convolution, a standard local depth-wise convolution and an oversized separable convolution. We will describe it in detail in Sec. 3.3. Basically, our spatial BGU is defined as:

$$
\begin{aligned}
X_1 &= \mathrm{ParC}(X), \\
X_2 &= \mathrm{PWConv}_1(X), \\
\mathrm{SpatialBGU}(X) &= \mathrm{PWConv}_2(X_1 \odot X_2).
\end{aligned}
$$

Channel BGU: For the channel mixing module, the original feed-forward network (FFN) of common transformers usually contains two point-wise convolutions separated by a GELU activation. The first layer expands the number of channels by a factor of α, and the second layer shrinks the dimension back to the original:

$$
\mathrm{FFN}(X) = \mathrm{GELU}(XW_1 + b_1)W_2 + b_2,
$$

where $W_1 \in \mathbf{R}^{C \times \alpha C}$ and $W_2 \in \mathbf{R}^{\alpha C \times C}$ indicate weights of the two point-wise convolutions, b_1 and b_2 are the bias terms, respectively. In our channel BGU, we split the hidden layer into two branches and merge with element-wise multiplication. The whole module is defined as:

$$
\begin{aligned}
X_1 &= \mathrm{GELU}(X\widetilde{W}_1 + \tilde{b}_1), \\
X_2 &= X\widetilde{W}_2 + \tilde{b}_2, \\
\mathrm{ChannelBGU}(X) &= (X_1 \odot X_2)\widetilde{W}_3 + \tilde{b}_3,
\end{aligned}
$$

where $\widetilde{W}_1, \widetilde{W}_2 \in \mathbf{R}^{C \times \tilde{\alpha} C}$ and $\widetilde{W}_3 \in \mathbf{R}^{\tilde{\alpha} C \times C}$ indicates weights of point-wise convolutions, $\tilde{b}_1$, $\tilde{b}_2$, $\tilde{b}_3$ denotes biases, respectively. We adjust $\tilde{\alpha}$ to fit the model size close to the original FFN (details in supplementary).

Comparisons with previous attention mechanisms: The classic channel attentions [27, 54, 40] and spatial attentions [57, 24] consist of two imbalanced branches: a heavy backbone branch and a light attention branch. The attention branch drops massive information by global average pooling, shared attention value across channels or space, and bottleneck structures. However, it contains a large number of parameters similar to the backbone branch. BGU is a compact attention mechanism with more balanced branches. There is no downsampling or bottleneck in each branch. Besides, BGU does not increase the number of parameters of the model.

3.3. Uniform local-global convolution

ParCNetV1 used two different network structures, traditional convolutional block MBConvs [25] in shallow layers and ParC operation in deep layers. We extend the global convolution to each block through the early and late stages, since it is shown that a large receptive field is also critical in the shallow layers, especially in downstream tasks [12, 36]. We design a unified block composed of both local and global convolutions for the entire network. As shown in Fig. 3, we adopt a point-wise convolution first to fuse channel information. Then we pass the feature into two

Models	No. Channels	No. Blocks
ParCNetV2-XT	(48, 96, 192, 320)	(3, 3, 9, 2)
ParCNetV2-T	(64, 128, 320, 512)	(3, 3, 12, 3)
ParCNetV2-S	(64, 128, 320, 512)	(3, 9, 24, 3)
ParCNetV2-B	(96, 192, 384, 576)	(3, 9, 24, 3)

Table 1. **Model configuration of ParCNetV2.** Each tuple represents the number of channels or blocks for the four stages.

branches, one of which is a standard 7x7 depth-wise convolution to extract local cues, and the other is an oversized convolution to model global independence. Finally, we add the two branches to create a multiscale feature. Formally, the uniform local-global convolution is defined as:

$$\begin{aligned} Y_{local} &= \mathrm{DWConv}\left(X\right), \\ Y_{global} &= \mathrm{ParC\text{-}O\text{-}W}\left(\mathrm{ParC\text{-}O\text{-}H}\left(X\right)\right), \\ \mathrm{ParC}(X) &= \mathrm{PWConv}\left(Y_{local} + Y_{global}\right). \end{aligned}$$

3.4. ParCNetV2

Based on the proposed modules above, we build ParCNetV2 with four different scales. We adopt a hierarchical architecture with 4-stage inspired by [37, 38], and the number of channels and blocks of each stage are listed in Tab. 1. ParCNetV2-XT is designed to fairly compare with ParC-ConvNeXt-T ($0.5\times$W), which is a four-stage version of ParCNetV1 [64]. ParCNetV2-T, ParCNetV2-S, and ParCNetV2-B are designed to compare with the state-of-the-art networks. The expand ratio $\tilde{\alpha}$ of channel BGU is set to 2.5, which is close to the original FFN in complexity.

4. Experiments

In this section, we exhibit quantitative and qualitative experiments to demonstrate the effectiveness of ParCNetV2. First of all, we conduct experiments on image classification on the ImageNet-1K [11]. We compare the performance with convolutional neural networks and show that our ParCNetV2 performs better over pure convolutional networks, including ParCNetV1. Then, we compare our model with transformers and hybrid neural networks. Next, we conduct experiments on downstream tasks including object detection and instance segmentation on COCO [35], and semantic segmentation on ADE20K dataset [68]. Finally, we compare the inference latency on GPUs and edge devices.

4.1. Performance Comparison with CNNs

We conduct image classification on ImageNet-1K [11], the most widely used benchmark dataset. We train the ParCNetV2 models on the training set and report top-1 accuracy on the validation set. We follow the same training hyperparameters and augmentations used in ConvNeXt [38] except that the batch size is restricted to 2048 and the initial learning rate is set to 4e-3. We also substitute LayerScale with Resscale [48] to stabilize training.

Models	Param(M)	MACs(G)	Top-1(%)
ParC-Net-S [64]	5.0	3.5	78.6
ParC-ConvNeXt-T($0.5\times$W) [64]	7.4	1.1	78.3
ParCNetV2-XT	7.4	1.6	**79.4**
ResNet50 [22, 56]	23	4.1	79.8
ReGNetY-4G [41, 56]	21	4.0	81.3
ConvNeXt-T [38]	29	4.5	82.1
SLaK-T [36]	30	5.0	82.5
PoolFormer-S24 [61]	21	3.6	80.3
ParCNetV1-27M [64]	27	4.5	82.1
RevCol-T [1]	30	4.5	82.2
ParCNetV2-T	25	4.3	**83.5**
ResNet101 [22, 56]	45	7.9	81.3
ReGNetY-8G [41, 56]	39	8.0	82.1
ConvNeXt-S [38]	50	8.7	83.1
SLaK-S [36]	55	9.8	83.8
RevCol-S [1]	60	9.0	83.5
ParCNetV2-S	39	7.8	**84.3**
ResNet152 [22, 56]	60	11.6	81.8
ReGNetY-16G [41, 56]	84	15.9	82.2
ConvNeXt-B [38]	89	15.4	83.8
RepLKNet-31B [12]	79	15.3	83.5
SLaK-B [36]	95	17.1	84.0
RevCol-B [1]	138	16.6	84.1
ParCNetV2-B	56	12.5	**84.6**

Table 2. **Comparison with the modern convolution networks on image classification.** All experiments are trained on ImageNet-1K dataset with 300 epochs. Top-1 accuracy on the validation set is reported. **ParC-ConvNeXt-T ($0.5\times$W)** [64]: ParCNetV1 of hierarchical 4-stage architecture the same as ParCNetV2. **ParCNetV1-27M**: ParCNetV1 with bigger backbone.

The comparison with pure convolution networks on image classification is listed in Tab. 2. It is clear that ParCNetV2 outperforms other convolutional networks by a large margin across various model scales, including variants of the ResNet (ResNet [22, 56]), NAS architecture (ReGNetY [41]), ConvNeXt [38], and MetaFormer architecture (PoolFormer [61]). Specifically, our ParCNetV2-T surpasses ParCNetV1-27M [64], which indicates that our methods go deeper along the larger convolutions and stronger attention mechanisms. In addition, ParCNetV2-S performs better than the other CNNs even twice larger in parameters and complexity (*e.g.*, ParCNetV2-S and RevCol-B), which indicates that our model is highly effective.

4.2. Performance Comparison with ViTs and Hybrid Models

Apart from CNNs, ParCNetV2 also beats various latest ViTs and Hybrid models. As shown in Tab. 3, compared with famous transformers such as Swin-T [37] and

Models	Mixing Type	Param (M)	MACs (G)	Top-1 (%)
DeIT-S [52]	Attn	22	4.6	79.9
T2T-ViT-14 [62]	Attn	21.5	4.8	81.5
Swin-T [37]	Attn	29	4.5	81.3
CSwin-T [13]	Attn	23	4.3	82.7
CvT-13 [58]	Attn + Conv	20	4.5	81.6
CoAtNet-0 [9]	Attn + Conv	25	4.2	81.6
Uniformer-S [33]	Attn + Conv	20	4.8	82.9
ParCNetV2-T	Conv	25	4.3	**83.5**
T2T-ViT-19 [62]	Attn	39	8.5	81.9
Swin-S [37]	Attn	50	8.7	83.0
CSwin-S [13]	Attn	35	6.9	83.6
CvT-21 [58]	Attn + Conv	32	7.1	82.5
CoAtNet-1 [9]	Attn + Conv	42	8.4	83.3
Uniformer-B [33]	Attn + Conv	50	8.3	83.9
ParCNetV2-S	Conv	39	7.8	**84.3**
DeiT-B/16 [52]	Attn	86	17.6	81.8
T2T-ViT-24 [62]	Attn	64	13.8	82.3
Swin-B [37]	Attn	88	15.4	83.5
CSwin-B [13]	Attn	78	15.0	84.2
CoAtNet-2 [9]	Attn + Conv	75	15.7	84.1
ParCNetV2-B	Conv	56	12.5	**84.6**

Table 3. **Comparison with state-of-the-art transformer and hybrid networks on ImageNet-1K classification dataset.** Top-1 accuracy on the validation set is reported.

CSwin-T [13], ParCNetV2-T improves the accuracy by a clear margin of 2.2% and 0.8% with comparable parameters and computational costs. This result demonstrates that our pure convolution model utilizes the design concepts from transformers in a more efficient way. Compared with hybrid models, ParCNetV2-T outperforms CvT [58], CoAtNet [9], Uniformer [33] and Next-ViT [31] with much fewer parameters. Combined with the above analysis of pure convolutions in Sec. 4.1, our proposed model has achieved better classification accuracy with comparable parameters and computation sizes over various kinds of architectures.

4.3. ParC V2 Performance on Downstream Tasks

To evaluate the transfer ability of ParC V2, we conduct experiments on the object detection and instance segmentation task with COCO [35] semantic segmentation task with ADE20K [68].

Object detection and instance segmentation on COCO. Following previous works [37, 38], we finetune Mask R-CNN and Cascade Mask R-CNN [2] on COCO dataset [35] with ParCNetV2 backbones. MMDetection [4] is used as the base framework. All models use pre-trained weights from ImageNet1K and are trained with $3\times$ schedule with multi-scale training. The experiment settings follow [38]. Tab. 4 shows object detection and instance segmentation results comparing our ParCNetV2 with Swin [37] and ConvNeXt [38]. ParCNetV2 outperforms both the transformer network and convolution network by a large margin across

backbone	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
	Mask R-CNN $3\times$ schedule					
Swin-T [37]	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T [38]	46.2	67.9	50.8	41.7	65.0	44.9
ParCNetV2-T	**48.9**	70.3	53.9	**43.7**	67.6	47.0
	Cascade Mask R-CNN $3\times$ schedule					
Swin-T [37]	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T [38]	50.4	69.1	54.8	43.7	66.5	47.3
ParCNetV2-T	**52.6**	71.0	57.3	**45.6**	68.6	49.8

Table 4. Comparisons on **COCO [35] object detection and instance segmentation.** We use Mask R-CNN and Cascade Mask R-CNN [2] as a basic framework. All models are pretrained on ImageNet-1K and trained on COCO for $3\times$ iterations.

backbone	Param(M)	MACs(G)	mIoU(%)
Swin-T [37]	60	945	45.8
ConvNeXt-T [38]	60	939	46.7
ParCNetV1-27M [64]	56	936	46.7
ParCNetV2-T	55	932	**49.4**
Swin-S [37]	81	1038	49.5
ConvNeXt-S [38]	82	1027	49.6
ParCNetV2-S	69	1005	**51.0**

Table 5. Comparisons on **ADE20K [68] semantic segmentation.** We use UperNet as a basic framework. All models are pretrained on ImageNet-1K and trained on ADE20K for 160K iterations. MACs are measured with the input size of (2048, 512).

different model complexities. Interestingly, in experiments using Cascade Mask R-CNN, ParCNetV2-T has already outperformed larger models such as Swin-S and ConvNeXt-S, achieving 51.9 AP^{bbox} and 45.0 AP^{mask}, which is a significant improvement of +0.7 AP^{bbox} and +0.6 AP^{mask}, respectively. For further information on experiments with backbones of different scales, please refer to the supplementary materials.

Semantic segmentation on ADE20K. We finetune UperNet [59] on the ADE20K [68] dataset with ParCNetV2 backbones. MMSegmentation [8] is used as the base framework. All models use pre-trained weights from ImageNet1K and are trained for 160K iterations with a batch size of 16. Experiment settings follow [38]. Tab. 5 lists the mIoU, model size, and MACs for different backbones. ParCNetV2 achieves a substantially higher mIoU than Swin and ConvNeXt, while taking fewer parameters and computation. Specifically, our model is +2.7% mIoU higher than ParCNetV1-27M [64], which validates the transferability of our ParCNetV2 model.

4.4. Ablation Study

In this section, we make an ablation study on ImageNet-1K classification to show that each component in our ParCNetV2 is critical. To speed up the experiment, we use the smaller ParCNetV2-XT in this section. Training settings are

Row	OC	S-BGU	C-BGU	Uniform	Param (M)	MACs (G)	Top-1 (%)
baseline	✓	✓	✓	✓	7.4	1.6	**79.4**
1		✓	✓	✓	7.2	1.4	78.9
2	✓		✓	✓	7.4	1.6	79.2
3	✓	✓		✓	7.4	1.5	79.1
4	✓	✓	✓		7.4	1.4	79.2

Table 6. **Ablation study of each component on the ImageNet-1K classification task.** We use smaller ParCNetV2-XT in ablation for fast evaluation. Top-1 accuracy on the validation set is reported. **OC**: Oversized Convolution. **S-BGU**: Spatial Bifurcate Gate Unit. **C-BGU**: Channel Bifurcate Gate Unit. **Uniform**: Uniform local-global convolution.

the same as image classification experiments in Sec. 4.2.

Oversized convolution. Oversized convolution increases the capacity of the model and encodes position information. Without oversized convolution, the model not only loses capacity and position information, but also loses the ability to learn long-range dependencies. By comparing baseline and Row 1, the accuracy of the model without oversized convolution drops substantially by 0.6% (79.4% *v.s.* 78.9%) top-1 accuracy. It demonstrates that long-range dependencies are important to networks.

Bifurcate gate units. The bifurcate gate unit is an important mechanism to introduce data-driven operations into ParCNetV2. It increases the non-linearity and enhances the fitting ability. There is a degradation of 0.2% (79.4% *v.s.* 79.2%) without spatial BGU, and 0.3%(79.4% *v.s.* 79.1%) without channel BGU as shown in baseline, Row 2 and Row 3. It is similar to the data-driven operation of the squeeze-and-excitation block in ParC V1, while our BGU differs in the following two points. First BGU does not increase parameters. With $\tilde{\alpha} = 2.5$, our channel BGU is slightly more lightweight than the original FFN. Second, the two branches in our BGU are more balanced. They share a similar number of parameters and computational costs, unlike the heavy main branch and lightweight channel attention in most methods.

Uniform local-global convolution. The objective of the uniform local-global convolution block is to standardize the blocks used across various stages. In ParCNetv1, MobileNetV2 blocks had to be mixed with ParC blocks to construct the entire network. However, in ParCNet V2, the entire network is built by stacking ParCNet V2 blocks, as illustrated in Figure 1 in the supplementary material. This uniform design offers greater flexibility and ease of combination with other structures. Additionally, the uniform design results in a performance gain of 0.2%.

4.5. Latency analysis

We analyze the inference latency of our ParCNetV2 on RTX3090 GPU and edge device RK3288. The Rockchip RK3288 is widely used in real-world applications such as smart TV and AI entrance guard system.

Models	Param(M) (M)	MACs (G)	Latency↓ (ms)	Memory↓ (MB)	Top-1↑ (%)
Swin-T	29	4.5	855	139	81.3
ConvNeXt-T	29	4.5	875	129	82.1
ParCNetV2-T	25	4.3	**840**	**118**	**83.5**
Swin-S	50	8.7	1576	222	83.0
ConvNeXt-S	50	8.7	1618	211	83.1
ParCNetV2-S	39	7.8	**1485**	**181**	**84.3**
Swin-B	88	15.4	2649	378	83.5
ConvNeXt-B	89	15.4	2708	364	83.8
ParCNetV2-B	56	12.5	**2339**	**252**	**84.6**

Table 7. **Inference on Arm (Quad Core Cortex-A17).** We compare the latency and memory cost during inference together with ImageNet-1K top-1 accuracy. Results are measured using RK3288 with batch size 1 and averaged over 100 iterations.

GPU inference latency. To ensure a fair comparison with large kernel convolution networks which use the *implicit gemm* acceleration algorithm, such as RepLKNet [12] and SLaK [36], we measure the inference latency of our ParCNetV2 models using a single NVIDIA RTX 3090 GPU with a batch size of 32, following the consistent implementation as theirs. As illustrated in Fig. 1, ParCNetV2 models achieve superior latency-accuracy trade-offs among large kernel networks, outperforming both Swin and ConvNeXt.

Arm inference latency. On RK3288, we port the models to the chip through ONNX and MNN and conducted each test for 100 iterations to measure the average inference speed. Tab. 7 demonstrates that ParCNetV2 runs faster and performs substantially better than Swin and ConvNeXt. Moreover, our model requires less memory, making it a more suitable option for edge computing applications.

5. Conclusion

This paper presents ParCNetV2, a pure convolutional neural network with state-of-the-art performance. It extends position-aware circular convolution with oversized convolutions and strengthens attention through bifurcate gate units. Besides, it utilizes a uniform local-global convolution block to unify the design of the early and late-stage convolution blocks. We conduct extensive experiments on image classification and semantic segmentation to show the effectiveness and superiority of the proposed ParCNetV2 architecture.

Acknowledgement

This work is supported in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011, in part by The National Key Research and Development Program of China under Grant No. 2018YFE0118400.

References

[1] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. *arXiv preprint arXiv:2212.11696*, 2022.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

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

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

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

[6] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594, 2018.

[7] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022.

[8] MMSegmentation Contributors. Mmsegmentation, an open source semantic segmentation toolbox, 2020.

[9] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[10] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

[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.

[12] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11963–11975, 2022.

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

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021.

[16] Zhanzhou Feng and Shiliang Zhang. Efficient vision transformer via token merger. *IEEE Transactions on Image Processing*, 2023.

[17] Zhanzhou Feng and Shiliang Zhang. Evolved part masking for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10386–10395, 2023.

[18] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021.

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

[20] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3828–3837, 2019.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[22] 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.

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

[24] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13713–13722, 2021.

[25] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

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

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

[28] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[29] Aisha Urooj Khan, Amir Mazaheri, Niels Da Vitoria Lobo, and Mubarak Shah. Mmft-bert: Multimodal fusion transformer with bert encodings for visual question answering. *arXiv preprint arXiv:2010.14095*, 2020.

[30] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[31] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022.

[32] Jianing Li, Shiliang Zhang, and Tiejun Huang. Multi-scale 3d convolution network for video based person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8618–8625, 2019.

[33] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

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

[35] 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.

[36] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022.

[37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[39] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[40] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 783–792, 2021.

[41] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[43] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in neural information processing systems*, 34:980–993, 2021.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[45] Jared Roesch, Steven Lyubomirsky, Marisa Kirisame, Logan Weber, Josh Pollock, Luis Vega, Ziheng Jiang, Tianqi Chen, Thierry Moreau, and Zachary Tatlock. Relay: A high-level compiler for deep learning. *arXiv preprint arXiv:1904.08368*, 2019.

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

[47] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[48] Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021.

[49] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021.

[50] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through at-

tention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[54] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020.

[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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[56] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

[57] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[58] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[59] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[60] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.

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

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

[63] Haokui Zhang, Wenze Hu, and Xiaoyu Wang. Cabvit: Cross attention among blocks for vision transformer. *arXiv preprint arXiv:2211.07198*, 2022.

[64] Haokui Zhang, Wenze Hu, and Xiaoyu Wang. Parc-net: Position aware circular convolution with merits from convnets and transformer. *networks (ConvNets)*, 5(33):21, 2022.

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

[66] Lianmin Zheng, Chengfan Jia, Minmin Sun, Zhao Wu, Cody Hao Yu, Ameer Haj-Ali, Yida Wang, Jun Yang, Danyang Zhuo, Koushik Sen, et al. Ansor: Generating {High-Performance} tensor programs for deep learning. In *14th USENIX symposium on operating systems design and implementation (OSDI 20)*, pages 863–879, 2020.

[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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[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 of Computer Vision*, 127(3):302–321, 2019.