# LA-layer: General local attention layer for full attention networks

Hui Lu[1*], Ronald Poppe[1], Albert Ali Salah[1,2]
[1] Department of Information and Computing Sciences, Utrecht University, Utrecht
[2] Department of Computer Engineering, Bogazici University, Istanbul, Turkey
{h.lu1, R.W.Poppe, a.a.salah}@uu.nl

***Abstract*—Attention layers have contributed to state-of-the-art results on vision tasks[1]. Still, they leave room for improvement because position information is used in a fixed manner, and the computation cost is typically high. To mitigate both issues, we propose a convolution-style local attention layer (LA-layer) as a replacement for traditional attention layers. LA-layers not only encode the position information of pixels in a convolutional manner, but also produce position offsets following a novel constrained rule so that keys will deform and result in larger receptive fields. Query and keys are processed by a novel aggregation function that outputs attention weights for the values. In our experiments with different types of ResNets, we replace convolutional layers with LA-layers and address image recognition, object detection and instance segmentation tasks. We consistently demonstrate performance gains, despite having fewer FLOPs and training parameters. Our code is available at: https://github.com/hotfinda/LA-layer.**

***Index Terms*—Local attention, CNN, Deformable Kernel**

## I. Introduction

Convolutional neural networks (CNNs) are the backbone for many computer vision tasks [1] [2]. One challenge with CNNs is to model long-range interactions between pixels to obtain large receptive fields. To overcome this issue, recent work introduces self-attention (SA) [3], [4] to capture long-range interactions of pixels at different position. Building on SA layers, vision transformer models (e.g., [5] [6] [7]) typically outperform CNNs in terms of accuracy by integrating transformer-style modules into CNN-based architectures.

Despite the success of SA layers, two issues remain. First, the self-attention layer is the content-based summarization of features, and lacks of position information of pixels in feature maps, which will cause decreased performance. The second issue is that most transformers neglect the performance improvement from the extraction of useful tokens, while focusing on enlarging the receptive field to achieve long-range dependencies. This leads to excessive memory use and computational cost for the attention layer.

To address these issues, researchers have proposed local attention layers to reduce the computation cost of SA layers. They also proposed to encode the position information of pixels in the feature map through manually designed encoders [5], [8] or convolution projection operations [9]. Both approaches result in additional computation cost. Moreover, the receptive field of local attention layers is constrained by the kernel size, and uninformative features are not filtered out.

Another solution is to generate position offsets from the feature map such that the local kernel becomes deformable. In this way, the network can learn to extract useful features and filter out irrelevant inputs. Moreover, the receptive field is enlarged. For example, inspired by the deformable convolution [10], the DAT model uses a deformable self-attention module [11]. However, the current deformable process is unconstrained, so the network tends to focus on a limited number of, potentially remote, informative regions, possibly missing out on complementary information. Also, the high computation and memory cost of attention layers limit the applicability of the deformable kernel.

In this work, we propose *LA-layer*, a novel convolution-style local attention layer. In the LA-layer, we first extract the single query within a kernel region-like convolution operation. Keys are added with position offsets adjusted through a constraint rule. This will expand the receptive field while avoiding a too narrow focus on a small number of regions. The query and keys pass through a novel attention aggregation function to produce the attention weights with lower computation and memory cost. The attention weights are used to aggregate all input values, and the final output is generated when the kernel slides over the whole feature map.

By integrating the LA-layer into common CNN models, we obtain full attention models of the same architecture with fewer parameters and FLOPs. Our two main contributions:

- We introduce LA-layer, a deformable local attention layer that can serve as an efficient and effective replacement for convolution layers. The LA-layer kernel deforms based on the proposed constraint rule to avoid focusing on specific regions, and the output is generated through an efficient novel attention aggregation function to highly reduce the computation cost.
- We use ResNets with LA-layers as full attention networks. Extensive experiments on image classification, object detection and instance segmentation demonstrate that LA-layers outperform popular backbones.

We first discuss related work on convolution and self-attention layers. We then introduce the LA-layer in Section III. In Section IV, we demonstrate the performance of the LA-layer on various core vision tasks. We conclude in Section V.

[1]"This is the uncorrected author proof, copyright with IEEE. Please cite as: "Lu, H., R. Poppe, A.A. Salah, "LA-layer: General local attention layer for full attention networks," IEEE Int. Conf. on Multimedia & Expo (ICME), Brisbane, 2023."

## II. Related work

**Convolution Layers and Extensions**: The convolution layer is the basic building block for CNNs, and the size of the receptive field is a key factor that highly affects the performance of the convolution layer [12]. Although larger kernels of convolution layers can result in larger receptive fields, this generally requires a carefully designed network architecture [13]. Otherwise, large kernels in convolution layers tend to harm the performance due to the lack of a locality constraint [14].

A common way to increase the receptive field of the convolution layer without using large kernels is to model the interactions between spatially distant regions with a limited receptive field through successive convolution layers [1]. While typically increasing the accuracy, this approach reduces the efficiency of CNNs. To overcome the short-range problem for small kernels, several extensions to the regular convolution layer have been proposed. Group convolutions [15] and depth-wise convolutions [16] are examples of such efforts. Another option is to modify the spatial scope for aggregation, which can enlarge the receptive field. One popular implementation of this idea is the dilated convolution [17], which increases the spacing inside the kernel to increase the receptive field of the convolution layer.

**Self-Attention and Extensions**: Researchers have begun to apply self-attention (SA) to computer vision tasks [3], [4]. This is achieved by employing SA over feature vectors across different spatial locations within an image, such that a larger content-based receptive field is obtained. Inspired by advantages of SA, researchers have further proposed full attention networks which replace the convolution layers inside CNN models with attention layers [8]. Full attention networks based on SA layers have achieved state-of-the-art results for various computer vision tasks [18]. Despite their popularity, global SA layers in such networks incur high computation and memory demands due to the complicated aggregation method of SA. To address this issue, researchers have introduced local attention layers to constrain the attention pattern fixed local windows (e.g., [8], [19]).

Although the local attention layer reduces the computation cost, most research focuses on the content-based summarization of features. They propose to encode the position information through manually designed encoders [5], [8] or convolution projection operations [9], which results in additional computation costs. Also, the local attention layer calculates the interaction of all pixels in the local window, which will include the irrelevant features. This will limit the performance of networks. Also, the receptive field of the local attention layer is constrained by the local window or kernel size.

An intuitive method to solve these issues is to generate the position offsets from the input feature map. Based on these offsets, the kernel becomes deformable, such that only useful features will be extracted and irrelevant features in the original kernel region will be ignored. Moreover, the receptive field will be naturally enlarged. Although a deformable kernel is realized in Deformable Convolution Networks (DCN, [10]), applying DCN to attention layers is a non-trivial problem. Compared to the convolution layer, the space complexity of the SA layer is generally bi-quadratic. Deformable DETR [20] implements the idea of deformable attention with a lower number of keys at each scale to reduce the computation cost. This works well as a detection head, but also causes a loss of information due to the strongly reduced number of keys. Based on the assumption that different queries may have similar attention maps, DAT [11] generates a few groups of position offsets for the local attention layers, such that they can share shifted keys and values for each query to achieve an efficient trade-off. However, since different pixels in the same group use a shared query offset, the network concentrates on specific regions, which limits the receptive field.

Instead of using convolution projection operations or position encoders [8], [19], our proposed LA-layer is a local convolution-style self-attention layer. The position information is implicitly encoded in a convolutional manner, inspired by implicit encoding of position information along with the feature maps that the convolution operation produces [21]. Compared with traditional works, our LA-layer uses deformable kernels based on the proposed constraint rule to focus the attention locally and obtain rich inforamtion for the model. Besides, we propose a novel aggregation method for LA-layers, which can highly reduce the computation and memory cost.

## III. Local Attention Layer (LA-layer)

We introduce a local attention layer (LA-layer) as a replacement for convolution layers in CNNs, to produce full attention networks. The framework of LA-layer is shown in Figure 1. We will introduce the LA-layer, followed by the implementation of full attention networks using the LA-layer.

### A. Overview of the LA-Layer

For the LA-layer, similar to the kernel in the convolution layer, we assign a local region, i.e., a $k \times k$ neighborhood, to aggregate the input features. We first revisit the local attention layer in recent vision transformers [8]. Taking a flattened feature map $x \in R^{W \times H \times C}$ as the input, the Query map $Q$, Key map $K$, and Value map $V$ are generated through the self-attention mechanism, which is formulated as:

$$Q = xW_q \tag{1}$$

$$K = xW_k \tag{2}$$

$$V = xW_v \tag{3}$$

where $W_q, W_k, W_v \in R^{d_{in} \times d_{out}}$ are learned transformation matrices. Local Query map $Q_k$, Key map $K_k$, and Value map $V_k$ are extracted from the feature maps $K$, $Q$, and $V$. At each anchor position $ij$, the attention matrix is produced by multiplying $q_{ij}$ and $K_k$. Content-based aggregated feature $y_{ij}$ is generated after $V_k$ and the attention matrix pass through the feature multiplication. Finally, we obtain the attention feature

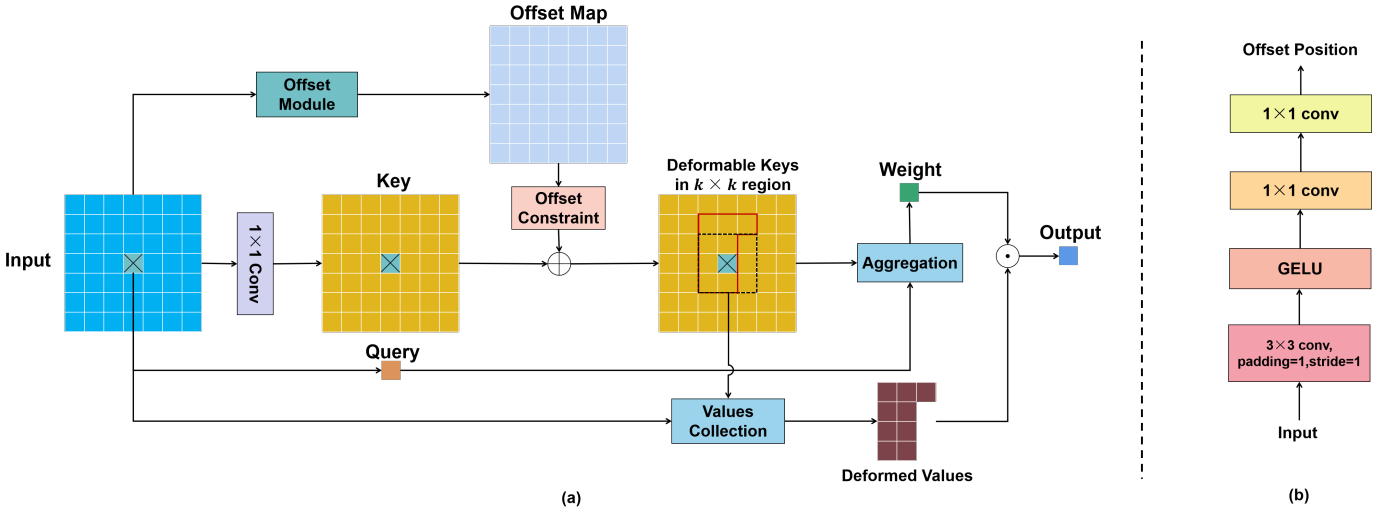


Fig. 1: **Schematic overview of the LA-layer**. (a) Computations repeated for each element in the input feature map. The output is the sum of deformable features under the constraint rule. (b) Offset module. The input feature map passes through a $3 \times 3$ convolution, GELU, and two $1 \times 1$ convolutions. The output is an $H \times W \times 2$ offset feature map. Values are the deformed position indices.

map by sliding the kernel through the entire input. For input $x_{ij}$ with corresponding position $ij$, output $y_{ij}$ can be computed as:

$$y_{ij} = \sum_{a,b \in N_k(i,j)} softmax_{ab}(q_{ij} k_{ab}^T) v_{ab} \tag{4}$$

where $q_{ij} = x_{ij} W_q$, $k_{ab} = x_{ab} W_k$, $v_{ab} = x_{ab} W_v$, with $a, b \in N_k(i,j)$ the pixels at positions $ab$ with spatial extent $k$ centered at $x_{ij}$.

Multiple attention heads are used to learn multiple distinct representations of the input [3]. This works by partitioning the pixel features $x_{ij}$ into $m$ groups $x_{ij}^n \in R^{d_{in}/m}$ in the depth channel and then calculate each group independently using distinct transformations. Finally, we concatenate the resulting representations in the depth channel and generate the final output.

To avoid the problems of the traditional local attention layer, a small network is used for the offset generation following [10]. It processes the input feature map and outputs the offset values for reference points in the kernel region. The input features are firstly passed through a $3 \times 3$ convolution to capture local features. Then, Gaussian Error Linear Unit (GELU) activation and two $1 \times 1$ convolutions are applied to produce the 2D offsets.

Although the learned position offsets can enlarge the receptive field, we notice that the offset values tend to concentrate on specific regions and neglect regions include less important features. This reduces the quantity of relevant features and limits the performance of the network. To solve this issue, we propose a simple but effective constraint rule during the deformation process. We first round up the 2D offsets $\Delta p$, and then assign the size of constraint region $l$, which is used to constrain the position offsets as:

$$\Delta p' = \frac{\Delta p}{|max(\Delta p)| + |min(\Delta p)|} l \tag{5}$$

where $l$ is typically in the order of 1–2 times the kernel size $k$, and $\Delta p'$ is the constrained offset position.

With Eq. 5, the learned offsets are limited by constraint region size $l$. When the offset moves toward specific regions outside $l$ (the value of $\Delta p$ is bigger than $l$), it will be scaled and move towards less specific regions inside $l$ during the training process. Since the kernel will slide over the whole feature map, all useful features will be included during the process, and features belong to the specific regions will also be collected.

After producing the offsets for a target position $t$ in local attention region, we propose a novel aggregation function to effectively aggregate the Key map and Query map, the result of which is combined with the Value map through multiplication. In this way, we could highly reduce the computation cost of the LA-layer.

For the local Query map $Q_k$, Key map $K_k$, and Value map $V_k$ extracted from feature maps $K$, $Q$, and $V$, our aggregation approach can be performed by using the following mathematical operation:

$$y_t = \sum \left\{ (v_{t'} + \Delta p_{t'}^{'}) \bigodot \frac{\sum \left[ (k_{t'} + \Delta p_{t'}^{'}) \bigodot q_t \right]}{\sum (k_{t'} + \Delta p_{t'}^{'})} \right\} \tag{6}$$

where $\bigodot$ represents the element-wise multiplication, $t'$ is the reference index in the kernel region, $y_t$ is the output of the LA-layer at position $t$, $k_{t'}$ is the key inside Key map $K_k$, $v_{t'}$ is the value inside Value map $V_k$. See Figure 1 for an illustration.

In Eq. 6, the term that is element-wise multiplied with $(v_{t'} + \Delta p_{t'}^{'})$ can be seen as a weighted average of attention value or the pixel similarity between $q_t$ and the pixels $k_{t'}$ for $v_{t'}$. Consequently, we can build up the content relationships among different locations while effectively avoiding expensive computations and storage of the attention matrix.

TABLE I: **ImageNet-1K image classification** on ResNet and ResNext backbones. Comparison with state-of-the-art local attention approaches. Differences with the original ResNet/ResNext models is shown in parentheses. Best results in **bold**.

| | Model | FLOPs (G) | Params (M) | Top-1 Acc.(%) | Top-5 Acc.(%) |
|---|---|---|---|---|---|
| Backbones | ResNet-50 [22] | 4.1 | 25.5 | 77.3 | 93.6 |
| | ResNet-101 | 7.9 | 44.6 | 78.5 | 94.2 |
| | ResNeXt-50 [23] | 4.2 | 25.0 | 78.2 | 93.9 |
| | ResNeXt-101 | 8.0 | 44.2 | 79.1 | 94.4 |
| Local attention | Stand-Alone-50 [8] | 3.6 | 18.0 | 77.6 | - |
| | Swin-T-ResNet-50 [5] | 4.6 | 30.7 | 78.4 | 94.0 |
| | DAT-T-ResNet-50 [11] | 4.6 | 30.7 | 78.9 | 94.5 |
| | LR-Net-50 [19] | 4.3 | 23.3 | 77.3 | 93.6 |
| | LR-Net-101 | 8.0 | 42.0 | 78.5 | 94.3 |
| | AA-ResNet-50 [25] | 4.2 | 25.8 | 77.7 | 93.8 |
| | AA-ResNet-101 | 8.1 | 45.4 | 78.7 | 94.4 |
| | CoTNet-50 [18] | 3.3 | 22.2 | 79.2 | 94.5 |
| | CoTNet-101 | 6.1 | 38.3 | 80.0 | 94.9 |
| | CoTNeXt-50 | 4.3 | 30.1 | 79.5 | 94.5 |
| | CoTNeXt-101 | 8.2 | 53.4 | 80.3 | 95.0 |
| LA-layer | LA-ResNet-50 (**ours**) | 3.4 | 21.0 | 79.7 (+2.4) | 94.9 (+1.3) |
| | LA-ResNet-101 (**ours**) | 5.7 | 35.9 | 81.0 (+2.5) | 95.7 (+1.5) |
| | LA-ResNeXt-50 (**ours**) | 4.2 | 24.2 | 80.6 (+2.4) | 95.4 (+1.5) |
| | LA-ResNeXt-101 (**ours**) | 7.6 | 39.5 | **81.5** (+2.4) | **95.8** (+1.4) |

TABLE II: **Object detection results** on MS COCO validation set, with Fast R-CNN and Cascade R-CNN as detection heads. For our models with LA-layer, the difference with the original ResNet/ResNext models is shown in parentheses. Best results for each detection head in **bold**.

| Method | Model | FLOPs | AP | $AP_{50}$ | $AP_{75}$ |
|---|---|---|---|---|---|
| Faster R-CNN | ResNet-50 [22] | 180G | 39.34 | 59.47 | 42.76 |
| | ResNet-101 | 246G | 41.46 | 61.99 | 45.38 |
| | ResNeXt-50 [23] | 279G | 41.31 | 62.23 | 44.91 |
| | ResNeXt-101 | 406G | 42.91 | 63.77 | 46.89 |
| | ResNeSt-50 [24] | 291G | 42.39 | 63.73 | 46.02 |
| | ResNeSt-101 | 422G | 44.13 | 61.91 | 47.67 |
| | LA-ResNet-50 (**ours**) | 164G | 44.28 (+4.94) | 65.61 (+6.14) | 47.98 (+5.22) |
| | LA-ResNet-101 (**ours**) | 215G | 46.63 (+5.19) | 68.07(+6.08) | 50.41(+5.03) |
| | LA-ResNeXt-50 (**ours**) | 274G | 44.60 (+3.29) | 66.29(+4.06) | 48.10(+3.19) |
| | LA-ResNeXt-101 (**ours**) | 384G | **46.71** (+3.80) | **68.12**(+4.35) | **50.75**(+3.86) |
| Cascade R-CNN | ResNet-50 [22] | 201G | 42.45 | 59.76 | 46.09 |
| | ResNet-101 | 274G | 44.13 | 61.91 | 47.67 |
| | ResNeXt-50 [23] | 313G | 44.53 | 62.45 | 48.38 |
| | ResNeXt-101 | 422G | 45.83 | 63.61 | 49.89 |
| | ResNeSt-50 [24] | 336G | 45.41 | 63.92 | 48.70 |
| | ResNeSt-101 | 451G | 47.51 | 66.06 | 51.35 |
| | LA-ResNet-50 (**ours**) | 173G | 46.81 (+4.36) | 65.40 (+5.64) | 50.42 (+4.33) |
| | LA-ResNet-101 (**ours**) | 226G | 48.83 (+4.70) | 67.59 (+5.68) | 52.86 (+5.19) |
| | LA-ResNeXt-50 (**ours**) | 301G | 47.56 (+3.03) | 66.24 (+3.70) | 51.29 (+2.91) |
| | LA-ResNeXt-101 (**ours**) | 399G | **49.64** (+3.81) | **68.25** (+4.64) | **53.68**(+3.79) |

### *B. Full Attention Network Implementation*

We use the LA-layer to replace the $3 \times 3$ convolution layers in ResNet [22], ResNeXt [23], and ResNeSt [24]. We chose these networks for illustration, because they are widely used and understood. The corresponding full attention networks are referred to as LA-ResNet, LA-ResNeXt, and LA-ResNeSt, respectively. We also replace the $3 \times 3$ convolution layers in other state-of-the-art local attention approaches for further comparison.

## IV. Experiments

We evaluate our LA-layer on common computer vision tasks: image classification, object detection, and semantic segmentation (Sections IV-A-IV-C), followed by qualitative results to better understand how our LA-layer leads to improved performance (Section IV-D). We investigate the influence of constraint region size $l$ in Section IV-E.

### *A. ImageNet Classification*

**Setup**: We perform experiments on ImageNet-1K image classification [26], which contains 1.28M training images and 50k test images. For the LA-layer, we use a kernel size of $k = 3$, constraint region size $l = 7$ and $m = 8$ attention heads. We adopt the training setup of [18]. Specifically, we perform SGD optimization with a batch size of 512 on 8 NVIDIA A100 GPUs for all experiments. The total training period is 110 epochs, with a weight decay of 0.0001 and a momentum of 0.9. For the first five epochs, the initial learning rate is 0.4 with linear warm-up. The learning rate is decayed via a cosine schedule [27].

**Results**: Table I shows the results of the full attention ResNet (LA-ResNet) compared to the convolution baseline, and state-of-the-art local attention and deformable attention methods. Using an advanced training setting highly improves the performance. More results with these advanced settings are presented in the supplementary material.

Compared to the ResNet-50 baseline, the full attention network LA-ResNet-50 achieves 2.4% higher classification accuracy (from 77.3% to 79.7%), while having 17.1% fewer floating point operations (FLOPs) and 17.6% fewer parameters to train. This performance gain is consistent for ResNet-101 (+2.5%), ResNeXt-50 (+2.4%) and ResNeXt-101 (+2.4%). Besides, LA-layer outperforms all state-of-the-art local attention methods with comparable FLOPs or parameters. For example, LA-ResNet-50 achieves higher top-1 accuracy (+0.8%) than deformable local attention method DAT-T-ResNet-50 [11]. Compared with Swin-ResNet-50 [14], LA-ResNet-50 achieves +1.3% higher top-1 accuracy with 20% fewer FLOPs. With similar FLOPs, our LA-ResNets outperform LR-Nets [19] by 2.4-2.5% on top-1 accuracy.

### *B. Object Detection*

**Setup**: To further understand how our local attention layer performs on a more fine-grained task, we experiment on object detection on MS COCO dataset [28]. We use LA-ResNets and LA-ResNeXts pretrained on ImageNet-1K as the backbones with Faster R-CNN [29] and Cascade R-CNN [30] as the detection heads. The standard AP metric of single scale is adopted for evaluation. For a fair comparison, we follow the method in [23] and train our models on the COCO-2017 training set (118K images) and evaluate them on COCO-2017 validation set (5K images). During the training process, the $1\times$ learning rate schedule is used, and the size of the shorter side is sampled from the range [640, 800] for each input image during the data augmentation process. All the other hyper-parameters remain the same for fair comparison with other backbones.

**Results.** We summarize our object detection results in Table II. Our models with LA-layer consistently shows better performance than the baselines with both Fast R-CNN and Cascade R-CNN detection heads. For example, compared to ResNet-50, our LA-ResNet-50 shows 4.94% higher AP with Fast R-CNN, and 4.36% higher AP with Cascade R-CNN. Compared to ResNeXt-101, our LA-ResNext-101 achieves

TABLE III: **Semantic segmentation results** on ADE20K validation split. The difference with the original ResNet/ResNeSt models is shown in parentheses. Best results in **bold**.

| Backbone | Pix Acc. | mIoU |
|---|---|---|
| ResNet-50 [22] | 80.39 | 42.10 |
| ResNet-101 | 81.11 | 44.14 |
| ResNeSt-50 [5] | 81.17 | 45.12 |
| ResNeSt-101 | 82.07 | 46.91 |
| LA-ResNet-50 (**ours**) | 80.91 (+0.42) | 43.31 (+1.21) |
| LA-ResNet-101 (**ours**) | 81.73 (+0.62) | 45.46 (+1.32) |
| LA-ResNeSt-50 (**ours**) | 81.70 (+0.53) | 46.22 (+1.10) |
| LA-ResNeSt-101 (**ours**) | **82.59** (+0.52) | **48.02** (+1.11) |

3.80% higher AP with Fast R-CNN, and 3.81% higher AP with Cascade R-CNN. Also, the number of FLOPs for all models with LA-layer is comparable. Again, these results demonstrate that the improvements are not achieved by using a more complex model, but originate from the ability to encode more informative features.

### C. Instance Segmentation

**Setup:** We also evaluate the effectiveness of the LA-layer on the challenging scene parsing dataset ADE20K [31]. We use DeepLabV3 [32] as the instance segmentation approach with ResNet and ResNext backbones pretrained on ImageNet-1K. A resolution of $512 \times 2048$ is used, and we report the pixel accuracy (pixAcc) and mean intersection-of-union (mIoU) as the evaluation metrics. We follow the method in [24] and train the models for 120 epochs on the ADE20K training set (20K images). The trained networks are evaluated on the ADE20K validation set (2K images).

**Results:** The evaluation results are shown in Table III. We observe that networks with LA-layers consistently outperform their ResNet and ResNeSt baselines with solely convolution layers for both pixel accuracy (pixAcc) and mean intersection-of-union (mIoU). The improvements for pixel accuracy are modest, but consistent. For the mIoU, the improvements are a bit higher. For example, compared to ResNeSt-101, our LA-ResNeSt-101 achieves 0.52% higher pixel accuracy and achieves double higher mIoU (1.11%). This further validates the effectiveness of our LA-layer when applied to downstream tasks.

### D. Qualitative Analysis of LA-layer Attention Map

To understand qualitatively how the LA-layer facilitates the extraction of spatially distributed image patterns, we use Grad-CAM [33] to visualize which parts of the input a trained model attends to. We compare ResNet-50, DAT-ResNet-50, and LA-ResNet-50 models trained on the image classification task on ImageNet-1K.

Figure 2 shows heatmaps for these models on three random samples. Both DAT and LA-layer models attend to more of the object regions than ResNet, which explains the higher performance of DAT and LA-layer models (see Table I). Although DAT also contains the deformable kernel that makes the focus areas more distributed over the whole object, the network tends to concentrate on specific regions of the object.

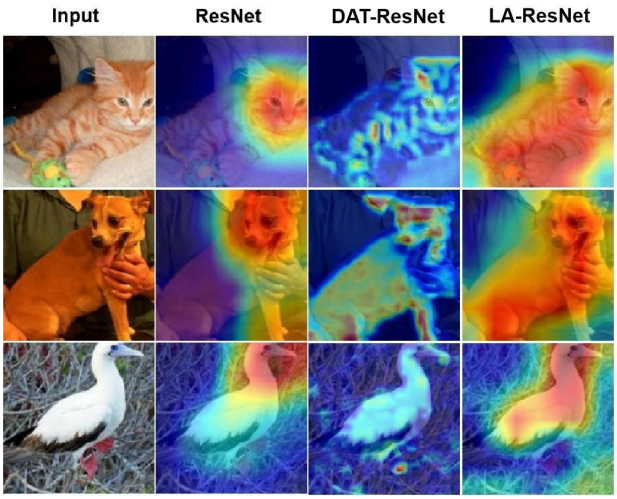


Fig. 2: **Grad-CAM comparisons** for three architectures. Our LA-layer covers more areas over the object compared with others.

TABLE IV: **Image classification ablation study** on ImageNet-1K with LA-ResNet models. Best results in **bold**.

| Top-1(%) / Value of $l$ / Model | 3 | 5 | 7 | 9 | 11 |
|---|---|---|---|---|---|
| LA-ResNet-50 | **82.3** | 82.0 | 81.8 | 81.2 | 81.2 |
| LA-ResNet-101 | 83.0 | 83.4 | **83.5** | 83.0 | 82.7 |
| LA-ResNeXt-50 | 82.7 | 82.9 | **83.0** | 82.6 | 82.5 |
| LA-ResNeXt-101 | 83.4 | **83.9** | **83.9** | 83.4 | 83.2 |

This reduces the quantity of captured relevant features and limits the receptive field of the network to some extent, thus reducing the performance of the network. Compared with DAT, the constraint rule on the deformable kernel in the LA-layer helps the network to cover more of the object of interest and allows for the integration of a multitude of informative, possibly complementary, features to contribute to the final image classification.

### E. Ablation: Effect of Constraint Region Size

We investigate the effect of the value of the constraint region size $l$ on the overall accuracy through an ablation study on the ImageNet-1K image classification task. We follow the training method in DeiT [34] to train these models from scratch. All experimental results are trained for 300 epochs with the AdamW optimizer. We vary $l$ from 3 to 11, while the kernel size is fixed at $3 \times 3$.

From Table IV we can see that for the LA-ResNet-50 model, the highest accuracy is obtained when $l$ is the same as the kernel size, and the performance continuously decreases when further increasing $l$. The situation changes for the LA-ResNet-101, where the highest accuracy is achieved when $l$ is around two times of the kernel size. The same applies to the LA-ResNeXt-50 and LA-ResNeXt-101. So we infer that the proper value of $l$ for LA-layer should be around 1–2 times the kernel size. When increasing the constraint region, kernels in different areas tend to concentrate on specific features and produce higher weights, thus lower the receptive field of the network

to some extent. In this case, informative features with lower weights are neglected, such that the performance starts to decrease. We therefore conclude that a good balance between focusing on more globally relevant features and including more local features is optimal. Our LA-layer can achieve just this balance.

## V. Conclusion

This paper introduces a novel local attention layer (LA-layer), a basic image feature extractor that overcomes the short-range problem of convolution layers and addresses limitations of traditional self-attention layers. LA-layers can straightforwardly replace convolution layers to obtain full attention models. Experimentation on ImageNet-1K image classification demonstrates improved performance over ResNet backbones and other local attention approaches. Moreover, LA-layers require fewer parameters and FLOPs than related methods, which suggests that the improved performance is due to the extraction of more informative features, rather than being the result of a more complex model. On object detection (MS COCO) and instance segmentation (ADE20K) tasks, models with LA-layers also show significantly better performance compared to the original networks. The visualization result based on GradCAM further validates the effectiveness of the LA-layer, and the ablation study on the constraint region size proposes a good balance between focusing on more globally relevant features and including more local features. We expect that these performance gains of the LA-layer also extend to more complex CNN architectures.

## Acknowledgment

This work is supported in part by the scholarship from China Scholarship Council (CSC) under the Grant No.202106290068.

## References

[1] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár, "Designing network design spaces," in *CVPR*, 2020, pp. 10428–10436.

[2] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[3] 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, pp. 5998–6008.

[4] 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," in *ICLR*, 2021.

[5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021, pp. 10012–10022.

[6] 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 vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[7] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021, pp. 6836–6846.

[8] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens, "Stand-alone self-attention in vision models," *NeurIPS*, vol. 32, 2019.

[9] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang, "CvT: Introducing convolutions to vision transformers," in *ICCV*, 2021, pp. 22–31.

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *ICCV*, 2017, pp. 764–773.

[11] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang, "Vision transformer with deformable attention," in *CVPR*, 2022, pp. 4794–4803.

[12] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[13] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," in *CVPR*, 2022, pp. 11963–11975.

[14] Namuk Park and Songkuk Kim, "How do vision transformers work?," in *ICLR*, 2022.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, vol. 25, pp. 1097–1105, 2012.

[16] François Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1251–1258.

[17] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv:1511.07122*, 2015.

[18] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei, "Contextual transformer networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[19] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin, "Local relation networks for image recognition," in *CVPR*, 2019, pp. 3464–3473.

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

[21] Md Amirul Islam, Sen Jia, and Neil D. B. Bruce, "How much position information do convolutional neural networks encode?," in *ICLR*, 2020.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[23] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 1492–1500.

[24] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al., "ResNeSt: Split-attention networks," in *CVPR*, 2022, pp. 2736–2746.

[25] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le, "Attention augmented convolutional networks," in *ICCV*, 2019, pp. 3286–3295.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[27] Ilya Loshchilov and Frank Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv:1608.03983*, 2016.

[28] 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, pp. 740–755.

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2961–2969.

[30] Zhaowei Cai and Nuno Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018, pp. 6154–6162.

[31] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Scene parsing through ADE20K dataset," in *CVPR*, 2017, pp. 633–641.

[32] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587*, 2017.

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021, pp. 10347–10357.