Focal Modulation Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

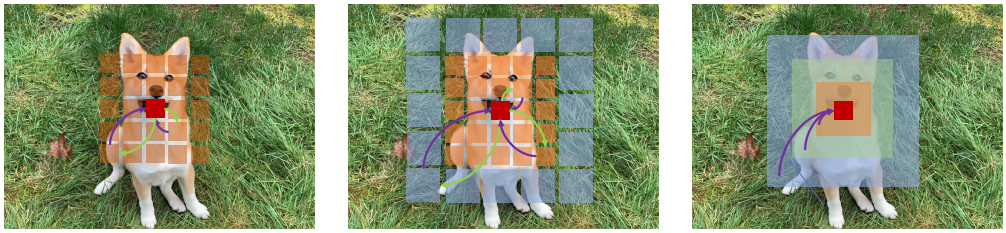
In this work, we propose *focal modulation network* (*FocalNet* in short), where self-attention (SA) is completely replaced by a *focal modulation* module for modeling token interactions. Focal modulation comprises three components: (i) hierarchical contextualization, implemented using a stack of depth-wise convolutional layers, to encode visual contexts from short to long ranges, (ii) gated aggregation to selectively aggregate context features for each token (query) based on its content, and (iii) element-wise modulation or affine transformation to fuse the aggregated context into the query. Extensive experiments show FocalNets outperform the state-of-the-art SA counterparts (*e.g.*, Swin Transformers) with similar computational cost on the tasks of image classification, object detection, and semantic segmentation. Specifically, FocalNets with tiny and base size achieve **82.3**% and **83.9**% top-1 accuracy on ImageNet-1K. After pretrained on ImageNet-22K, it attains **86.5**% and **87.3**% top-1 accuracy when finetuned with resolution 224^2 and 384^2, respectively. When transferred to downstream tasks, FocalNets exhibit remarkable superiority. For object detection with Mask R-CNN, FocalNet base trained with $1\times$ outperforms the Swin counterpart by **2.1** points and even surpasses Swin trained with $3\times$ schedule (**49.0** *v.s.* 48.5). For semantic segmentation with UperNet, FocalNet base at single-scale outperforms Swin by **2.4**, and also beats Swin at multi-scale (**50.5** *v.s.* 49.7). These results render focal modulation a favorable alternative to SA for effective and efficient visual modeling.

1 Introduction

Transformers [52], originally proposed for natural language processing (NLP), have become a prevalent architecture in computer vision since the seminal work of Vision Transformer (ViT) [14]. Its promise has been demonstrated in various vision tasks including image classification [49, 55, 59, 36, 68, 51], object detection [3, 76, 73, 11], segmentation [53, 57, 9], and beyond [30, 71, 4, 7, 54, 28]. In Transformers, the self-attention (SA) is arguably the key to its success which enables input-dependent global interactions, in contrast to convolution operation which constrains interactions in a local region with a shared kernel. Despite this advantages, the efficiency of SA has been a concern due to its quadratic complexity over the number of visual tokens, especially for high-resolution inputs. To address this, many works have proposed SA variants by token coarsening [55], window attention [36, 51, 68], or the combination [63, 10]. Meanwhile, a number of hybrid models have been proposed by augmenting SA with (depth-wise) convolution to capture long-range dependencies with a good awareness of local structures [59, 16, 62, 15, 13].

In this work, we aim to answer the fundamental question: *Is there a more efficient and effective way than (hybrid) SA to model input-dependent long-range interactions?* We start with an analysis of the current SoTA methods. In Fig. 1(a), we show a window-wise attention between the red query token and the surrounding orange tokens proposed in Swin Transformer [36]. With a simple window-shift strategy, Swin attains superior performance to ResNets across various vision tasks. To enlarge the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Window-wise SA (b) Focal Attention (c) Focal Modulation

Figure 1: Illustrative comparison among (a) Window-wise Self-Attention (SA) [36], (b) Focal Attention (FA) [63] and (c) the proposed Focal Modulation. Given the query token ■, window-wise SA captures spatial context from its surrounding tokens ■, FA, in addition, uses far-away summarized tokens ■, and Focal Modulation first encodes spatial context at different levels of granularity into summarized tokens (■ , ■ , ■), which are then selectively fused into the query token based on the query content. Green and purple arrows represent the attention interactions and query-dependent aggregations, respectively (we do not draw all arrows for clarity). Both local self-attention and focal attention involve heavy interaction and aggregation operations, while our focal modulation turn both of them light-weight. Figures better viewed in color.

receptive field, focal attention [63] is proposed to additionally aggregate summarized visual tokens far away to capture coarse-grained, long-range visual dependencies, as shown in Fig. 1(b). To generate the output, both methods involve a heavy interaction (green arrows) followed by an equally heavy aggregation (purple arrows) between the query and a large number of spatially distributed tokens (context features), which are extracted via either window partition or unfolding. In this work, we take an alternative way by first aggregating contexts around each query and then modulating the query with the aggregated context. This alteration still enables input-dependent token interaction, but significantly eases the process by decoupling the aggregation with individual queries and making the interaction light-weight upon a couple of features. As shown in Fig. 1(c), we can simply apply query-agnostic aggregations (*e.g.*, depth-wise convolution) to generate summarized tokens at different levels of granularity. Afterwards, these summarized contexts are selectively aggregated depending on the query content, and finally fused into the query vector. We call this new method *focal modulation* and replace SA with it for input-dependent token interaction, resulting in a simpler and attention-free architecture, called *Focal Modulation Network* (or *FocalNet* in short).

Extensive experiments on image classification, object detection and segmentation, show that our FocalNets consistently and significantly outperform the SoTA SA counterparts with comparable costs. Notably, our FocalNet achieves **82.3**% and **83.9**% top-1 accuracy using tiny and base model size, but with comparable and doubled throughput than Swin and Focal Transformer, respectively. When pretrained on ImageNet-22K, our FocalNets achieve **86.5**% and **87.3**% in 224^2 and 384^2 resolution, respectively, which are comparable or better than Swin at similar cost. The advantage is particularly significant when transferred to dense prediction tasks. For object detection on COCO [33], our FocalNets with tiny and base model size achieve **46.1** and **49.0** box mAP on Mask R-CNN $1\times$, surpassing Swin with $3\times$ schedule (46.0 and 48.5 box mAP). For semantic segmentation on ADE20k [74], our FocalNet with base model size achieves **50.5** mIoU at single-scale evaluation, outperforming Swin at multi-scale evaluation (49.7 mIoU). Finally, we apply our focal modulation to monolithic ViT and also demonstrate superior performance across different model sizes.

2 Related Work

Self-attentions. Self-attention (SA) [52] is first introduced in Vision Transformer (ViT) [14] by splitting an image into a sequence of visual tokens. This simple strategy has demonstrated superior performance to modern convolutional neural networks (ConvNets) such as ResNet [21] when trained with optimized recipes [14, 49]. Afterwards, multi-scale architectures [5, 55, 62], light-weight convolution layers [59, 16, 31], local self-attention mechanisms [36, 68, 10] and learnable attention weights [67] have been proposed to boost the performance and support high-resolution input. More comprehensive surveys are covered in [27, 18, 27]. Our focal modulation significantly differs from SA by first aggregating the contexts from different levels of granularity and then modulating individual query tokens, rendering an attention-free model architecture. For context aggregation, our method is inspired by focal attention proposed in [63]. However, the context aggregation for focal modulation is performed at each query location instead of target location, followed by a modulation rather than an attention. These differences in mechanism lead to significant improvement of efficiency

and performance as well. Another closely related work is Poolformer [66] which uses a pooling to summarize the local context and a simple subtraction to adjust the individual inputs. Though achieving decent efficiency, Poolformer lags behind popular vision transformers like Swin on performance. As we will show later, capturing local structures at different levels is essential for superior performance.

MLP architectures. Visual MLPs can be categorized into two groups: (i) Global-mixing MLPs, such as MLP-Mixer [46] and ResMLP [48], perform global communication among visual tokens through spatial-wise projections augmented by various techniques, such as gating, routing, and Fourier transforms [35, 39, 44, 45]. (ii) Local-mixing MLPs sample nearby tokens for interactions, using spatial shifting, permutation, and pseudo-kernel mixing [65, 23, 32, 6, 17]. Recently, Mix-Shift-MLP [72] exploits both local and global interactions with MLPs, in a similar spirit of focal attention [63]. Both MLP architectures and our focal modulation network are attention-free. However, focal modulation with multi-level context aggregation naturally captures the structures in both short- and long-range, and thus achieves a better accuracy-efficiency trade-off.

Convolutions. ConvNets have been the primary driver of the renaissance of deep neural networks in computer vision. The field has evolved rapidly since the emerge of VGG [40], InceptionNet [42] and ResNet [21]. Representative works that focus on the efficiency of ConvNets are MobileNet [24], ShuffleNet [70] and EfficientNet [43]. Another line of works aimed at integrating global context to compensate ConvNets such as SE-Net [26], Non-local Network [56], GCNet [2], LR-Net [25] and C3Net [64], *etc*. Introducing dynamic operation is another way to augment ConvNets as demonstrated in Involution [29] and DyConv [8]. Recently, ConvNets strike back from two aspects: (i) convolution layers are integrated to SA and bring significant gains [59, 16, 31, 15] or the vice versa [50]; (ii) ResNets have closed the gap to ViTs using similar data augmentation and regularization strategies [58], and replacing SA with (dynamic) depth-wise convolution [19, 37] can surpass Swin. Our focal modulation network also exploits depth-wise convolution as the micro-architecture but goes beyond by introducing a multi-level context aggregation and input-dependent modulation. We will show this new module significantly outperforms raw depth-wise convolution.

3 Focal Modulation Network

3.1 From Self-Attention to Focal Modulation

Given a visual feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ as input, a generic visual modeling generates for each visual token (query) $\boldsymbol{x}_i \in \mathbb{R}^C$ a feature representation $\boldsymbol{y}_i \in \mathbb{R}^C$ via the interaction $\mathcal{T}$ with its surroundings $\mathbf{X}$ (*e.g.,* neighboring tokens) and aggregation $\mathcal{M}$ over the contexts. The self-attention modules use a late aggregation procedure formulated as

$$\boldsymbol{y}_i = \mathcal{M}_1(\mathcal{T}_1(\boldsymbol{x}_i, \mathbf{X}), \mathbf{X}), \tag{1}$$

where the aggregation $\mathcal{M}_1$ over the contexts $\mathbf{X}$ is performed after the query-target attention scores are computed via interaction $\mathcal{T}_1$. In contrast, we propose focal modulation to generate refined representation $\boldsymbol{y}_i$ using an early aggregation procedure formulated as

$$\boldsymbol{y}_i = \mathcal{T}_2(\mathcal{M}_2(\boldsymbol{x}_i, \mathbf{X}), \boldsymbol{x}_i), \tag{2}$$

where the context features are aggregated using $\mathcal{M}_2$ first, then the query interacts with the aggregated feature using $\mathcal{T}_2$ to fuse the contexts to form $\boldsymbol{y}_i$. Comparing (2) with (1), we see that (i) the context aggregation of focal modulation $\mathcal{M}_2$ amortizes the computation of contexts via a shared operator (*e.g.*, depth-wise convolution), while $\mathcal{M}_1$ in SA is more computationally expensive as it requires summing over non-shareable attention scores for different queries; (ii) the interaction $\mathcal{T}_2$ is a lightweight operator between a token and its context, while $\mathcal{T}_1$ involves computing token-to-token attention scores, which has quadratic complexity. Fig. 2(a) and (b) show SA and focal modulation, respectively.

Specifically, in this study we implement focal modulation of (2) as

$$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot \mathcal{M}_2(\boldsymbol{x}_i, \mathbf{X}), \tag{3}$$

where $q(\cdot)$ is a query projection function, $\odot$ is the element-wise multiplication operator. That is, the interaction operator $\mathcal{T}_2$ is implemented using a simple $q(\cdot)$ and $\odot$. The proposed focal modulation has the following favorable properties:

- **Translation invariance**. Since $q(\cdot)$ and $\mathcal{M}_2(\cdot)$ are always centered at the target visual token and no positional embedding is used, the modulation is invariant to translation of input feature map $\mathbf{X}$.

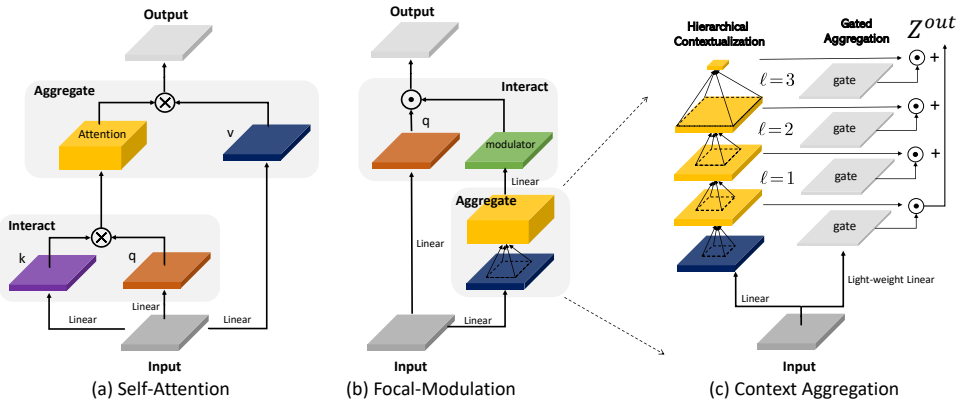


Figure 2: Left: Comparing SA (a) and focal modulation (b) side by side. Right: Detailed illustration of context aggregation in focal modulation (c).

- **Explicit input-dependency**. Instead of a set of learnable parameters, the modulator is computed via $\mathcal{M}_2$ by aggregating the local features around target location i, hence our focal modulation is explicitly input-dependent.
- **Spatial- and channel-specific**. The target location i as a pointer for $\mathcal{M}_2$ enables spatial-specific modulation. The element-wise multiplication enables channel-specific modulation.
- **Decoupled feature granularity**. $q(\cdot)$ preserve the finest information for individual tokens, while $\mathcal{M}_2$ extracts the coarser context. They are decoupled but combined through modulation.

In what follows, we describe in detail the implementation of $\mathcal{M}_2$ in Eq. (3).

3.2 Context Aggregation via $\mathcal{M}_2$

It has been proved that both short- and long-range contexts are important for visual modeling [63, 13, 37]. However, a single aggregation with larger receptive field is not only computationally expensive in time and memory, but also undermines the local fine-grained structures which are particularly useful for dense prediction tasks. Inspired by [63], we propose to implement $\mathcal{M}_2$ through a multi-scale hierarchical context aggregation. As depicted in Fig. 2 (c), the aggregation procedure consists of two steps: *hierarchical contextualization* to extract contexts from local to global ranges at different levels of granularity and *gated aggregation* to condense all context features at different granularity levels into a single feature vector, namely *modulator*.

Step 1: Hierarchical Contextualization. Given input feature map $\mathbf{X}$, we first project it into a new feature space with a linear layer $\mathbf{Z}^0 = f_z(\mathbf{X}) \in \mathbb{R}^{H\times W\times C}$. Then, a hierarchical presentation of contexts is obtained using a stack of L depth-wise convolutions. At focal level $\ell \in \{1, ..., L\}$, the output $\mathbf{Z}^\ell$ is derived by:

$$\mathbf{Z}^\ell = f_a^\ell(\mathbf{Z}^{\ell-1}) \triangleq \mathsf{GeLU}(\mathsf{Conv}_{dw}(\mathbf{Z}^{\ell-1})), \tag{4}$$

where f_a^ℓ is the contextualization function at the ℓ-th level, implemented via a depth-wise convolution Conv_{dw} with kernel size k^ℓ followed by a GeLU activation function [22]. The use of depth-wise convolution for hierarchical contextualization of Eq. (4) is motivated by its desirable properties. Compared to pooling [66, 26], depth-wise convolution is learnable and structure-aware. In contrast to regular convolution, it is channel-wise and thus computationally much cheaper.

Hierarchical contextualization of Eq. (4) generates L levels of feature maps. At level ℓ, the effective receptive field is $r^\ell = 1 + \sum_{i=1}^{\ell}(k^\ell - 1)$, which is much larger than the kernel size k^ℓ. To capture global context of the whole input, which could be high-resolution, we apply a global average pooling on the L-th level feature map $\mathbf{Z}^{L+1} = \mathsf{Avg\text{-}Pool}(\mathbf{Z}^L)$. Thus, we obtain in total $(L+1)$ feature maps $\{\mathbf{Z}^\ell\}_{\ell=1}^{L+1}$, which collectively capture short- and long-range contexts at different levels of granularity.

Step 2: Gated Aggregation. In this step, the $(L+1)$ feature maps obtained via hierarchical contextualization are condensed into a *modulator*, *i.e.*, a single feature vector. In an image, the relation between a visual token (query) and its surrounding contexts often depends on the content itself. For example, we might heavily rely on local fine-grained features for encoding the queries of

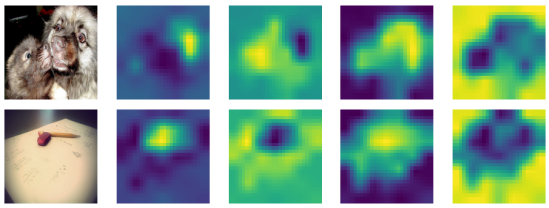


Figure 3: Visualization of gating values $\mathbf{G}$ in Eq. (5) at last layer of our FocalNet ($L = 3$) pretrained on ImageNet-1K. The columns from left to right are input images, gating maps at focal level 1,2,3 and global level.

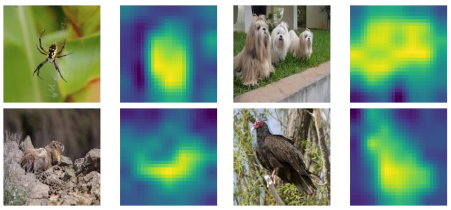


Figure 4: Visualization of modulator values (corresponding to the right side of $\odot$ in Eq. (6)) at the last layer in FocalNet. The original modulator map is upsampled for display.

salient visual objects, but mainly global coarse-grained features for the queries of background scenes. Based on this intuition, we use a gating mechanism to control how much to aggregate from different levels for each query. Specifically, we use a linear layer to obtain a spatial- and level-aware gating weights $\mathbf{G} = f_g(\mathbf{X}) \in \mathbb{R}^{H\times W\times (L+1)}$. Then, we perform a weighted sum through an element-wise multiplication to obtain a single feature map $\mathbf{Z}^{out}$ which has the same size as the input $\mathbf{X}$,

$$\mathbf{Z}^{out} = \sum_{\ell=1}^{L+1} \mathbf{G}^{\ell} \odot \mathbf{Z}^{\ell} \tag{5}$$

where $\mathbf{G}^{\ell} \in \mathbb{R}^{H\times W\times 1}$ is a slice of $\mathbf{G}$ for the level ℓ. When visualizing these gating maps in Fig. 3, we surprisingly find our FocalNet indeed learns gathering the context from different focal levels adaptively as we expect. As we can see, for a token on a small object, it focuses more on the fine-grained local structure, while a token in a uniform background needs to be aware of much larger contexts. Until now, all the aggregation is spatial. To model the communication across different channels, we use another linear layer $h(.)$ to obtain the modulator $\mathbf{M} = h(\mathbf{Z}^{out}) \in \mathbb{R}^{H\times W\times C}$.

Focal Modulation. Given the implementation of $\mathcal{M}_2$ as described above, focal modulation of Eq.(3) can be rewritten at the token level as

$$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot h(\sum_{\ell=1}^{L+1} \boldsymbol{g}_i^{\ell} \cdot \boldsymbol{z}_i^{\ell}) \tag{6}$$

where $\boldsymbol{g}_i^{\ell}$ and $\boldsymbol{z}_i^{\ell}$ are the gating value and visual feature at location i of $\mathbf{G}^{\ell}$ and $\mathbf{Z}^{\ell}$, respectively. In Fig. 4, we visualize the magnitude of modulator $\mathbf{M}$ at the last layer of our FocalNet. Interestingly, the modulators automatically pay more attention to the foregrounds regions inducing the image category, which implies a novel way of interpreting our FocalNets.

3.3 Complexity

In focal modulation as Eq. (6), there are mainly three linear projections $q(\cdot)$, $h(\cdot)$, and $f_z(\cdot)$ for $\mathbf{Z}^0$. Besides, it requires a lightweight linear function $f_g(\cdot)$ for gating and L depth-wise convolution $f_a^{\{1,\ldots,L\}}$ for hierarchical contextualization. Therefore, the overall number of learnable parameters is $3C^2 + C(L+1) + C\sum_{\ell}(k^{\ell})^2$. Since L and $(k^{\ell})^2$ are typically much smaller than C, the model size is mainly determined by the first term as we will show in Sec. 4. Regarding the time complexity, besides the linear projections and the depth-wise convolution layers, the element-wise multiplications introduce $\mathcal{O}(C(L+2))$ for each visual token. Hence, the total complexity for a feature map is $\mathcal{O}(HW \times (3C^2 + C(2L+3) + C\sum_{\ell}(k^{\ell})^2))$. For comparison, a window-wise attention in Swin Transformer with window size w is $\mathcal{O}(HW \times (3C^2 + 2Cw^2))$.

3.4 Network Architectures

For fair comparisons, we use the same stage layouts and hidden dimensions as in SoTA methods Swin [36] and Focal Transformers [63], but replace the SA modules with the focal modulation modules. We thus construct a series of Focal Modulation Network (FocalNet) variants. In FocalNets, we only need to specify the number of focal levels (L) and the kernel size (k^{ℓ}) at each level. For simplicity, we gradually increase the kernel size by 2 from lower focal levels to higher ones, *i.e.*, $k^{\ell} = k^{\ell-1} + 2$. To match the complexities of Swin and Focal Transformers, we design a small receptive field (SRF) and a large receptive field (LRF) version for each of the four layouts by using 2 and 3 focal levels, respectively. We use non-overlapping convolution layers for patch embedding at the beginning (kernel size=4×4, stride=4) and between two stages (kernel size=2×2, stride=2), respectively. The detailed configurations of our FocalNet variants are summarized in Appendix.

Model	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 (%)
ResNet-50 [21]	25.0	4.1	1294	76.2
ResNet-101 [21]	45.0	7.9	745	77.4
ResNet-152 [21]	60.0	11.0	522	78.3
ResNet-50-SB [58]	25.0	4.1	1294	79.8
ResNet-101-SB [58]	45.0	7.9	745	81.3
ResNet-152-SB [58]	60.0	11.6	522	81.8
DW-Net-T [19]	24.2	3.8	1030	81.2
DW-Net-B [19]	74.3	12.9	370	83.2
Mixer-B/16 [47]	59.9	12.7	455	76.4
gMLP-S [34]	19.5	4.5	785	79.6
gMLP-B [34]	73.4	15.8	301	81.6
ResMLP-S24 [48]	30.0	6.0	871	79.4
ResMLP-B24 [48]	129.1	23.0	61	81.0
DeiT-Small/16 [49]	22.1	4.6	939	79.9
DeiT-Base/16 [49]	86.6	17.5	291	81.8
PVT-Small [55]	24.5	3.8	794	79.8
PVT-Medium [55]	44.2	6.7	517	81.2
PVT-Large [55]	61.4	9.8	352	81.7
PoolFormer-m36 [66]	56.2	8.8	463	82.1
PoolFormer-m48 [66]	73.5	11.6	347	82.5
Swin-Tiny [36]	28.3	4.5	760	81.2
FocalNet-T (SRF)	28.4	4.4	743	**82.1**
Swin-Small [36]	49.6	8.7	435	83.1
FocalNet-S (SRF)	49.9	8.6	434	**83.4**
Swin-Base [36]	87.8	15.4	291	83.5
FocalNet-B (SRF)	88.1	15.3	280	**83.7**
FocalAtt-Tiny [63]	28.9	4.9	319	82.2
FocalNet-T (LRF)	28.6	4.5	696	**82.3**
FocalAtt-Small	51.1	9.4	192	**83.5**
FocalNet-S (LRF)	50.3	8.7	406	**83.5**
FocalAtt-Base [63]	89.8	16.4	138	83.8
FocalNet-B (LRF)	88.7	15.4	269	**83.9**

Table 1: ImageNet-1K classification comparison.

Model	Overlapped PatchEmbed	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 (%)
FocalNet-T (SRF)		28.4	4.4	743	82.1
FocalNet-T (SRF)	✓	30.4	4.4	730	**82.4**
FocalNet-S (SRF)		49.9	8.6	434	83.4
FocalNet-S (SRF)	✓	51.8	8.6	424	**83.4**
FocalNet-B (SRF)		88.1	15.3	286	83.7
FocalNet-B (SRF)	✓	91.6	15.3	278	**84.0**

Table 2: Effect of overlapped patch embedding.

Model	Depth	Dim.	#Params.	FLOPs	Throughput	Top-1
FocalNet-T (SRF)	2-2-6-2	96	28.4	4.4	743	82.1
FocalNet-T (SRF)	3-3-16-3	64	25.1	4.0	663	**82.7**
FocalNet-S (SRF)	2-2-18-2	96	49.9	8.6	434	83.4
FocalNet-S (SRF)	4-4-28-4	64	38.2	6.4	440	**83.5**
FocalNet-B (SRF)	2-2-18-2	128	88.1	15.3	280	83.7
FocalNet-B (SRF)	4-4-28-4	96	85.1	14.3	247	**84.1**

Table 3: Effect of deeper and thinner networks.

Model	Img. Size	#Params	FLOPs	Throughput	Top-1
ResNet-101x3 [21]	384^2	388.0	204.6	-	84.4
ResNet-152x4 [21]	480^2	937.0	840.5	-	85.4
ViT-B/16 [14]	384^2	86.0	55.4	99	84.0
ViT-L/16 [14]	384^2	307.0	190.7	30	85.2
Swin-Base [36]	$224^2/224^2$	88.0	15.4	291	85.2
FocalNet-B	$224^2/224^2$	88.1	15.3	280	**85.6**
Swin-Base [36]	$384^2/384^2$	88.0	47.1	91	86.4
FocalNet-B	$224^2/384^2$	88.1	44.8	94	**86.5**
Swin-Large [36]	$224^2/224^2$	196.5	34.5	155	86.3
FocalNet-L	$224^2/224^2$	197.1	34.2	144	**86.5**
Swin-Large [36]	$384^2/384^2$	196.5	104.0	49	**87.3**
FocalNet-L	$224^2/384^2$	197.1	100.6	50	**87.3**

Table 4: ImageNet-1K finetuning results with models pretrained on ImageNet-22K. Numbers before and after "/" are resolutions used for pretraining and finetuning, respectively. To adapt to higher resolution, we use three focal levels.

4 Experiment

4.1 Image Classification

We compare different methods on ImageNet-1K classification [12]. Following the recipes in [49, 36, 63], we train FocalNet-T, FocalNet-S and FocalNet-B with ImageNet-1K training set and report Top-1 accuracy (%) on the validation set. Training details are described in the appendix.

To verify the effectiveness of FocalNet, we compare it with three groups of methods based on ConvNets, Transformers and MLPs. The results are reported in Table 1. We see that FocalNets outperform the conventional CNNs (*e.g.*, ResNet [21] and the augmented version [58]), MLP architectures such as MLP-Mixer [47] and gMLP [34], and Transformer architectures DeiT [49] and PVT [55]. In particular, we compare FocalNets against Swin and Focal Transformers which use the same architecture to verify FocalNet's stand-alone effectiveness at the bottom part. We see that FocalNets with small receptive fields (SRF) achieve consistently better performance than Swin Transformer but with similar model size, FLOPs and throughput. For example, the tiny FocalNet improves Top-1 accuracy by 0.9% over Swin-Tiny. To compare with Focal Transformers (FocalAtt), we change to large receptive fields (LRF) though it is still much smaller than the one used in FocalAtt. Focal modulation outperforms the strong and sophisticatedly designed focal attention across all model sizes. More importantly, its run-time speed is much higher than FocalAtt by getting rid of many time-consuming operations like rolling and unfolding.

Model augmentation. We investigate whether some commonly used techniques for vision transformers can also improve our FocalNets. First, we study the effect of using overlapped patch embedding for downsampling [16]. Following [59], we change the kernel size and stride from $(4, 4)$ to $(7, 4)$ for patch embedding at the beginning, and $(2, 2)$ to $(3, 2)$ for later stages. The comparisons are reported in Table 2. Overlapped patch embedding improves the performance for models of all sizes, with slightly increased computational complexity and time cost. Second, we make our FocalNets deeper but thinner as in [13, 75]. In Table 3, we change the depth layout of our FocalNet-T from 2-2-6-2 to 3-3-16-3, and FocalNet-S/B from 2-2-18-2 to 4-4-28-4. Meanwhile, the hidden dimension at first stage is reduced from 96, 128 to 64, 96, respectively. These changes lead to smaller model sizes and fewer FLOPs, but higher time cost due to the increased number of sequential blocks. It turns out that

Backbone	#Params	FLOPs	Mask R-CNN 1x						Mask R-CNN 3x					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [21]	44.2	260	38.0	58.6	41.4	34.4	55.1	36.7	41.0	61.7	44.9	37.1	58.4	40.1
PVT-Small[55]	44.1	245	40.4	62.9	43.8	37.8	60.1	40.3	43.0	65.3	46.9	39.9	62.5	42.8
Twins-SVT-S [10]	44.0	228	43.4	66.0	47.3	40.3	63.2	43.4	46.8	69.2	51.2	42.6	66.3	45.8
Swin-Tiny [36]	47.8	264	43.7	66.6	47.7	39.8	63.3	42.7	46.0	68.1	50.3	41.6	65.1	44.9
FocalNet-T (SRF)	48.6	267	**45.9** (+2.2)	**68.3**	**50.1**	**41.3**	**65.0**	**44.3**	**47.6** (+1.6)	**69.5**	**52.0**	**42.6**	**66.5**	**45.6**
FocalAtt-Tiny [63]	48.8	291	44.8	67.7	49.2	41.0	64.7	44.2	47.2	69.4	51.9	42.7	66.5	45.9
FocalNet-T (LRF)	48.9	268	**46.1** (+1.3)	**68.2**	**50.6**	**41.5**	**65.1**	**44.5**	**48.0** (+0.8)	**69.7**	**53.0**	**42.9**	**66.5**	**46.1**
ResNet101 [21]	63.2	336	40.4	61.1	44.2	36.4	57.7	38.8	42.8	63.2	47.1	38.5	60.1	41.3
ResNeXt101-32x4d [61]	62.8	340	41.9	62.5	45.9	37.5	59.4	40.2	44.0	64.4	48.0	39.2	61.4	41.9
PVT-Medium [55]	63.9	302	42.0	64.4	45.6	39.0	61.6	42.1	44.2	66.0	48.2	40.5	63.1	43.5
Twins-SVT-B [10]	76.3	340	45.2	67.6	49.3	41.5	64.5	44.8	48.0	69.5	52.7	43.0	66.8	46.6
Swin-Small [36]	69.1	354	46.5	68.7	51.3	42.1	65.8	45.2	48.5	70.2	53.5	43.3	67.3	46.6
FocalNet-S (SRF)	70.8	356	**48.0** (+1.5)	**69.9**	**52.7**	**42.7**	**66.7**	**45.7**	**48.9** (+0.4)	**70.1**	**53.7**	**43.6**	67.1	**47.1**
FocalAtt-Small [63]	71.2	401	47.4	69.8	51.9	42.8	66.6	46.1	48.8	70.5	53.6	**43.8**	67.7	47.2
FocalNet-S (LRF)	72.3	365	**48.3** (+0.9)	**70.5**	**53.1**	**43.1**	**67.4**	**46.2**	**49.3** (+0.5)	**70.7**	**54.2**	**43.8**	**67.9**	**47.4**
ResNeXt101-64x4d [61]	102.0	493	42.8	63.8	47.3	38.4	60.6	41.3	44.4	64.9	48.8	39.7	61.9	42.6
PVT-Large[55]	81.0	364	42.9	65.0	46.6	39.5	61.9	42.5	44.5	66.0	48.3	40.7	63.4	43.7
Twins-SVT-L [10]	119.7	474	45.9	-	-	41.6	-	-	-	-	-	-	-	-
Swin-Base [36]	107.1	497	46.9	69.2	51.6	42.3	66.0	45.5	48.5	69.8	53.2	43.4	66.8	46.9
FocalNet-B (SRF)	109.4	496	**48.8** (+1.9)	**70.7**	**53.5**	**43.3**	**67.5**	**46.5**	**49.6** (+1.1)	**70.6**	**54.1**	**44.1**	**68.0**	**47.2**
FocalAtt-Base [63]	110.0	533	47.8	70.2	52.5	43.2	67.3	46.5	49.0	70.1	53.6	43.7	67.6	47.0
FocalNet-B (LRF)	111.4	507	**49.0** (+1.2)	**70.9**	**53.9**	**43.5**	**67.9**	**46.7**	**49.8** (+0.8)	**70.9**	**54.6**	**44.1**	**68.2**	**47.2**

Table 5: COCO object detection and instance segmentation results with Mask R-CNN [20]. Grays rows are the numbers from our FocalNets.

going deeper improves the performance of FocalNets significantly. These results demonstrate that the commonly used model augmentation techniques developed for vision transformers can be easily adopted to improve the performance of FocalNets.

ImageNet-22K pretraining. We investigate the effectiveness of FocalNets when pretrained on ImageNet-22K which contains 14.2M images and 21K categories. Training details are described in the appendix. We report the results in Table 4. Though FocalNet-B/L are both pretrained with 224×224 resolution and directly transferred to target domain with 384×384 image size, we can see that they consistently outperform Swin Transformers.

4.2 Detection and Segmentation

Object detection and instance segmentation. We make comparisons on object detection with COCO 2017 [33]. We choose Mask R-CNN [20] as the detection method and use FocalNet-T/S/B pretrained on ImageNet-1K as the backbones. All models are trained on the 118k training images and evaluated on 5K validation images. We use two standard training recipes, $1\times$ schedule with 12 epochs and $3\times$ schedule with 36 epochs. Following [36], we use the same multi-scale training strategy by randomly resizing the shorter side of an image to $[480, 800]$. Similar to [63], we increase the kernel size k^ℓ by 6 for context aggregation at all focal levels to adapt to higher input resolutions. Instead of up-sampling the relative position biases as in [63], FocalNets uses simple zero-padding for the extra kernel parameters. This expanding introduces negligible overhead but helps extract longer range contexts. For training, we use AdamW [38] as the optimizer with initial learning rate 10^{-4} and weight decay 0.05. All models are trained with batch size 16. We set the stochastic drop rates to 0.1, 0.2, 0.3 in $1\times$ and 0.3, 0.5, 0.5 in $3\times$ training schedule for FocalNet-T/S/B, respectively.

The results are shown in Table 5. We measure both box and mask mAP, and report the results for both small and large receptive field models. Comparing with Swin Transformer, FocalNets improve the box mAP (AP^b) by 2.2, 1.5 and 1.9 in $1\times$ schedule for tiny, small and base models, respectively. In $3\times$ schedule, the improvements are still consistent and significant. Remarkably, the $1\times$ performance of FocalNet-T/B (45.9/48.8) rivals Swin-T/B (46.0/48.5) trained with $3\times$ schedule. When comparing with FocalAtt [63], FocalNets with large receptive fields consistently outperform under all settings and cost much less FLOPs. For instance segmentation, we observe the similar trend as that of object detection for FocalNets. To further verify the generality of FocalNets, we train three detection models, Cascade Mask R-CNN [1], Sparse RCNN [41] and ATSS [69] with FocalNet-T as the backbone. We train all models with $3\times$ schedule, and report the box mAPs in Table 6. As we can see, FocalNets bring clear gains to all three detection methods over the previous SoTA methods.

Semantic segmentation. We benchmark FocalNets on semantic segmentation, a dense prediction task that requires fine-grained understanding and long-range interactions. We use ADE20K [74] for our experiments and follow [36] to use UperNet [60] as the segmentation method. With FocalNet-

Method	Backbone	#Param.	FLOPs	AP^b	AP^b_{50}	AP^b_{75}
C. Mask R-CNN [1]	R-50 [21]	82.0	739	46.3	64.3	50.5
	DW-Net-T [19]	82.0	730	49.9	68.6	54.3
	Swin-T [36]	85.6	742	50.5	69.3	54.9
	FocalNet-T (SRF)	86.4	746	**51.5**	**70.1**	**55.8**
	FocalAtt-T [63]	86.7	770	51.5	**70.6**	55.9
	FocalNet-T (LRF)	87.1	751	**51.5**	70.3	**56.0**
Sparse R-CNN [41]	R-50 [21]	106.1	166	44.5	63.4	48.2
	Swin-T [36]	109.7	172	47.9	67.3	52.3
	FocalNet-T (SRF)	110.5	172	**49.6**	**69.1**	**54.2**
	FocalAtt-T [63]	110.8	196	49.0	69.1	53.2
	FocalNet-T (LRF)	111.2	178	**49.9**	**69.6**	**54.4**
ATSS [69]	R-50 [21]	32.1	205	43.5	61.9	47.0
	Swin-T [36]	35.7	212	47.2	66.5	51.3
	FocalNet-T (SRF)	36.5	215	**49.2**	**68.1**	**54.2**
	FocalAtt-T [63]	36.8	239	49.5	**68.8**	53.9
	FocalNet-T (LRF)	37.2	220	**49.6**	68.7	**54.5**

Table 6: A comparison between our FocalNet with previous CNNs/Transformers across different object detection methods, trained using the $3\times$ schedule.

Backbone	Crop Size	#Param.	FLOPs	mIoU	+MS
ResNet-101 [21]	512	86	1029	44.9	-
Twins-SVT-L [10]	512	133	-	48.8	50.2
DW-Net-T [19]	512	56	928	45.5	-
DW-Net-B [19]	512	132	924	48.3	-
Swin-T [36]	512	60	941	44.5	45.8
FocalNet-T (SRF)	512	61	944	**46.5**	**47.2**
FocalAtt-T [63]	512	62	998	45.8	47.0
FocalNet-T (LRF)	512	61	949	**46.8**	**47.8**
Swin-S [36]	512	81	1038	47.6	49.5
FocalNet-S (SRF)	512	83	1035	**49.3**	**50.1**
FocalAtt-S [63]	512	85	1130	48.0	**50.0**
FocalNet-S (LRF)	512	84	1044	**49.1**	**50.1**
Swin-B [36]	512	121	1188	48.1	49.7
FocalNet-B (SRF)	512	124	1180	**50.2**	**51.1**
FocalAtt-B [63]	512	126	1354	49.0	50.5
FocalNet-B (LRF)	512	126	1192	**50.5**	**51.4**

Table 7: Semantic segmentation on ADE20K [74]. All models are trained with UperNet [60]. Single- and multi-scale (MS) mIoU are reported on validation set.

Model	Formula	#Param.	FLOPs	Throughput	Top-1
FocalNet-T (LRF)	$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot h(\sum_{\ell=1}^{L+1} \boldsymbol{g}_i^\ell \cdot \boldsymbol{z}_i^\ell)$	28.6	4.49	696	82.3
→ **Depth-width ConvNet**	$\boldsymbol{y}_i = q(\mathsf{GeLU}(h(\boldsymbol{z}_i^L)))$	28.6	4.47	738	81.6 (-0.7)
→ **Pooling Aggregator**	$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot h(\sum_{\ell=1}^{L+1} \boldsymbol{g}_i^\ell \cdot \mathsf{Avg\text{-}Pool}(\boldsymbol{z}_i^{\ell-1}))$	28.3	4.37	676	80.5 (-1.8)
→ **Global Pooling Aggregator**	$\boldsymbol{y}_i = q(\boldsymbol{x}_i) \odot h(\boldsymbol{g}_i \cdot \mathsf{Avg\text{-}Pool}(f_z(X)))$	28.3	4.36	883	75.7 (-6.7)
→ **Multi-scale Self-Attention (QKV first)**	$\boldsymbol{y}_i = MHSA(\boldsymbol{x}_i, \boldsymbol{z}_i^1, ..., \boldsymbol{z}_i^{L+1}), f_z, q, h = Identity(\cdot)$	28.6	4.61	456	81.5 (-0.8)
→ **Multi-scale Self-Attention (QKV later)**	$\boldsymbol{y}_i = MHSA(\boldsymbol{x}_i, \boldsymbol{z}_i^1, ..., \boldsymbol{z}_i^{L+1}), f_z, q, h = Identity(\cdot)$	28.6	7.26	448	80.8 (-1.5)
→ **Sliding-window Self-Attention**	$\boldsymbol{y}_i = MHSA(\boldsymbol{x}_i, \mathcal{N}(\boldsymbol{x}_i)), \lvert\mathcal{N}(\boldsymbol{x}_i)\rvert = 7 \times 7 - 1$	28.3	4.49	103	81.5 (-0.8)

Table 8: We convert our FocalNet to other model types and report the performance.

T/S/B trained on ImageNet-1K as the backbones, we train UperNet for 160k iterations with input resolution 512×512 and batch size 16. For comparisons, we report both single- and multi-scale (MS) mIoU. Table 7 shows the results with different backbones. FocalNet outperforms Swin and Focal Transformer significantly under all settings. Even for the base models, FocalNet (SRF) exceeds Swin Transformer by 2.1 and 1.4 at single- and multi-scale, respectively. Compared with Focal Transformer, FocalNets outperform Focal Transformer, with a larger gain than that of Swin Transformer, and consume much less FLOPs. These results demonstrate the superiority of FocalNets on the pixel-level dense prediction tasks, in addition to the instance-level object detection task.

4.3 Network Inspection

Model Variants. We compare in Table 8 six different model variants derived from FocalNet.

- **Depth-wise ConvNet.** It feeds the feature vectors at the top level L to a two-layer MLP. The resultant model is close to DW-Net [19]. Although it can achieve 81.6% accuracy, surpassing Swin Transformer (81.3%), it underperforms FocalNet by 0.7%. Focal modulation uses depth-wise convolution as a component but further aggregates hierarchical contexts and combines them with fine-grained query features through modulation.
- **Pooling Aggregator.** It replaces the depth-wise convolution module with average pooling, and is similar to MetaFormer [66] in terms of token aggregation. Average pooling has slightly lower complexity but leads to a significant drop of accuracy (1.8%). Compared with depth-wise convolution, average pooling is permutation-invariant and thus incapable of capturing visual structures, which interprets the performance degradation.
- **Global Pooling Aggregator.** It removes local aggregations at all levels and only keeps the global one ($\mathbf{Z}^{L+1}$). This variant resembles SENet [26]. It turns out that global context alone is insufficient for visual modeling, leading to a significant 6.7% drop.
- **Multi-scale Self-Attention.** Given the summarized tokens at different levels, a straightforward way to combine them is performing a SA among all of them. We have developed two SA methods: computing q, k, v before and after aggregation, respectively. Both methods result in some visible performance drop and increase the run time latency, compared to FocalNet.
- **Sliding-window Self-Attention.** Finally, we apply a sliding-window SA for each visual token within a window. Since it involves dense interactions for each fine-grained tokens, the time and memory cost explodes, and the performance is worse than FocalNet.

Model	FLOPs	Throughput	Top-1	AP^b	AP^m
FocalNet-T (LRF)	4.48	696	82.3	46.2	41.6
Additive	4.49	670	81.5 (-0.8)	45.6 (-0.6)	41.1 (-0.5)
No global pool	4.48	683	82.0 (-0.3)	45.8 (-0.4)	41.2 (-0.4)
Top-only	4.49	698	81.9 (-0.4)	45.7 (-0.5)	41.2 (-0.4)
No gating	4.48	707	81.9 (-0.4)	45.6 (-0.6)	41.1 (-0.5)

Table 9: Component analysis for focal modulation. Four separate changes are made to the original FocalNet. Throughput is reported on image classification. All variants have almost the same size (28.6M) as the default model.

Levels (Kernels)	Receptive Field	#Param.	FLOPs	Throughput	Top-1
2 (3-5)	7	28.4	4.41	743	82.1
3 (3-5-7)	13	28.6	4.49	696	82.3
0 (n/a)	0	28.3	4.35	883	75.7
1 (3)	3	28.3	4.37	815	82.0
4 (3-5-7-9)	21	29.0	4.59	592	82.2
1 (13)	13	28.8	4.59	661	81.9

Table 10: Model performance with number of focal levels L. "Receptive Field" refers to effective receptive field at the top level regardless of the global average pooling.

Component Analysis. Here we ablate FocalNet to study the relative contribution of each component. The result is reported in Table 9, where we investigate the impact of the following model architecture changes on model performance:

- **Replacing Multiplication with Addition**: we change the element-wise multiplication to addition in Eq. (6), which converts the modulator into a bias term. This leads to 0.7% accuracy drop, which indicates that element-wise multiplication is a more powerful way of modulation than addition.
- **No Global Aggregation**: we remove the top global average pooling in focal modulation. It hurts the performance by 0.3%. Even though the hierarchical aggregation already covers a relatively large receptive field, global information ($\mathbf{Z}^{L+1}$) is still useful for capturing global context.
- **Top-only Aggregation**: Instead of aggregating the feature maps from all focal levels, we only use the top level map. In this case, the features at lower levels that are more "local" and "fine-grained" are completely discarded. This change leads to 0.4% performance drop, which verifies our hypothesis that features at different levels and spatial scopes compensate each other.
- **None-gating Aggregation**: We remove the gating mechanism when aggregating the multiple levels of feature maps. This causes 0.4% drop. As we discussed earlier, the dependencies between visual token (query) and its surroundings differ based on the query content. The proposed gating mechanism helps the model to *adaptively* learn where and how much to interact.

We study the effect of varying the focal level (*i.e.* the number of depth-wise convolution layers L). In our experiments reported above, the results show that large receptive field in general achieves better performance (LRF *v.s.* SRF). Here, we investigate by further altering L. In additional to setting $L = 2$ and 3, we also try $L = 0$, $L = 1$, and $L = 4$. As shown in Table 10, increasing L brings slight improvement and finally reaches a plateau. Surprisingly, a single level with kernel size 3 can already obtain a decent performance. When we increase the single-level kernel size from 3 to 13, there is a slight 0.1% drop, and a 0.4% gap to the one with three levels but same size of receptive field (second row). This indicates that simply increasing the receptive field does not necessarily improve the performance, and a hierarchical aggregation for both fine- and coarse-grained context is crucial. We recommend $L = 2, 3$ as a good accuracy-speed trade-off.

Monolithic Architectures. We replace all SA modules in ViTs with focal modulation to construct monolithic FocalNet-T/S/B. We use three focal levels with kernel sizes 3,5 and 7, so that the effective receptive field is close to the global SA in ViT. As shown in Table 11, FocalNets consistently outperform ViT counterparts, with comparable FLOPs and inference speed.

Model	Dim	#Param.	FLOPs	Th. (imgs/s)	Top-1
ViT-T/16	192	5.7	1.3	2834	72.2
FocalNet-T/16	192	5.9	1.1	2334	**74.1** (+1.9)
ViT-S/16	384	22.1	4.6	1060	79.9
FocalNet-S/16	384	22.4	4.3	920	**80.9** (+1.0)
ViT-B/16	768	86.6	17.6	330	81.8
FocalNet-B/16	768	87.2	16.9	300	**82.4** (+0.6)

Table 11: Comparisons between FocalNet and ViT both with monolithic architectures.

5 Conclusion

We have proposed *focal modulation*, a new mechanism that enables input-dependent token interactions for visual modeling. It consists of a hierarchical contextualization to gather for each query token its contexts from short to long ranges, a gated aggregation to adaptively aggregate context features based on the query content, followed by a simple modulation. With *focal modulation*, we built a series of simple yet attention-free Focal Modulation Networks (FocalNets). Extensive experiments show that FocalNets significantly outperform the SoTA SA counterparts (*e.g.,* Swin and Focal Transformer) with similar time-/memory-cost on the tasks of image classification, object detection and semantic segmentation. These encouraging results render focal modulation a favorable alternative to SA for effective *and* efficient visual modeling.

References

[1] 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.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[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] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Jiashi Feng. Augmented transformer with adaptive graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024*, 2021.

[5] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification, 2021.

[6] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. CycleMLP: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021.

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *arXiv preprint arXiv:2103.15436*, 2021.

[8] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020.

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021.

[11] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020.

[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.

[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. *arXiv preprint arXiv:2107.00652*, 2021.

[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] Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021.

[16] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

[17] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. *arXiv preprint arXiv:2108.13341*, 2021.

[18] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020.

[19] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv preprint arXiv:2106.04263*, 2021.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] 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.

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[23] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *arXiv preprint arXiv:2106.12368*, 2021.

[24] 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.

[25] 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*, pages 3464–3473, 2019.

[26] 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.

[27] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[28] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. *arXiv preprint arXiv:2105.04447*, 2021.

[29] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12321–12330, 2021.

[30] Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. Trear: Transformer-based rgb-d egocentric action recognition. *arXiv preprint arXiv:2101.03904*, 2021.

[31] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

[32] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*, 2021.

[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 *ECCV*, 2014.

[34] Hanxiao Liu, Zihang Dai, David So, and Quoc Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34, 2021.

[35] Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to MLPs. *arXiv preprint arXiv:2105.08050*, 2021.

[36] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[37] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Yuxuan Lou, Fuzhao Xue, Zangwei Zheng, and Yang You. Sparse-mlp: A fully-mlp architecture with conditional computation. *arXiv preprint arXiv:2109.02008*, 2021.

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

[41] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020.

[42] 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.

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

[44] Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. Sparse mlp for image recognition: Is self-attention really necessary? *arXiv preprint arXiv:2109.05422*, 2021.

[45] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. *arXiv preprint arXiv:2111.12294*, 2021.

[46] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. MLP-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[47] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.

[48] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. ResMLP: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

[49] 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.

[50] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation. *arXiv preprint arXiv:2112.13692*, 2021.

[51] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021.

[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 *NeurIPS*, 2017.

[53] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020.

[54] Ning Wang, Wengang Zhou, Jie Wang, and Houqaing Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. *arXiv preprint arXiv:2103.11681*, 2021.

[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. *arXiv preprint arXiv:2102.12122*, 2021.

[56] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[57] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020.

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

[59] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[60] 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.

[61] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[62] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021.

[63] 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. *arXiv preprint arXiv:2107.00641*, 2021.

[64] Jianwei Yang, Zhile Ren, Chuang Gan, Hongyuan Zhu, and Devi Parikh. Cross-channel communication networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1297–1306, 2019.

[65] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S^2-MLPv2: Improved spatial-shift mlp architecture for vision. *arXiv preprint arXiv:2108.01072*, 2021.

[66] 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. *arXiv preprint arXiv:2111.11418*, 2021.

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

[68] 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. *arXiv preprint arXiv:2103.15358*, 2021.

[69] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020.

[70] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[71] Jiaojiao Zhao, Xinyu Li, Chunhui Liu, Shuai Bing, Hao Chen, Cees GM Snoek, and Joseph Tighe. Tuber: Tube-transformer for action detection. *arXiv preprint arXiv:2104.00969*, 2021.

[72] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Mixing and shifting: Exploiting global and local dependencies in vision mlps. *arXiv preprint arXiv:2202.06510*, 2022.

[73] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020.

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

[75] Daquan Zhou, Yujun Shi, Bingyi Kang, Weihao Yu, Zihang Jiang, Yuan Li, Xiaojie Jin, Qibin Hou, and Jiashi Feng. Refiner: Refining self-attention for vision transformers. *arXiv preprint arXiv:2106.03714*, 2021.

[76] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] in supplementary material
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] in supplementary material
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]