Focal Modulation Networks

Jianwei Yang, Chunyuan Li, and Jianfeng Gao

Microsoft Research, Redmond
{jianwyan,chunyl,jfgao}@microsoft.com

Abstract. In this work, we propose *focal modulation network* (*FocalNet* in short), where self-attention (SA) is completely replaced by a *focal modulation* module that is more effective and efficient 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 at different granularity levels, (*ii*) gated aggregation to selectively aggregate context features for each visual token (query) based on its content, and (*iii*) modulation or element-wise affine transformation to fuse the aggregated features into the query vector. Extensive experiments show that FocalNets outperform the state-of-the-art SA counterparts (*e.g.*, Swin Transformers) with similar time and memory cost on the tasks of image classification, object detection, and semantic segmentation. Specifically, our 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. FocalNets exhibit remarkable superiority when transferred to downstream tasks. For object detection with Mask R-CNN, our FocalNet base trained with 1× already surpasses Swin trained with 3× schedule (**49.0** *v.s.* 48.5). For semantic segmentation with UperNet, FocalNet base evaluated at single-scale outperforms Swin evaluated 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 in real-world applications. Code is available at: https://github.com/microsoft/FocalNet.

Keywords: Focal Modulation Network, Self-Attention, Depth-wise Convolution, Image Classification, Object Detection, Segmentation.

1 Introduction

Transformers [60], originally proposed for natural language processing (NLP), have become a prevalent architecture in computer vision since the seminal work of Vision Transformer (ViT) [16]. Its promise has been demonstrated in various vision tasks including image classification [56,63,67,41,78,59], object detection [3,87,83,12], segmentation [61,65,9], and beyond [35,81,4,7,62,33].

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.

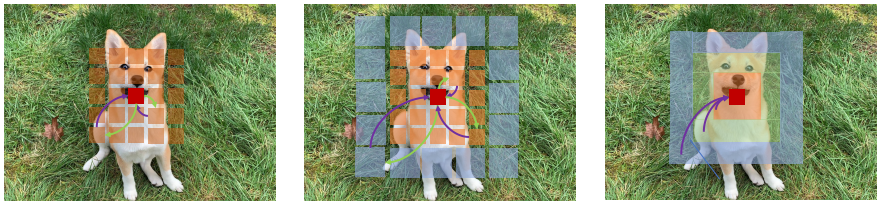

Fig. 1: Illustrative comparison among (a) Window-wise Self-Attention (SA) [41], (b) Focal Attention (FA) [71] and (c) the proposed Focal Modulation. Given the query token, window-wise SA captures spatial context from its surrounding tokens, FA, in addition, uses far-away summarized tokens, and Focal 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.

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 [63], window attention [41,59,78], or the combination [71,10]. 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 [67,18,70,17,15]. Interestingly, these works exhibit a clear trend of merging SA and convolution for visual modeling. Nevertheless, all of them heavily rely on SA and thus still suffer from the efficiency problem.

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 [41]. With a simple window-shift strategy, Swin attains superior performance to ResNets across various vision tasks. To enlarge the receptive field, focal attention [71] 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. Our method is inspired by focal attention [71] in that both methods perform multiple levels of aggregation to capture fine- and coarse-grained visual contexts. However, unlike focal attention that extracts the summarized tokens at target locations followed by attention, our method extracts at each query position and replaces the attention with a simple modulation, *i.e.*, element-wise affine transformation. Hence, 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, *Focal Modulation Network* (or *FocalNet* in short).

In addition to the capacity of input-dependent long-range interactions as that in SA and focal attention, focal modulation has several merits: (*i*) it can naturally leverage the built-in convolution operation for fast and translation-invariant context encoding (or contextualization); (*ii*) it does not require window partitioning, positional embedding, separate heads and softmax *etc.*, which allows fast adaptation to different resolutions and tasks; and (*iii*) with a few stacked contextualization levels, it can rapidly capture a large effective receptive field, thus is more efficient for high-resolution image encoding than SA and focal attention models. 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 [38], our FocalNets with tiny and base model size achieve **46.1** and **49.0** box mAP on Mask R-CNN 1×, surpassing Swin with 3× schedule (46.0 and 48.5 box mAP). For semantic segmentation on ADE20k [85], we observe a similar trend that 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 (**74.1**% *v.s.* 72.2%, **80.9**% *v.s.* 79.9% and **82.4**% *v.s.* 81.8% for tiny, small and base models). With all these encouraging results, we hereby recommend FocalNets to be a strong alternative architecture for effective and efficient visual modeling.

2 Related Work

Self-attentions. Self-attention (SA) is first introduced in Transformer [60], which then stimulates a revolution in natural language processing. Later, Vision Transformer (ViT) [16] introduces self-attention for generic image encoding 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 [24] when training with optimized recipes [16,56]. Afterwards, multi-scale architectures [5,63,70], light-weight convolution layers [67,18,36], local self-attention mechanisms [41,78,10] and learnable attention weights [75] have been proposed to boost the performance and support high-resolution input. More comprehensive surveys are included in [32,21,32]. Our focal modulation is aimed at recovering the ability of capturing input-dependent long-range dependencies in SA. However, it 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 [71] where both extract contexts of different granularity to efficiently capture large receptive field. However, focal modulation differs form focal attention significantly in that it performs the context aggregation at each query location instead of target locations, 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 [74] which uses a pooling to summarize the local context and a simple subtraction to adjust the individual inputs. Though achieving decent efficiency for its simplicity, Poolformer lags behind popular vision transformers like Swin on performance. As we will show later, capturing local structures at different levels and performing modulation are both essential to the final superior performance.

MLP architectures. Several works show that building models solely based on multi-layer perceptrons (MLPs) achieves surprisingly competitive results on ImageNet classification with only spatial- and channel-wise token mixing [53,55,40,37]. Visual MLPs can be categorized into two groups. (*i*) Global-mixing MLPs, such as MLP-Mixer [53] and ResMLP [55], perform global communication between visual tokens through spatial-wise projections, by exploring efficient interactions among all tokens with various techniques, such as gating, routing, and Fourier transforms [40,45,51,52]. (*ii*) Local-mixing MLPs sample nearby tokens for interactions, using spatial shifting, permutation, and pseudo-kernel mixing [73,26,37,6,19]. Recently, Mix-Shift-MLP [82] exploits both local and global interactions with MLPs, in a similar spirit of focal attention [71]. Both MLP architectures and our focal modulation network are attention-free architecture and share the same goal of improving efficiency. However, our focal modulation with multi-level context aggregation naturally capture 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 neural networks in computer vision, due to several desirable properties: built-in inductive biases such as translation equivalence and computation sharing, and the "sliding window" strategy which is intrinsic to visual processing. The field has evolved rapidly since the emerge of VGG [46], InceptionNet [48] and ResNet [24]. Representative works that focus on the efficiency of ConvNets are MobileNet [27],

ShuffleNet [80] and EfficientNet [50]. Another line of works aimed at integrating global context to compensate ConvNets such as SE-Net [29], Non-local Network [64], GCNet [2], LR-Net [28] and C3Net [72], *etc.* Introducing dynamic operation is another way to augment ConvNets as demonstrated in Involution [34] and DyConv [8]. Recently, ConvNets strike back and ignite an increasingly intensive debate on the comparison with ViTs mainly for two reasons: (*i*) a number of works employ convolution layers and similar architecture designs to augment the SA and bring significant gains [67,18,36,17] or the vice versa [57]; (*ii*) ResNets have closed the gap to ViTs using similar data augmentation and regularization strategies [66], and replacing SA with (dynamic) depth-wise convolution can surprisingly surpass Swin slightly [22]. Our focal modulation network falls into ConvNets family broadly. It also exploits depth-wise convolution as the micro-architecture but goes beyond by introducing a multi-level context aggregation and input-dependent modulation to adjust the individual tokens adaptively. We will show this new module significantly outperforms raw depth-wise convolution.

Concurrent works. We notice three concurrent works, including ConvNeXT [42], RepLKNet [14] and Visual Attention Networks (VAN) [20]. All these works are motivated by large receptive field and exploit convolutions with large or dilated kernels as the main building block. However, our method differs from ConvNeXT and RepLKNet in that it uses a modulation module to adaptively adjust the individual tokens with the gathered context. In this sense, our FocalNet is close to VAN but exploits an adaptive aggregation of multi-level visual contexts.

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 this paper, 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),

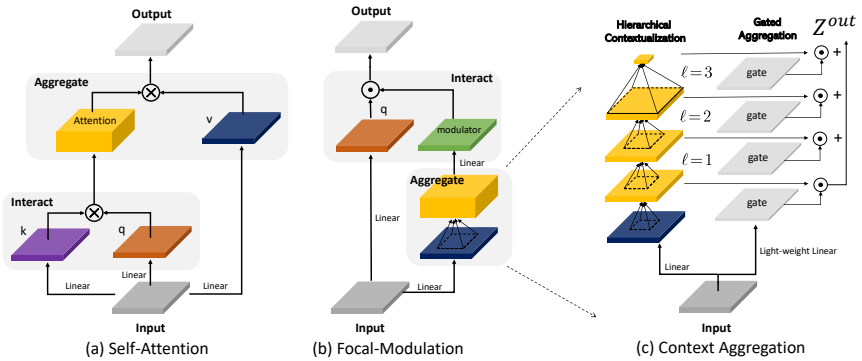

Fig. 2: Left: Comparing SA (a) and focal modulation (b) side by side. Right: Detailed illustration of context aggregation in focal modulation (c).

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 the architectures of 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 $\mathbf{X}$.
- **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**. $q(\cdot)$ is applied to individual tokens to preserve the finest information, while $\mathcal{M}_2$ extracts the coarser grained context. They are decoupled but adaptively 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-range and long-range contexts are important for visual modeling [71,15,42]. 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 [71], 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 [25]. The use of depth-wise convolution for hierarchical contextualization of Eq. (4) is motivated by its desirable properties. Compared to pooling [74,29], 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. Larger receptive fields capture more global contexts with coarser granularity. 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 of the query. For example, we might heavily rely on local fine-grained features for encoding the queries of salient visual objects (*e.g.,* the "dog" in Fig. 1), but mainly global coarse-grained features for the queries of background scenes (*e.g.,* the "grass" in Fig. 1). Based on this intuition, we use a gating mechanism to control how much to aggregate from feature maps at different levels $\mathbf{Z}^\ell$ 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 $\mathbf{X}$,

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

Algorithm 1: Pseudo code for Focal Modulation.

```
# Input/output shape: (B, H, W, C); Batchsize B; Feature map height H, width W, dim C
# Focal levels: L; Conv kernel size at level ℓ: k^ℓ
def init( ):
    pj_in, pj_cxt = Linear(C, 2*C + (L+1)), Conv2d(C, C, 1)
    hc_layers = [Sequential(Conv2d(C, C, k^ℓ, groups=C), GeLU()) for ℓ in range(L)]
    pj_out = Sequential(Linear(C, C), Dropout())
def forward(x, m=0):
    x = pj_in(x).permute(0, 3, 1, 2)
    q, z, gate = split(x, (C, C, L+1), 1)
    for ℓ in range(L):
        z = hc_layers[ℓ](z)           # Eq.(4), hierarchical contextualization
        m = m + z * gate[:, ℓ:ℓ+1]    # Eq.(5), gated aggregation
    m = m + GeLU(z.mean(dim=(2,3))) * gate[:,L:]
    x = q * pj_cxt(m)                 # Eq.(6), focal modulation
    return  pj_out( x.permute(0, 2, 3, 1) )
```

where $\mathbf{G}^\ell \in \mathbb{R}^{H\times W\times 1}$ is a slice of $\mathbf{G}$ for the level ℓ. 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. We summarize the proposed focal modulation procedure in Pytorch-style pseudo code in Algorithm 1. As we can see, it can be easily implemented with a few convolution and linear layers.

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 $f_a^{\{1,\dots,L\}}$ 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 $\mathcal{O}(HW \times (3C^2 + 2Cw^2))$.

3.3 Discussions

Window-wise SA is performed based on the following formula:

$$\boldsymbol{y}_i = \sum_{j \in \mathcal{N}(i)} \mathsf{Softmax}(\frac{q(\boldsymbol{x}_i)k(\mathbf{X})^\top}{\sqrt{C}})_j v(\boldsymbol{x}_j) \tag{7}$$

where q, k, v are three linear projection functions, $\mathcal{N}(\cdot)$ is the set of token indices in the neighborhood defined by the window. In Eq. (7), a heavy interaction between the query token and all target tokens is needed before the weighted sum. In contrast, in the proposed focal modulation in Eq. (6), $q(\boldsymbol{x}_i)$ is taken out of the summation over $\mathcal{N}(i)$, making the computation of token-wise interactions light-weight and decoupled with the feature aggregation.

Depth-wise Convolution has been used to augment the local structural modeling for SA [67,15,18] or enable efficient long-range interactions [27,22,42]. Though not constrained, our focal modulation also employs depth-wise convolution to build the hierarchical context representations, and the resultant focal modulation networks broadly belong to the ConvNet family. According to Eq. (6), focal modulation recovers depth-wise convolutions when removing the hierarchical aggregation and modulation, which however are both essential as demonstrated in our experiments.

Squeeze-and-Excitation (SE) and PoolFormer can also be considered as special cases of focal modulation. SE exploits a global average pooling to get the squeezed global context representation, and then a multi-layer perception (MLP) followed by a $\mathsf{Sigmoid}$ to obtain the excitation scalar or modulator for each channel. In contrast, focal modulation is input-dependent in that it extracts the "squeezed" and "focal" context specifically for each query token. Setting $L = 0$, focal modulation becomes $q(\boldsymbol{x}_i) \odot h(f_g(\boldsymbol{x}_i) \cdot \mathsf{Avg\text{-}Pool}(f_z(\mathbf{X})))$ which closely approximates SE. On the other hand, PoolFormer uses sliding-window average pooling to extract the context and subtraction for modulation.

3.4 Network Architectures

To examine the efficacy of focal modulation, we compare it with the previous SoTA methods, *e.g.* Swin [41] and Focal [71] Transformers. We observe in our experiments that different configurations (*e.g.*, depths, dimensions, *etc*) lead to different performance. For a fair comparison, we use the same stage layouts and hidden dimensions, 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.*, $k^\ell = k^{\ell-1} + 2$. The configurations of four FocalNet variants are summarized in Table 1. To match the complexities of Swin and Focal Transformers, we design a small receptive field

Name	Depth	Dimension (d)	Levels (L)	Kernel Size (k^1)	Receptive Field (r^L)
FocalNet-T (SRF/LRF)	[2,2,6,2]	[96,192,384,768]			
FocalNet-S (SRF/LRF)	[2,2,18,2]	[96,192,384,768]	[2,2,2,2]	[3,3,3,3]	[7,7,7,7]
FocalNet-B (SRF/LRF)	[2,2,18,2]	[128,256,512,1024]	[3,3,3,3]	[3,3,3,3]	[13,13,13,13]
FocalNet-L (SRF/LRF)	[2,2,18,2]	[192,384,768,1536]			

Table 1: Model configurations at four stages for FocalNet. The depth layouts and hidden dimension (d) are the same to Swin [41] and Focal Transformers [71]. SRF and LRF means small and large receptive field, respectively. The only difference is the number of focal levels (L) and starting kernel size ($k^{\ell=1}$). The last column lists the effective receptive field at top focal level at each stage (r^L).

(SRF) and a large receptive field (LRF) version for each of the four layouts by using 2 and 3 focal levels, respectively. We also 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.

4 Experiment

4.1 Image Classification

We compare different methods on ImageNet-1K classification [13]. Following the recipes in [56,41,71], 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, respectively. The results are reported in Table 2. We see that FocalNets outperform the conventional CNNs (*e.g.*, ResNet [24] and the augmented version [66]), MLP architectures such as MLP-Mixer [54] and gMLP [39], and Transformer architectures DeiT [56] and PVT [63]. In particular, we compare FocalNet 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, focal modulation's run-time speed is much higher than FocalAtt by getting rid of many time-consuming operations like rolling and unfolding. When comparing LRF with SRF models, we observe 0.2, 0.1 and 0.2 improvements, respectively, with minor extra overhead. This indicates that increasing the receptive field can in general improve the performance, which we discuss in more detail in the following sections. Overall, the results demonstrate that the proposed focal modulation is a strong competitor to the commonly used methods for visual token interactions.

Model	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 (%)
ResNet-50 [24]	25.0	4.1	1294	76.2
ResNet-101 [24]	45.0	7.9	745	77.4
ResNet-152 [24]	60.0	11.0	522	78.3
ResNet-50-SB [66]	25.0	4.1	1294	79.8
ResNet-101-SB [66]	45.0	7.9	745	81.3
ResNet-152-SB [66]	60.0	11.6	522	81.8
DW-Net-T [22]	24.2	3.8	1030	81.2
DW-Net-B [22]	74.3	12.9	370	83.2
Mixer-B/16 [54]	59.9	12.7	455	76.4
gMLP-S [39]	19.5	4.5	785	79.6
gMLP-B [39]	73.4	15.8	301	81.6
ResMLP-S24 [55]	30.0	6.0	871	79.4
ResMLP-B24 [55]	129.1	23.0	61	81.0
DeiT-Small/16 [56]	22.1	4.6	939	79.9
DeiT-Base/16 [56]	86.6	17.5	291	81.8
PVT-Small [63]	24.5	3.8	794	79.8
PVT-Medium [63]	44.2	6.7	517	81.2
PVT-Large [63]	61.4	9.8	352	81.7
PoolFormer-m36 [74]	56.2	8.8	463	82.1
PoolFormer-m48 [74]	73.5	11.6	347	82.5
Swin-Tiny [41]	28.3	4.5	760	81.2
FocalNet-T (SRF)	28.4	4.4	743	**82.1**
Swin-Small [41]	49.6	8.7	435	83.1
FocalNet-S (SRF)	49.9	8.6	434	**83.4**
Swin-Base [41]	87.8	15.4	291	83.5
FocalNet-B (SRF)	88.1	15.3	280	**83.7**
FocalAtt-Tiny [71]	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 [71]	89.8	16.4	138	83.8
FocalNet-B (LRF)	88.7	15.4	269	**83.9**

Table 2: ImageNet-1K classification for models trained with ImageNet-1K. We particularly compare FocalNets with Swin [41] and FocalAtt [71].

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 3: 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 4: Effect of deeper and thinner networks.

Model	Img. Size	#Params	FLOPs	Throughput	Top-1
ResNet-101x3 [24]	384^2	388.0	204.6	-	84.4
ResNet-152x4 [24]	480^2	937.0	840.5	-	85.4
ViT-B/16 [16]	384^2	86.0	55.4	99	84.0
ViT-L/16 [16]	384^2	307.0	190.7	30	85.2
Swin-Base [41]	$224^2/224^2$	88.0	15.4	291	85.2
FocalNet-B	$224^2/224^2$	88.1	15.3	280	**85.6**
Swin-Base [41]	$384^2/384^2$	88.0	47.1	91	86.4
FocalNet-B	$224^2/384^2$	88.1	44.8	94	**86.5**
Swin-Large [41]	$224^2/224^2$	196.5	34.5	155	86.3
FocalNet-L	$224^2/224^2$	197.1	34.2	144	**86.5**
Swin-Large [41]	$384^2/384^2$	196.5	104.0	49	**87.3**
FocalNet-L	$224^2/384^2$	197.1	100.6	50	**87.3**

Table 5: 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.

Model augmentation. Above we conducted a strictly fair comparison with Swin and Focal Transformer by only replacing the token interaction module but keeping all the other parts the same. Recently, many works reported superior performance using other orthogonal techniques such as overlapped patch embedding [18], deeper architectures [15,86], *etc.* Here, we investigate whether these commonly used techniques can also improve the performance of FocalNets. First, we study the effect of using overlapped patch embedding for downsampling. Following [67], 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 3. 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 [15]. In Table 4, 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, we reduce the initial hidden dimension from 96, 128 to 64, 96, respectively. These changes lead to smaller model sizes and FLOPs, but lower throughput due to the increased number of sequential blocks. It turns out that going deeper improves the performance of FocalNets significantly. These results demonstrate that the commonly used model augmentation techniques developed for Transformers can be easily adopted to improve the performance of FocalNets. We leave the incorporation of other

Model	Depth	Dim	#Param.	FLOPs	Throughput (imgs/s)	Top-1
ViT-T/16	12	192	5.7	1.3	2834	72.2
FocalNet-T/16	12	192	5.9	1.1	2334	**74.1** (+1.9)
ViT-S/16	12	384	22.1	4.6	1060	79.9
FocalNet-S/16	12	384	22.4	4.3	920	**80.9** (+1.0)
ViT-B/16	12	768	86.6	17.6	330	81.8
FocalNet-B/16	12	768	87.2	16.9	300	**82.4** (+0.6)

Table 6: Comparisons between FocalNet and ViT both with monolithic architectures. We use three focal levels ($L = 3$) in the focal modulation modules.

augmentation techniques, such as convolutional FFN [18], token labeling [31] to future work.

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 5. 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, indicating that FocalNets are equally or more scalable and data-efficient.

Monolithic Architectures. We study whether focal modulation can generalize to monolithic architectures in ViT [16], by replacing all SA modules in ViT with the focal modulation modules to construct monolithic FocalNet-T/S/B. For focal modulation, 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 6, FocalNets achieve much better performance than their ViT counterparts, with relatively small reduction of inference speed (18% for tiny and 10% for small and base models). At the tiny scale, FocalNet outperforms ViT by 1.9%. At the base scale, it surpasses ViT by 0.6%. Note that the FLOPs of FocalNets are smaller than ViTs. It means that we can potentially accelerate the inference speed of FocalNets. These results show that focal modulation can be generalized well to monolithic architectures, and its superiority to SA can be potentially transferred to other domains such as NLP, where the monolithic architecture is denominating. We leave it to future work.

4.2 Detection and Segmentation

Object detection and instance segmentation. We make comparisons on object detection with COCO 2017 [38]. We choose Mask R-CNN [23] 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 [41], we use the same multi-scale training strategy by randomly resizing the shorter side of an image to $[480, 800]$. Similar to [71], we increase the kernel size k^ℓ by 6 for context

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 [24]	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[63]	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
ViL-Small [78]	45.0	174	41.8	64.1	45.1	38.5	61.1	41.4	43.4	64.9	47.0	39.6	62.1	42.4
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 [41]	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 [71]	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 [24]	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 [69]	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 [63]	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
ViL-Medium [78]	60.1	261	43.4	65.9	47.0	39.7	62.8	42.1	44.6	66.3	48.5	40.7	63.8	43.7
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 [41]	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 [71]	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 [69]	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[63]	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
ViL-Base [78]	76.1	365	45.1	67.2	49.3	41.0	64.3	44.2	45.7	67.2	49.9	41.3	64.4	44.5
Twins-SVT-L [10]	119.7	474	45.9	-	-	41.6	-	-	-	-	-	-	-	-
Swin-Base [41]	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 [71]	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 7: COCO object detection and instance segmentation results with Mask R-CNN [23]. Grays rows are the numbers from our FocalNets.

aggregation at all focal levels to adapt to higher input resolutions. Instead of upsampling the relative position biases as in [71], 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 [44] 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× and 0.3, 0.5, 0.5 in 3× training schedule for FocalNet-T/S/B, respectively.

The results are shown in Table 7. 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× schedule for tiny, small and base models, respectively. In 3× schedule, the improvements are still consistent and significant. Remarkably, the 1× performance of FocalNet-T/B (45.9/48.8) rivals Swin-T/B (46.0/48.5) trained with 3× schedule. When comparing with FocalAtt [71], 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 [47] and ATSS [79] with FocalNet-T as the backbone. We train all models with 3× schedule, and report the box mAPs in Table 8. As we can see, FocalNets bring clear gains to all three detection methods over the previous SoTA methods.

Semantic segmentation. Finally, we benchmark FocalNets on semantic segmentation, a dense prediction task that requires fine-grained understanding and long-range interactions. We use ADE20K [85] for our experiments and follow [41]

Method	Backbone	#Param.	FLOPs	AP^b	AP^b_{50}	AP^b_{75}
C. Mask R-CNN [1]	R-50 [24]	82.0	739	46.3	64.3	50.5
	DW-Net-T [22]	82.0	730	49.9	68.6	54.3
	Swin-T [41]	85.6	742	50.5	69.3	54.9
	FocalNet-T (SRF)	86.4	746	**51.5**	**70.1**	**55.8**
	FocalAtt-T [71]	86.7	770	51.5	**70.6**	55.9
	FocalNet-T (LRF)	87.1	751	**51.5**	70.3	**56.0**
Sparse R-CNN [47]	R-50 [24]	106.1	166	44.5	63.4	48.2
	Swin-T [41]	109.7	172	47.9	67.3	52.3
	FocalNet-T (SRF)	110.5	172	**49.6**	**69.1**	**54.2**
	FocalAtt-T [71]	110.8	196	49.0	69.1	53.2
	FocalNet-T (LRF)	111.2	178	**49.9**	**69.6**	**54.4**
ATSS [79]	R-50 [24]	32.1	205	43.5	61.9	47.0
	Swin-T [41]	35.7	212	47.2	66.5	51.3
	FocalNet-T (SRF)	36.5	215	**49.2**	**68.1**	**54.2**
	FocalAtt-T [71]	36.8	239	49.5	**68.8**	53.9
	FocalNet-T (LRF)	37.2	220	**49.6**	68.7	**54.5**

Table 8: A comparison between our FocalNet with previous CNNs/Transformers across different object detection methods, trained using the 3× schedule.

Backbone	Crop Size	#Param.	FLOPs	mIoU	+MS
ResNet-101 [24]	512	86	1029	44.9	-
Twins-SVT-L [10]	512	133	-	48.8	50.2
DW-Net-T [22]	512	56	928	45.5	-
DW-Net-B [22]	512	132	924	48.3	-
Swin-T [41]	512	60	941	44.5	45.8
FocalNet-T (SRF)	512	61	944	**46.5**	**47.2**
FocalAtt-T [71]	512	62	998	45.8	47.0
FocalNet-T (LRF)	512	61	949	**46.8**	**47.8**
Swin-S [41]	512	81	1038	47.6	49.5
FocalNet-S (SRF)	512	83	1035	**49.3**	**50.1**
FocalAtt-S [71]	512	85	1130	48.0	**50.0**
FocalNet-S (LRF)	512	84	1044	**49.1**	**50.1**
Swin-B [41]	512	121	1188	48.1	49.7
FocalNet-B (SRF)	512	124	1180	**50.2**	**51.1**
FocalAtt-B [71]	512	126	1354	49.0	50.5
FocalNet-B (LRF)	512	126	1192	**50.5**	**51.4**

Table 9: Semantic segmentation on ADE20K [85]. All models are trained with UperNet [68]. Single- and multi-scale evaluations are reported on validation set in last two columns.

to use UperNet [68] as the segmentation method. With FocalNet-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 9 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. In Sec. 3.3, we draw the connection between FocalNet and other token interaction methods. Here, we compare in Table 10 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 [22]. Although it can achieve 81.6% accuracy, surpassing Swin Transformer (81.3%), it underperforms FocalNet by 0.7%. As we discussed earlier, focal modulation uses depth-wise convolution as a component but goes beyond it in that focal modulation aggregates contexts at different levels of granularity 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 [74] 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.

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

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 11: 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 12: 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.

- **Global Pooling Aggregator.** It removes local aggregations at all levels and only keeps the global one ($\mathbf{Z}^{L+1}$). This variant resembles SENet [29]. 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.

Component Analysis. Here we ablate FocalNet to study the relative contribution of each component. The result is reported in Table 11, 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.

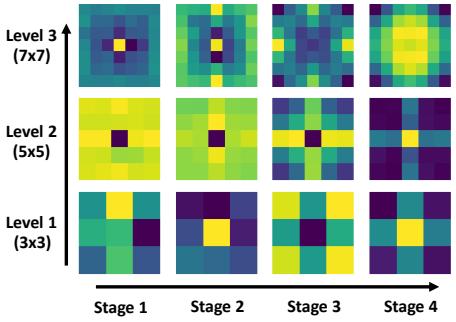

Fig. 3: Visualization of learned kernels at three levels and four stages in FocalNet-T (LRF). For clarity, we only show for the last layer of each stage.

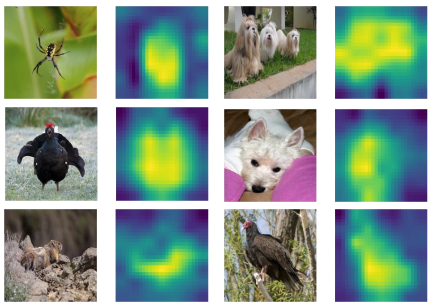

Fig. 4: Visualization of modulator values (corresponding to the right side of $\odot$ in Eq. (6)) at the last layer in FocalNet-T (LRF).The original modulator map is upsampled for display.

– **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.

Finally, 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 12, 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.

Model Interpretation We visualize what FocalNets have learned based on parameters and activations. In Fig. 3, we show the learned depth-wise convolution filters in FocalNet-T (LRF), with yellow color indicating higher magnitudes. We see that our model learns to focus more on local information at earlier stages, and more on global contexts at later stages. In Fig. 4, we visualize the extracted modulator $\mathbf{M}$ at the last layers for each image, and take the average over all channels. Interestingly, the generated modulators detect foreground and boundary regions for visual objects. In Fig. 5, we further visualize the gating maps learned in our FocalNet. It shows that for different image locations, our model indeed learns gathering the context from different focal levels adaptively.

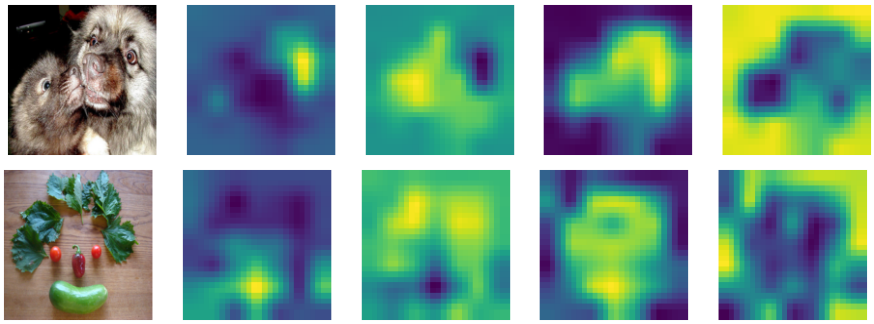

Fig. 5: Visualization of gating values $\mathbf{G}$ at last layer of our FocalNet-B (LRF) pretrained on ImageNet-1K. The columns from left to right are input images, gating maps at focal level 1,2,3 and global level. More visualizations in Appendix.

	Image Classification						Object Detection						Segmentation		
Model	Multi-scale				Monolithic		Mask R-CNN		C. Mask R-CNN				UperNet		
	Tiny	Small	Base	Large	Small	Base	Tiny 3×		Tiny 3×				Tiny	Small	Base
Metric	Top-1 Acc.				Top-1 Acc.		AP^b	AP^m	AP^b	AP^b_{50}	AP^b_{75}		mIoU		
ConvNeXt [42]	82.1	83.1	83.8	**86.6**	79.7	82.0	46.2	41.7	50.4	69.1	54.8		46.7	49.6	49.9
FocalNet (Ours)	**82.3**	**83.5**	**83.9**	86.5	**80.9**	**82.4**	**47.6**	**42.6**	**51.5**	**70.1**	**55.8**		**47.2**	**50.1**	**51.1**

Table 13: Comparison with ConvNeXts with compiled results on a range of computer vision tasks. The numbers of ConvNeXt are reported in [42].

4.4 Comparisons with ConvNeXt

In Sec. 2, we briefly discuss several concurrent works to ours. Among them, ConvNeXts [42] achieves new SoTA on some challenging vision tasks. Here, we quantitatively compare FocalNets with ConvNeXts by summarizing the results on a series of vision tasks in Table 13. FocalNets outperform ConvNeXt in most cases across the board. These numbers should be compared with cautions since they may use different model architectures and training settings.

5 Conclusion

We have presented *focal modulation*, a new mechanism that enables input-dependent token interactions for visual modeling. It consists of a hierarchical contextualization step to gather for each query token its contexts from local to global ranges at different levels of granularity, a gated aggregation step to adaptively aggregate context features into modulators based on the query content, followed by a simple modulation step. With *focal modulation*, we built a series of simple yet attention-free Focal Modulation Networks (FocalNets). Extensive experimental results 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 in real-world applications.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
4. Chang, S., Wang, P., Wang, F., Li, H., Feng, J.: Augmented transformer with adaptive graph for temporal action proposal generation. arXiv preprint arXiv:2103.16024 (2021)
5. Chen, C.F., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification (2021)
6. Chen, S., Xie, E., Ge, C., Liang, D., Luo, P.: CycleMLP: A mlp-like architecture for dense prediction. arXiv preprint arXiv:2107.10224 (2021)
7. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. arXiv preprint arXiv:2103.15436 (2021)
8. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11030–11039 (2020)
9. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34** (2021)
10. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting spatial attention design in vision transformers. arXiv preprint arXiv:2104.13840 (2021)
11. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 702–703 (2020)
12. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. arXiv preprint arXiv:2011.09094 (2020)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
14. Ding, X., Zhang, X., Zhou, Y., Han, J., Ding, G., Sun, J.: Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. arXiv preprint arXiv:2203.06717 (2022)
15. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652 (2021)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
17. Gao, P., Lu, J., Li, H., Mottaghi, R., Kembhavi, A.: Container: Context aggregation network. arXiv preprint arXiv:2106.01401 (2021)

18. Guo, J., Han, K., Wu, H., Xu, C., Tang, Y., Xu, C., Wang, Y.: Cmt: Convolutional neural networks meet vision transformers. arXiv preprint arXiv:2107.06263 (2021)
19. Guo, J., Tang, Y., Han, K., Chen, X., Wu, H., Xu, C., Xu, C., Wang, Y.: Hire-mlp: Vision mlp via hierarchical rearrangement. arXiv preprint arXiv:2108.13341 (2021)
20. Guo, M.H., Lu, C.Z., Liu, Z.N., Cheng, M.M., Hu, S.M.: Visual attention network. arXiv preprint arXiv:2202.09741 (2022)
21. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al.: A survey on visual transformer. arXiv preprint arXiv:2012.12556 (2020)
22. Han, Q., Fan, Z., Dai, Q., Sun, L., Cheng, M.M., Liu, J., Wang, J.: Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. arXiv preprint arXiv:2106.04263 (2021)
23. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
25. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
26. Hou, Q., Jiang, Z., Yuan, L., Cheng, M.M., Yan, S., Feng, J.: Vision permutator: A permutable mlp-like architecture for visual recognition. arXiv preprint arXiv:2106.12368 (2021)
27. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
28. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3464–3473 (2019)
29. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
30. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European conference on computer vision. pp. 646–661. Springer (2016)
31. Jiang, Z.H., Hou, Q., Yuan, L., Zhou, D., Shi, Y., Jin, X., Wang, A., Feng, J.: All tokens matter: Token labeling for training better vision transformers. Advances in Neural Information Processing Systems **34** (2021)
32. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. arXiv preprint arXiv:2101.01169 (2021)
33. Li, B., Zheng, C., Giancola, S., Ghanem, B.: Sctn: Sparse convolution-transformer network for scene flow estimation. arXiv preprint arXiv:2105.04447 (2021)
34. Li, D., Hu, J., Wang, C., Li, X., She, Q., Zhu, L., Zhang, T., Chen, Q.: Involution: Inverting the inherence of convolution for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12321–12330 (2021)
35. Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., Li, W.: Trear: Transformer-based rgb-d egocentric action recognition. arXiv preprint arXiv:2101.03904 (2021)
36. Li, Y., Zhang, K., Cao, J., Timofte, R., Van Gool, L.: Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021)
37. Lian, D., Yu, Z., Sun, X., Gao, S.: As-mlp: An axial shifted mlp architecture for vision. arXiv preprint arXiv:2107.08391 (2021)

38. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
39. Liu, H., Dai, Z., So, D., Le, Q.: Pay attention to mlps. Advances in Neural Information Processing Systems **34** (2021)
40. Liu, H., Dai, Z., So, D.R., Le, Q.V.: Pay attention to MLPs. arXiv preprint arXiv:2105.08050 (2021)
41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
42. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
43. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
44. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
45. Lou, Y., Xue, F., Zheng, Z., You, Y.: Sparse-mlp: A fully-mlp architecture with conditional computation. arXiv preprint arXiv:2109.02008 (2021)
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
47. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. arXiv preprint arXiv:2011.12450 (2020)
48. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
49. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
50. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)
51. Tang, C., Zhao, Y., Wang, G., Luo, C., Xie, W., Zeng, W.: Sparse mlp for image recognition: Is self-attention really necessary? arXiv preprint arXiv:2109.05422 (2021)
52. Tang, Y., Han, K., Guo, J., Xu, C., Li, Y., Xu, C., Wang, Y.: An image patch is a wave: Phase-aware vision mlp. arXiv preprint arXiv:2111.12294 (2021)
53. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., et al.: MLP-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601 (2021)
54. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al.: Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems **34** (2021)
55. Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Joulin, A., Synnaeve, G., Verbeek, J., Jégou, H.: ResMLP: Feedforward networks for image classification with data-efficient training. arXiv preprint arXiv:2105.03404 (2021)
56. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877 (2020)

57. Touvron, H., Cord, M., El-Nouby, A., Bojanowski, P., Joulin, A., Synnaeve, G., Jégou, H.: Augmenting convolutional networks with attention-based aggregation. arXiv preprint arXiv:2112.13692 (2021)
58. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 32–42 (2021)
59. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12894–12904 (2021)
60. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
61. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. arXiv preprint arXiv:2012.00759 (2020)
62. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. arXiv preprint arXiv:2103.11681 (2021)
63. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122 (2021)
64. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
65. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. arXiv preprint arXiv:2011.14503 (2020)
66. Wightman, R., Touvron, H., Jégou, H.: Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476 (2021)
67. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808 (2021)
68. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018)
69. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
70. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9981–9990 (2021)
71. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641 (2021)
72. Yang, J., Ren, Z., Gan, C., Zhu, H., Parikh, D.: Cross-channel communication networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. pp. 1297–1306 (2019)
73. Yu, T., Li, X., Cai, Y., Sun, M., Li, P.: S^2-MLPv2: Improved spatial-shift mlp architecture for vision. arXiv preprint arXiv:2108.01072 (2021)
74. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. arXiv preprint arXiv:2111.11418 (2021)
75. Yuan, L., Hou, Q., Jiang, Z., Feng, J., Yan, S.: Volo: Vision outlooker for visual recognition. arXiv preprint arXiv:2106.13112 (2021)

76. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
77. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
78. Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., Gao, J.: Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. arXiv preprint arXiv:2103.15358 (2021)
79. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: 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. pp. 9759–9768 (2020)
80. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
81. Zhao, J., Li, X., Liu, C., Bing, S., Chen, H., Snoek, C.G., Tighe, J.: Tuber: Tube-transformer for action detection. arXiv preprint arXiv:2104.00969 (2021)
82. Zheng, H., He, P., Chen, W., Zhou, M.: Mixing and shifting: Exploiting global and local dependencies in vision mlps. arXiv preprint arXiv:2202.06510 (2022)
83. Zheng, M., Gao, P., Wang, X., Li, H., Dong, H.: End-to-end object detection with adaptive clustering transformer. arXiv preprint arXiv:2011.09315 (2020)
84. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13001–13008 (2020)
85. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
86. Zhou, D., Shi, Y., Kang, B., Yu, W., Jiang, Z., Li, Y., Jin, X., Hou, Q., Feng, J.: Refiner: Refining self-attention for vision transformers. arXiv preprint arXiv:2106.03714 (2021)
87. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

A More Implementation Details

Training settings for ImageNet-1K. We follow Swin [41] to use the same set of data augmentations including Random Augmentation [11], Mixup [77], CutMix [76] and Random Erasing [84]. For model regularization, we use Label Smoothing [49] and DropPath [30]. For all models, the initial learning rate is set to 10^{-3} after 20 warm-up epochs beginning with 10^{-6}. For optimization, we use AdamW [44] and a cosine learning rate scheduler [43]. The weight decay and the gradient clipping norm is set to 0.05 and 5.0, respectively. We set the stochastic depth drop rates to 0.2, 0.3 and 0.5 for our tiny, small and base models, respectively. During training, images are randomly cropped to 224×224, and a center crop is used during evaluation. Throughput/Speed is measured on one V100 GPU with batch size 128, following [41].

Training settings for ImageNet-22K. We train FocalNet-B and FocalNet-L for 90 epochs with a batch size of 4096 and input resolution 224×224. The initial learning rate is set to 10^{-3} after a warmup of 5 epochs. We set the the stochastic depth drop rates to 0.2 for both networks. For stability, we use LayerScale [58] with initial value 10^{-4} for all layers. The other settings follow those for ImageNet-1K. After the pretraining, we finetune the models on ImageNet-1K for 30 epochs with initial learning rate of 3×10^{-5}, cosine learning rate scheduler and AdamW optimizer. The stochastic depth drop rate is set to 0.3 and both CutMix and Mixup are muted during the finetuning.

B Downstream Tasks

B.1 Object Detection

Effect of kernel size. We study how the various kernel sizes affect the object detection performance when finetuning FocalNet-T (LRF) with $k^{\ell=1} = 3$ pretrained on ImageNet-1K. In Fig. 6, we vary the kernel size at first level $k^{\ell=1}$ from

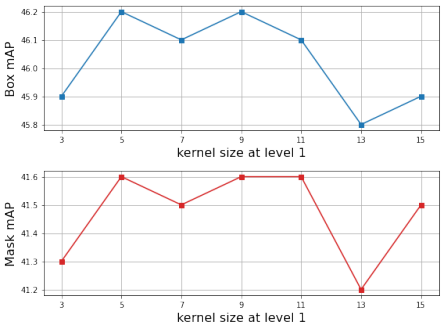

Fig. 6: Box and mask mAP for Mask R-CNN 1× training. We use FocalNet-T (LRF) as the baseline model and vary its kernel size at first level $k^{\ell=1} \in \{3, 5, 7, 9, 11, 13, 15\}$.

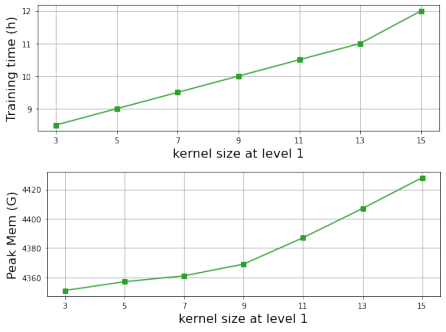

Fig. 7: Training time (wall-clock) and peak memory for Mask R-CNN 1×. We train Focalnet-T (LRF) with different kernel sizes on 16 V100 GPUs with batch size 16.

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}
FocalNet-T (SRF)	48.6	267	45.9	68.3	50.1	41.3	65.0	44.3	47.6	69.5	52.0	42.6	66.5	45.6
FocalNet-T (LRF)	48.9	268	46.1	68.2	50.6	41.5	65.1	44.5	48.0	69.7	53.0	42.9	66.5	46.1
FocalNet-T (SRF)†	45.8	261	46.8	69.1	51.2	41.9	65.6	44.6	48.5	70.0	53.2	43.3	67.0	46.3
FocalNet-S (SRF)	70.8	356	48.0	69.9	52.7	42.7	66.7	45.7	48.9	70.1	53.7	43.6	67.1	47.1
FocalNet-S (LRF)	72.3	365	48.3	70.5	53.1	43.1	67.4	46.2	49.3	70.7	54.2	43.8	67.9	47.4
FocalNet-S (SRF)†	59.5	312	48.1	70.5	52.8	43.1	67.2	46.2	49.2	70.6	53.9	43.8	67.6	47.2
FocalNet-B (SRF)	109.4	496	48.8	70.7	53.5	43.3	67.5	46.5	49.6	70.6	54.1	44.1	68.0	47.2
FocalNet-B (LRF)	111.4	507	49.0	70.9	53.9	43.5	67.9	46.7	49.8	70.9	54.6	44.1	68.2	47.2
FocalNet-B (SRF)†	107.1	481	49.6	71.2	54.6	44.0	68.2	47.6	50.2	71.0	55.0	44.3	68.1	47.9

Table 14: Gray rows are additional results using deeper but thinner FocalNets in Table 4 as the backbone for Mask R-CNN.

3 to 15 for object detection finetuning. We have two interesting observations: (*i*) though the pretrained model used $k^{\ell=1} = 3$, it can be finetuned with different kernel sizes to adapt high-resolution object detection task; (*ii*) a moderate kernel size (5,7,9,11) have a slightly better performance than a kernel size which is too small (3) or too big (13,15), probably because small kernel cannot capture the long-range dependency while big kernel misses the detailed local context. In Fig. 7, we further show the corresponding wall-clock time cost and peak memory when training on 16 V100 GPUs with batch size 16. Accordingly, increasing the kernel size gradually increases the training memory and time cost. For a good performance/cost trade-off, we therefore set $k^{\ell=1} = 9$ for all the object detection finetuning experiments in our main submission.

Results with deeper and thinner FocalNets. In our main submission, we compared with previous SoTA methods Swin and Focal Transformer in a restricted way by using the same network depth layout. Meanwhile, we also showed that different depth layouts lead to different image classification performance. Here, we investigate how the layout affects the object detection performance. We use the deeper but thinner FocalNets in Table 4 of our main submission as the backbones. Specifically, 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, we reduce the initial hidden dimension from 96, 128 to 64, 96, respectively. In Table 14, we add the additional gray rows to compare with the results reported in our main submission. In Table 15, we further show the $1\times$ results of deeper and thinner FocalNets with large receptive field. Accordingly, the object detection performance (both box and mask mAP) are boosted over the shallower and wider version of FocalNets with same receptive field. On one hand, this trend suggests a feasible way to improve the performance for our FocalNet, and further demonstrate its effectiveness for both image classification and object detection. On the other hand, it suggests that keeping network configuration (depth, hidden dimension, *etc.*) the same is important for a fair comparison with previous works.

B.2 Semantic Segmentation

In Table 16, we report the results using the deeper and thinner FocalNets as the backbone for semantic segmentation. As we can see, for FocalNet-T, increasing

Backbone	#Param.	FLOPs	AP^b	AP^m
Swin-Tiny	47.8	264	43.7	39.8
FocalAtt-Tiny	48.8	291	44.8	41.0
FocalNet-T (SRF)	48.6	267	45.9	41.3
FocalNet-T (SRF)†	45.8	261	46.8	41.9
FocalNet-T (LRF)	48.9	268	46.1	41.5
FocalNet-T (LRF)†	46.1	262	46.7	41.9
Swin-Small	69.1	354	46.5	42.1
FocalAtt-Small	71.2	401	47.4	42.8
FocalNet-S (SRF)	70.8	356	48.0	42.7
FocalNet-S (SRF)†	59.5	312	48.1	43.1
FocalNet-S (LRF)	72.3	365	48.3	43.1
FocalNet-S (LRF)†	60.0	315	48.6	43.3
Swin-Base	107.1	497	46.9	42.3
FocalAtt-Base	110.0	533	47.8	43.3
FocalNet-B (SRF)	109.4	496	48.8	43.3
FocalNet-B (SRF)†	107.1	481	49.6	44.0
FocalNet-B (LRF)	111.4	507	49.0	43.5
FocalNet-B (LRF)†	107.9	485	49.9	44.2

Table 15: Additional results of Mask R-CNN 1× with deeper and thinner FocalNets (LRF) in gray rows. We use the same pretrained model as FocalNet (SRF)†, but add an extra focal level on top with kernel initialized with all-zeros.

Backbone	#Param.	FLOPs	mIoU	+MS
ResNet-101 [24]	86	1029	44.9	-
Twins-SVT-L [10]	133	-	48.8	50.2
Swin-T [41]	60	941	44.5	45.8
FocalAtt-T [71]	62	998	45.8	47.0
FocalNet-T (SRF)	61	944	46.5	47.2
FocalNet-T (LRF)	61	949	46.8	47.8
FocalNet-T (SRF)†	55	934	47.4	48.5
Swin-S [41]	81	1038	47.6	49.5
FocalAtt-S [71]	85	1130	48.0	50.0
FocalNet-S (SRF)	83	1035	49.3	50.1
FocalNet-S (LRF)	84	1044	49.1	50.1
FocalNet-S (SRF)†	69	986	49.4	50.3
Swin-B [41]	121	1188	48.1	49.7
FocalAtt-B [71]	126	1354	49.0	50.5
FocalNet-B (SRF)	124	1180	50.2	51.1
FocalNet-B (LRF)	126	1192	50.5	51.4
FocalNet-B (SRF)†	117	1159	51.0	51.9

Table 16: Semantic segmentation on ADE20K [85]. All models are trained with UperNet [68]. Grays rows are additional results with deeper yet thinner FocalNets (SRF).

the depth does not bring extra improvement. For larger models, however, a deeper version outperforms the shallow ones, particularly on FocalNet-B.

C Additional Model Interpretation

Our focal modulation consists of three main components: (*i*) convolution for contextualization; (*ii*) gating mechanism for aggregation of multiple granularity and (*iii*) linear projection for generator modulator. In our main submission, we have shown that the kernels learned in our model tend to look at small local regions at earlier stages while larger regions at later stages. The modulator $\mathbf{M}$ computed at the last layer of our FocalNet shows an emerge of object localization capacity even though no class guidance is provided. As shown in Table 10 of our main submission, the gating mechanism $\mathbf{G}$ in our focal modulation also plays an important role to selectively aggregate the contexts from different granularity levels. Here, we visualize the gating values at the last layer of FocalNet to investigate where our model focus on at different focal levels, when modulating each visual token for different images.

We use FocalNet-B (LRF) which exploits three levels of granularity ($\mathbf{G}^{\ell=1,2,3}$) and additional global context $\mathbf{G}^{\ell=4}$. Hence, we have four gating maps in total at the last layer for each input image. On a set of randomly selected ImageNet-1K validation images, we show the gating maps in Fig. 8, 9 and 10. Surprisingly, we found an intriguing property: the gating values learned by our FocalNet consistently show an intuitive pattern. For the visual tokens at object regions ($\ell = 1$), their gating values are much higher than those outside object regions at first level. When looking more closely, we can see that the predicted gating

values mainly lie on the most complicated textures within object regions. At the second level $\ell = 2$, the gating values are still higher in object regions but the peak values usually move to the object boundaries instead. At the third level $\ell = 3$, the whole object regions have higher gating values than background regions. Finally at level $\ell = 4$, we find there is a clear distinction between foreground and background regions when aggregating the global contexts. The foreground regions usually show less interest in the global context and the other way around for the background regions. Even for those images containing multiple foreground objects, our model still shows coherent patterns. Comparing the gating values for first three levels and the last global context, we can find our model does gather more information from local regions when modulating foreground visual tokens and more global context for background tokens. This aligns well with our intuitions discussed in our main submission. As a side product, the predicted gating values also provide meaningful cues for object localization and even segmentation, which we leave for future exploration.

D Limitation and Social Impact

Limitations. In this work, we have demonstrated focal modulation is an effective yet efficient way for visual modeling. The main goal of this work is to develop a new way for visual token interaction. Though it seems straightforward, a more comprehensive study is needed to verify whether the proposed focal modulation networks can be applied to other domains such as pure NLP tasks. Moreover, when coping with multi-modality tasks, SA can be feasibly transformed to cross-attention by alternating the queries and keys. The proposed focal modulation requires the number of gathered contexts same to that of queries so that an element-wise multiplication can be conducted for modulation. Hence, how to perform the so-called cross-modulation needs more exploration.

Social Impact. This work is mainly focused on architecture design for computer vision tasks. We have trained the models on various datasets and tasks. One concern is that it might be biased to the training data. When it is trained on large-scale webly-crawled image data, the negative impact might be amplified due to the potential offensive or biased contents in the data. To avoid this, we need to have a careful sanity check on the training data and the model's predictions before training the model and deploying it to the realistic applications.

Fig. 8: Visualization of gating values $\mathbf{G}$ at last layer of our FocalNet-B (LRF) pretrained on ImageNet-1K. From left to right, we show input image, and gating weights $\mathbf{G}^\ell, \ell = 1, 2, 3, 4$.

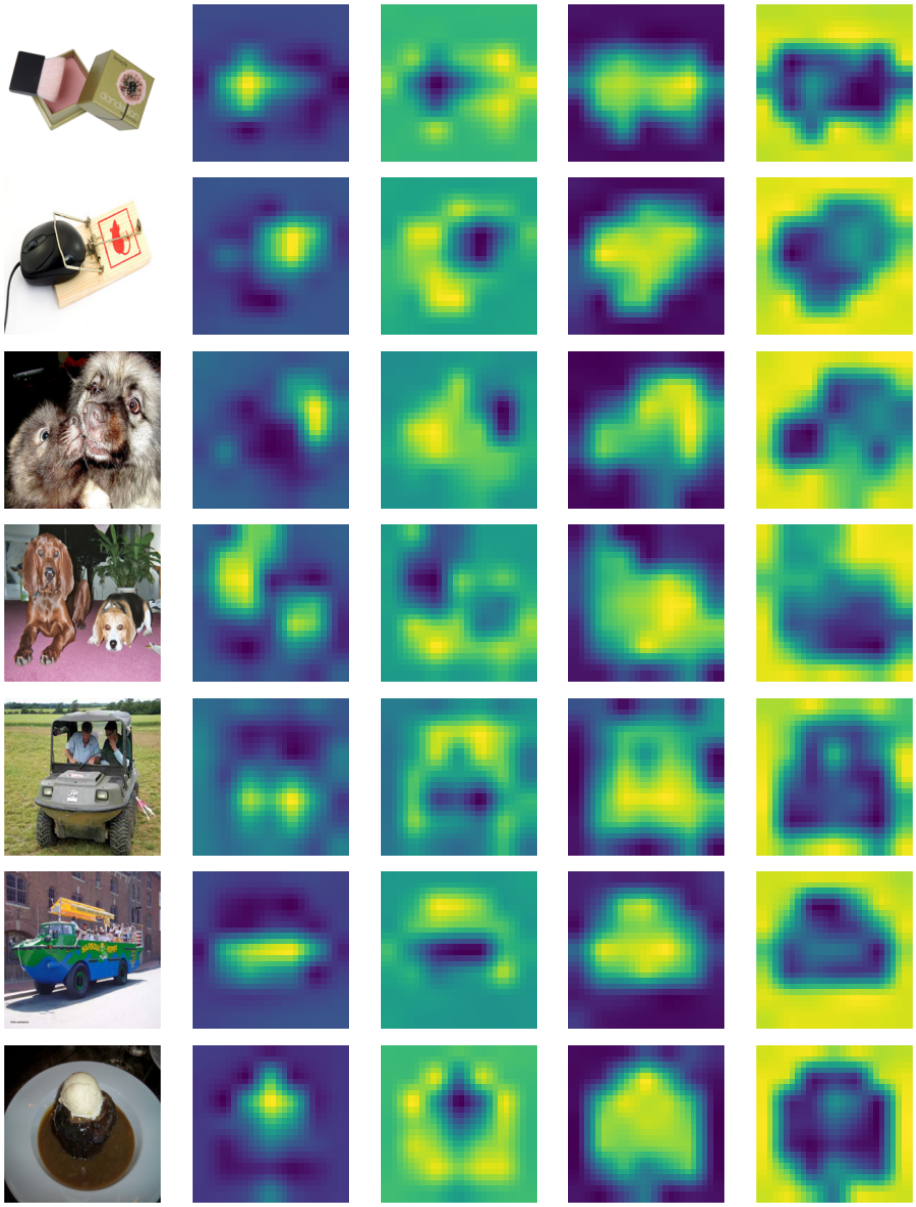

Fig. 9: Visualization of gating values **G** at last layer of our FocalNet-B (LRF) pretrained on ImageNet-1K. The order from left to right column is same to Fig. 8

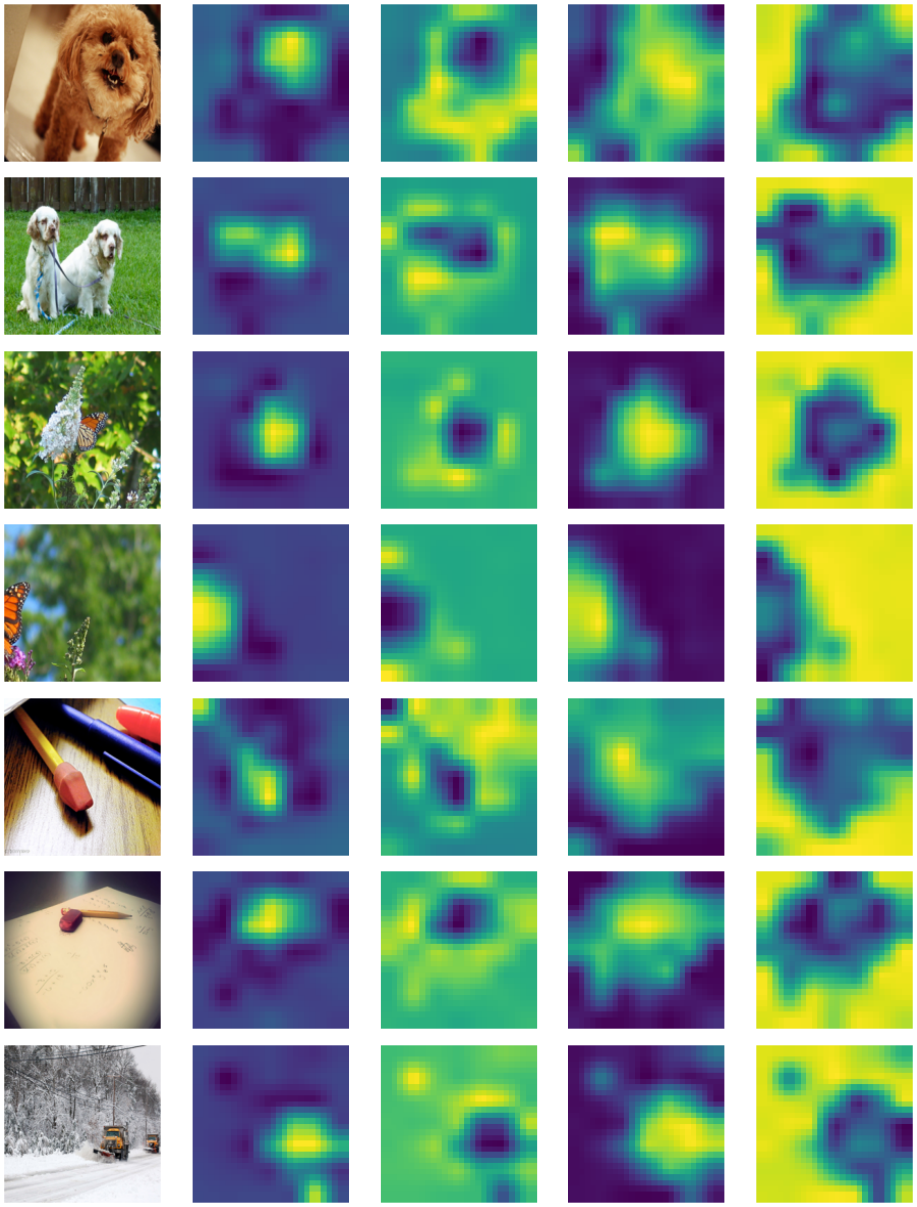

Fig. 10: Visualization of gating values **G** at last layer of our FocalNet-B (LRF) pretrained on ImageNet-1K. The order from left to right column is same to Fig. 8