arXiv:1904.05873v1 [cs.CV] 11 Apr 2019

An Empirical Study of Spatial Attention Mechanisms in Deep Networks

Xizhou Zhu[1,3†*] Dazhi Cheng[2,3†*] Zheng Zhang[3*] Stephen Lin[3] Jifeng Dai[3]

[1]University of Science and Technology of China
[2]Beijing Institute of Technology
[3]Microsoft Research Asia

ezra0408@mail.ustc.edu.cn

{v-dachen,zhez,stevelin,jifdai}@microsoft.com

Abstract

Attention mechanisms have become a popular component in deep neural networks, yet there has been little examination of how different influencing factors and methods for computing attention from these factors affect performance. Toward a better general understanding of attention mechanisms, we present an empirical study that ablates various spatial attention elements within a generalized attention formulation, encompassing the dominant Transformer attention as well as the prevalent deformable convolution and dynamic convolution modules. Conducted on a variety of applications, the study yields significant findings about spatial attention in deep networks, some of which run counter to conventional understanding. For example, we find that the comparison of query and key content in Transformer attention is negligible for self-attention, but vital for encoder-decoder attention. On the other hand, a proper combination of deformable convolution with key content saliency achieves the best accuracy-efficiency tradeoff in self-attention. Our results suggest that there exists much room for improvement in the design of attention mechanisms.

1. Introduction

Attention mechanisms enable a neural network to focus more on relevant elements of the input than on irrelevant parts. They were first studied in natural language processing (NLP), where encoder-decoder attention modules were developed to facilitate neural machine translation [2, 31, 16]. In computing the output for a given query element (*e.g.*, a target word in the output sentence), certain key elements (*e.g.*, source words in the input sentence) are prioritized according to the query. Later, self-attention modules were presented for modeling intra-sentence relations [7, 29, 33, 34, 41], where both the key and query are from the same set of elements. In a milestone paper [41], the Transformer attention module is presented, superseding past works and substantially surpassing their performance. The success of attention modeling in NLP has led to its adoption in computer vision, where different variants of Transformer attention are applied to recognition tasks such as object detection and semantic segmentation [22, 43, 19, 24, 51, 15], where the query and key are visual elements such as image pixels or regions of interest.

In determining the attention weight assigned to a certain key for a given query, there exist just a few properties of the input that are commonly considered. One is the content of the query. For the case of self-attention, the query content may be the features at the query pixel in an image, or of a word in a sentence. Another is the content of the key, where a key may be a pixel within a local neighborhood of the query, or another word within the sentence. The third is the relative position of the query and key.

Based on these input properties, there are four possible *attention factors* from which the attention weight for a key with respect to a query is determined, as these factors must account for information about the key. Specifically, these factors are (1) the query and key content, (2) the query content and relative position, (3) the key content only, and (4) the relative position only. In the latest version of Transformer attention [11], attention weights are expressed as a sum of four terms ($\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, $\mathcal{E}_4$), one for each of these attention factors as illustrated in Fig. 1. The nature of the dependencies involved with these terms vary. For example, the first two ($\mathcal{E}_1$, $\mathcal{E}_2$) are sensitive to the query content. While, the latter two ($\mathcal{E}_3$, $\mathcal{E}_4$) do not account for query content, but rather they mainly capture salient key elements and exploit global positional biases, respectively. Although attention weights can be decomposed into terms

*Equal contribution. †This work is done when Xizhou Zhu and Dazhi Cheng are interns at Microsoft Research Asia.

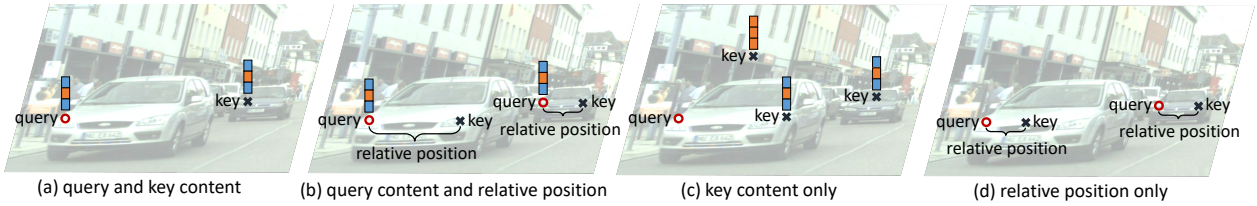


Figure 1. Illustration of different attention terms. The color bar above a sampling point denotes its content feature. The existence of content features and/or relative position indicates that the term uses them for attention weight calculation.

based on these factors, their relative significance in various inference problems has not been closely examined in the literature. Moreover, prevalent modules like deformable convolution [10, 52] and dynamic convolution [44], though seemingly orthogonal to Transformer attention, also employ mechanisms that focus on certain parts of an input. Whether these modules can all be viewed from a unified perspective and how their operational mechanisms differ also have not been explored.

In this work, we perceive Transformer attention, deformable convolution, and dynamic convolution modules as various instantiations of spatial attention, involving different subsets of the attention factors and accounting for these factors in different ways. Towards disentangling the effects of different attention factors and mechanisms, we present an empirical study of spatial attention, in which various elements of attention mechanisms are ablated within a generalized attention formulation. This investigation is conducted on a variety of applications, namely neural machine translation, semantic segmentation, and object detection. From this study, we find that: 1) In the Transformer attention module, the query-sensitive terms, especially the query and key content term, play a minor role in self-attention. But in encoder-decoder attention, the query and key content term is vital. 2) Though deformable convolution utilizes an attention mechanism based only on the query content and relative position term, it operates more effectively and efficiently on image recognition than the counterpart in Transformer attention. 3) In self-attention, the factors of query content & relative position and key content only are the most important. A proper combination of deformable convolution and the key content only term in Transformer attention delivers higher accuracy than that of the Transformer attention module, with much lower computational overhead on image recognition tasks.

结论：需要进一步总结

The observations made in this paper challenge the conventional understanding of current spatial attention mechanisms. For example, it is widely believed that their success can mainly be attributed to query-sensitive attention, especially the query and key content term. This understanding perhaps originates from the initial success of the encoder-decoder attention module in neural machine translation. Thus, in some recent variants [43, 24, 48, 15], like the non-local block [43] and criss-cross attention module [24], only the query and key content term is kept, with all the other terms removed. These modules still function well in self-attention applications, which strengthen this perception. However, our study suggests that this understanding is incorrect. We find that these attention modules with only query-sensitive terms actually perform on par with those with only query-irrelevant terms. Our study further suggests that this degeneration is likely due to the design of the attention modules, rather than an inherent characteristic of self-attention, since deformable convolution is found to exploit query content & relative position effectively and efficiently in image recognition tasks.

This empirical analysis suggests that there is much room for improvement in the design of spatial attention mechanisms in deep networks. Its findings are used in this paper to make some initial headway in this direction, and it is hoped that this study will spur further investigation into the operational mechanisms used in modeling spatial attention.

2. Related Work

Development and application of attention-based modules. The field of NLP has witnessed steady development of attention mechanisms in recent years [2, 31, 16, 41, 38, 11]. Starting from the introduction of an attention module in neural machine translation [2], various attention factors and weight assignment functions based on these factors have been utilized. In [31], the inner product of vectors encoding query and key contents is recommended for computing attention weights, and absolute spatial positions are incorporated as an attention factor. In [16], the weight assignment additionally accounts for the inner product of spatial positions encoded in high-dimensional vectors. The landmark work of Transformer [41] set a new standard, and its latest variants use relative positions instead of absolute positions for better generalization ability [38, 11]. In this paper, we conduct the empirical study on the latest instantiation of Transformer attention [11] from this family of works.

Motivated by their success in NLP tasks [2, 31, 16, 7, 29, 33, 34, 41], attention mechanisms have also been employed in computer vision applications such as relational reasoning among objects [3, 37], image captioning [46], image generation [50, 47], image recognition [22, 43, 19, 24, 51, 15], and video recognition [53, 45]. In vision, the key and

query refer to visual elements, but aside from that, most of these works use a formulation similar to Transformer attention. Since the effects of different attention module elements may vary with the target application, we conduct the empirical study on three different tasks that have been influenced greatly by attention modeling, namely neural machine translation in NLP, and object detection and semantic segmentation in computer vision.

Aside from Transformer attention, there are variants of convolution, such as deformable convolution [10, 52] and dynamic convolution [44], that also can be viewed as types of attention mechanisms which operate on a subset of the attention factors using different attention weight functions. They also are included in the study for examination.

It is worth mentioning a dual form of spatial attention, called channel-wise feature attention [42, 49, 23, 15]. As different feature channels encode different semantic concepts, these works seek to capture the correlations among these concepts through activation/deactivation of certain channels. Meanwhile, in the spatial domain, relationships among elements at different spatial positions are modeled, with the same attention weights on feature channels assigned to related spatial positions. The development of channel-wise feature attention has been focused on certain image recognition tasks, like semantic segmentation and image classification. In this paper, our empirical study specifically examines spatial attention mechanisms designed for broad application.

Analysis of spatial attention mechanisms. There exists relatively little analysis of spatial attention mechanisms despite their prevalence in deep networks. This research has largely been conducted by visualizing or analyzing the learned attention weights of a whole attention module on only NLP tasks [17, 40, 18, 25]. Many works [17, 40, 18] suggest that attention weight assignment in encoder-decoder attention plays a role similar to word alignment in traditional approaches [1, 8, 30, 6]. The implicit underlying assumption in these works is that the input elements accorded high attention weights are responsible for the model outputs. However, recent research casts doubt on this assumption [25], finding that attention weights do not correlate well with feature importance measures, and that counterfactual attention weight configurations do not yield corresponding changes in prediction.

In this paper, we conduct the first comprehensive empirical study on the elements of spatial attention modules over both NLP and computer vision tasks. Different attention factors and weight assignment functions are carefully disentangled, with their effects directly measured by the final performance on these tasks.

3. Study of Spatial Attention Mechanisms

To facilitate our study, we develop a generalized attention formulation that is able to represent various module designs. We then show how the dominant attention mechanisms can be represented within this formulation, and how ablations can be conducted using this formulation with respect to different attention module elements.

Generalized attention formulation

Given a query element and a set of key elements, an attention function adaptively aggregates the key contents according to attention weights that measure the compatibility of query-key pairs. To allow the model to attend to key contents from different representation subspaces and different positions, the outputs of multiple attention functions (heads) are linearly aggregated with learnable weights. Let q index a query element with content z_q, and k index a key element with content x_k. Then the multi-head attention feature y_q is computed as

$$y_q = \sum_{m=1}^{M} W_m \Big[\sum_{k \in \Omega_q} A_m(q, k, z_q, x_k) \odot W'_m x_k \Big], \qquad (1)$$

where m indexes the attention head, Ω_q specifies the supporting key region for the query, $A_m(q, k, z_q, x_k)$ denotes the attention weights in the m-th attention head, and W_m and W'_m are learnable weights. Usually, the attention weights are normalized within Ω_q, as $\sum_{k \in \Omega_q} A_m(q, k, z_q, x_k) = 1$.

In encoder-decoder attention, the key and the query are from two different sets of elements, where in most applications the two sets of elements need to be properly aligned. For example, in the encoder-decoder attention of neural machine translation, the key and the query elements correspond to the words in the input and the output sentences, respectively, where proper alignment is necessary for correct translation. Meanwhile, in self-attention, the key and the query are from the same set of elements. For example, both the key and the query are of words in the input or output sentence. In such scenarios, the self-attention mechanism is expected to capture intra-relationships among the elements, and usually the query and the key contents are modeled by the same set of features, *i.e.*, $x = z$.

Transformer attention

In the most recent instantiation of the Transformer attention module [11], the attention weight of each query-key pair is computed as the sum of four terms $\{\mathcal{E}_j\}_{j=1}^{4}$ that are based on different attention factors, as

$$A_m^{\text{Trans}}(q, k, z_q, x_k) \propto \exp\Big(\sum_{j=1}^{4} \mathcal{E}_j \Big), \qquad (2)$$

normalized by $\sum_{k \in \Omega_q} A_m^{\text{Trans}}(q, k, z_q, x_k) = 1$ where the supporting key region Ω_q spans the key elements (*e.g.*, the

attention mechanism		spatial properties	query content	key content	relative position	complexity
Transformer attention	$\mathcal{E}_1$	dense, global	✓	✓		$O(N_s^2C + N_sC^2)$
	$\mathcal{E}_2$	dense, global	✓		✓	$O(N_s^2C + N_sC^2)$
	$\mathcal{E}_3$	dense, global		✓		$O(N_sC^2)$
	$\mathcal{E}_4$	dense, global			✓	$O(N_s^2C + N_sC^2)$
Regular convolution		sparse, local			✓	$O(N_sC^2N_k)$
Deformable convolution		sparse, global	✓		✓	$O(N_sC^2N_k)$
Dynamic convolution		sparse, local	✓		✓	$O(N_sCN_gN_k + N_sC^2)$

Table 1. Comparison of different attention mechanisms. N_s denotes number of spatial elements, i.e. width by height for images, and number of tokens for text; C denotes representation dimension; N_k denotes kernel size of convolution ($N_k = 3 \times 3$ for images and $N_k = 3$ for text, by default); N_g denotes number of feature groups in dynamic convolution.

whole input sentence). By default, 8 attentional heads are utilized in this paper.

The $\mathcal{E}_1$ and $\mathcal{E}_2$ terms are sensitive to the query content. The $\mathcal{E}_1$ term measures the compatibility of the query and key content, as $\mathcal{E}_1 = z_q^\top U_m^\top V_m^{\mathrm{C}} x_k$, where U_m, V_m^{C} are learnable embedding matrices for the query and key content, respectively. It enables the network to focus more on the keys compatible with the query in terms of content. A possible outcome is the correspondence between similar query and key elements, as illustrated in Fig. 1 (a). For the $\mathcal{E}_2$ term, it is based on the query content and relative position, as $\mathcal{E}_2 = z_q^\top U_m^\top V_m^{\mathrm{R}} R_{k-q}$, where R_{k-q} encodes the relative position $k-q$ by projecting it to a high-dimensional representation through computing sine and cosine functions of different wavelengths[1] [41]. V_m^{R} is a learnable embedding matrix for the encoded relative position R_{k-q}. This term allows the network to adaptively determine where to assign high attention weights based on the query content. It may help to disentangle appearance from spatial transformations in image recognition, as illustrated in Fig. 1 (b).

The $\mathcal{E}_3$ and $\mathcal{E}_4$ terms are irrelevant to the query content. The $\mathcal{E}_3$ term involves key content only, as $\mathcal{E}_3 = u_m^\top V_m^{\mathrm{C}} x_k$, where u_m is a learnable vector. It captures salient key content which should be focused on for the task, and is irrelevant to the query. An illustration is shown in Fig. 1 (c). As for the $\mathcal{E}_4$ term, it involves relative position only, as $\mathcal{E}_4 = v_m^\top V_m^{\mathrm{R}} R_{k-q}$, where v_m is a learnable vector. It captures global positional bias between the key and query elements, as illustrated in Fig. 1 (d).

It is widely believed that query-sensitive prioritization, especially the query and key content compatibility term $\mathcal{E}_1$, is the key to the success of Transformer attention. Thus, in some recent variants [43, 24, 48, 15], only $\mathcal{E}_1$ is kept, while the other terms are all removed.

In Transformer attention, both W_m and W'_m in Eq. (1) are learnable. W'_m projects the features of x_k to a relatively low dimension for reducing computational overhead, and W_m projects the aggregated features back to the same dimension as y_q.

[1]For 2-d image data, we separately encode the x-axis relative position R^{X}_{k-q} and y-axis relative position R^{Y}_{k-q}, and concatenate them to be the final encoding $R_{k-q} = [R^{\mathrm{X}}_{k-q}, R^{\mathrm{Y}}_{k-q}]$.

Regular and deformable convolution

Regular and deformable convolution can be deemed as special instantiations of spatial attention mechanisms, where subsets of the attention factors are involved.

In regular convolution, given a query element, a fixed number of key elements (*e.g.*, 3×3) are sampled, according to predetermined positional offsets with respect to the query. From the perspective of Eq. (1), the attention weight of regular convolution can be expressed as

$$A_m^{\text{regular}}(q,k) = \begin{cases} 1 & \text{if } k = q + p_m \\ 0 & \text{else,} \end{cases} \tag{3}$$

where each sampled key element is of a separate attention head (*e.g.*, 3×3 regular convolution corresponds to 9 attention heads), and p_m denotes the offset for the m-th sampling position. In addition, the weight W'_m in Eq. (1) is fixed as identity, leaving W_m as learnable. In regular convolution, only relative position is involved, without learnable parameters for adapting attention to content. The supporting key region Ω_q is restricted to a local window centered at the query position and determined by the convolution kernel size.

In deformable convolution [10, 52], learnable offsets are added to adjust the sampling positions of the key elements, so as to capture spatial transformations. The learnable offsets are predicted based on the query content, and are thus dynamic to the input. The key and the query elements are from the same set. It can also be incorporated into the generalized attention formulation as a special instantiation of self-attention, where the attention weight is

$$A_m^{\text{deform}}(q,k,x_q) = G(k, q + p_m + w_m^\top x_q), \tag{4}$$

where p_m also denotes a predetermined offset, and $w_m^\top x_q$ projects the query content x_q to a deformation offset according to a learnable vector w_m[2]. $G(a,b)$ is the bilinear interpolation kernel in N-d space, which can be decomposed into 1-d bilinear interpolations as $G(a,b) = \prod_{n=1}^{N} g(a_n, b_n)$, where a_n and b_n denote the n-th dimension of a and b respectively, and $g(a_n, b_n) = \max(0, 1 -$

[2]Following [10], the learning rate of w_m is set to 0.1 times that of other parameters to stabilize training.

$|a_n - b_n|$). Similar to regular convolution, the weight W'_m in Eq. (1) is fixed as identity.

In deformable convolution, the attention factors are query content and relative position. The supporting key region Ω_q can span over all the input elements due to the introduced learnable offsets, while non-zero weights are assigned to a sparse set of key elements where bilinear interpolation is performed.

Dynamic convolution

Dynamic convolution [44] is recently proposed to replace the Transformer attention module in self-attention, and is claimed to be simpler and more efficient. It is built upon depth-wise separable convolution [21] with shared dynamic kernel weights, which are predicted based on the query content. In depth-wise separable convolution, a standard convolution is factorized into a depth-wise convolution and a 1×1 convolution called a point-wise convolution, for reducing computation and model size. In depth-wise convolution, a single filter is applied to each input channel, which is fixed for all positions. In dynamic convolution, the kernel weights for the depth-wise convolution are dynamically predicted from the input features, followed by a Softmax normalization. For computational savings, the input channels are divided into several groups, where each group shares the same dynamic kernel weights. In the system of [44], an orthogonal module called the gated linear unit (GLU) [12] is applied before the dynamic convolution module to improve accuracy. We include the GLU to respect the original design.

Dynamic convolution can also be incorporated into the general attention formulation in Eq. (1) with minor modifications, where each input feature channel is of a separate attention head. It can be expressed as

$$y_q = \sum_{c=1}^{C_{\text{in}}} W_c \Big[\sum_{k \in \Omega_q} A_c^{\text{dynamic}}(q, k, x_q) \cdot x_{k,c} \Big], \tag{5}$$

where c enumerates the channels of the input features (C_{in} channels in total), $x_{k,c}$ denotes the feature value at the c-th channel of x_k, and W_c is of the 1×1 point-wise convolution. $A_c^{\text{dynamic}}(q, k, x_q)$ is the attention weight specified by the dynamic kernel in depth-wise convolution, written as

$$A_c^{\text{dynamic}}(q, k, x_q) = \begin{cases} K_{j,c} & \text{if } k = q + p_j \\ 0 & \text{else,} \end{cases} \tag{6}$$

where p_j denotes the j-th sampling position in the dynamic kernel, and $K_{j,c}$ is the corresponding kernel weight. Zero attention weight is assigned to keys outside of the kernel. The kernel weight $K_{j,c}$ is predicted from the input features, and is shared among channels in the same group, as

$$K_{j,c} = K_{j,g}^{\text{share}} \propto \exp\left(d_{j,g}^\top x_q\right), \quad g = \lceil \frac{c}{C_{\text{in}}/N_g} \rceil. \tag{7}$$

The input features are divided into N_g groups ($N_g = 16$ by default). $K_{j,g}^{\text{share}}$ denotes the dynamic kernel weight for the g-th group, and $d_{j,g}$ is the corresponding learnable weight vector. $K_{j,g}^{\text{share}}$ is normalized by $\sum_{j=1}^{N_k} K_{j,g}^{\text{share}} = 1$, where N_k denotes the number of elements in the dynamic kernel.

In dynamic convolution, attention assignment is based on the query content and relative position factor. The supporting key region Ω_q is restricted to a local window around the query position covered by the dynamic kernel.

Comparing attention mechanisms

Tab. 1 compares the three attention mechanisms discussed above. Transformer attention exploits comprehensive content and position information from both query and key. The $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_4$ terms require computation proportional to the product of the query and key element numbers, because they involve a traversal of each query-key pair. The $\mathcal{E}_3$ term captures key content only, and thus involves computation linear to the key element number. In neural machine translation, the key and query elements are commonly dozens of words in a sentence, so the computational overheads of $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_4$ are comparable to $\mathcal{E}_3$. In image recognition, the key and query elements consist of numerous pixels in an image. The computational overheads of $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_4$ are thus much heavier than $\mathcal{E}_3$. Note that when the four terms are put together, some computational overhead can be shared among them.

Similar to the $\mathcal{E}_2$ term, deformable convolution also is based on query content and relative position. But deformable convolution samples just a sparse set of key elements for each query, and the complexity is linear to the query element number. Deformable convolution is thus much faster to compute than $\mathcal{E}_2$ for image recognition, and is comparable in speed to $\mathcal{E}_2$ for machine translation.

Dynamic convolution also relies on query content and relative position. The attention weights of key elements are assigned by the dynamic convolution kernel, based on the query content. Non-zero attention weights only exist in a local range covered by the dynamic kernel. The computational overhead is proportional to the product of the kernel size and query element number. Compared to the $\mathcal{E}_2$ term, the computational overhead can be considerably lower if the kernel size is much smaller than the key element number.

We seek to further disentangle the effects of different attention factors, and to facilitate comparison to other instantiations of spatial attention that use a subset of the factors. Thus, manual switches are introduced into the Transformer attention module, which enable us to manually activate / deactivate particular terms. This is expressed as

$$\hat{A}_m^{\text{Trans}}(q, k, z_q, x_k) \propto \exp\Big(\sum_{j=1}^{4} \beta_j^{\text{Trans}} \mathcal{E}_j \Big), \tag{8}$$

where $\{\beta_j^{\text{Trans}}\}$ takes values in $\{0, 1\}$ to control the activa-

tion of corresponding terms, and $\hat{A}_m^{\text{Trans}}(q, k, z_q, x_k)$ is normalized by $\sum_{k \in \Omega_q} \hat{A}_m^{\text{Trans}}(q, k, z_q, x_k) = 1$.

Incorporating attention modules into deep networks

We incorporate various attention mechanisms into deep networks to study their effects. There are different design choices in inserting the modules, *e.g.*, whether to connect them in series or in parallel, and where to place the modules in the backbone network. We empirically observed the results to be quite similar for different well-considered designs. In this paper, we select the design choices in Fig. 2.

For the object detection and semantic segmentation tasks, ResNet-50 [20] is chosen as the backbone and just the self-attention mechanism is involved. The Transformer attention module is incorporated by applying it on the 3×3 convolution output in the residual block. For insertion into a pre-trained model without breaking the initial behavior, the Transformer attention module includes a residual connection, and its output is multiplied by a learnable scalar initialized to zero, as in [43]. The manner of incorporating dynamic convolution is the same. To exploit deformable convolution, the 3×3 regular convolution in the residual block is replaced by its deformable counterpart. The resulting architecture is called "Attended Residual Block", shown in Fig. 2 (a).

In the neuron machine translation (NMT) task, the network architecture follows the Transformer base model [41], where both self-attention and encoder-decoder attention mechanisms are involved. Different from the original paper, we update the absolute position embedding in the Transformer attention module by the latest relative position version [11] as in Eq. 2. Because both deformable convolution and dynamic convolution capture self-attention, they are added to only the blocks capturing self-attention in Transformer. For dynamic convolution, we replace the Transformer attention module by dynamic convolution directly, as in [44]. The architecture is shown in Fig. 2 (b). For its deformable convolution counterpart, because the Transformer model does not utilize any spatial convolution (with kernel size larger than 1), we insert the deformable convolution unit (with kernel size of 3) prior to the input of the Transformer attention module. The resulting architecture is called "Transformer + Deformable", shown in Fig. 2 (c).

4. Experiments and Analysis

4.1. Experimental settings

Image Object Detection

Models are trained on the 118k images of the COCO 2017 [28] train set. Evaluation is done on the 5k images of the COCO 2017 validation set. Accuracy is measured by the standard mean AP scores at different box IoUs (mAP).

Faster R-CNN [36] with Feature Pyramid Networks (FPN) [27] is chosen as the baseline system. ImageNet [13] pre-trained ResNet-50 is utilized as the backbone. The attended residual blocks in Fig. 2 (a) are applied in the last two stages (conv4 and conv5 stages) of ResNet-50. In Transformer attention, the relative position encoding is of the same dimension as the content feature embedding, specifically 256-d and 512-d in the conv4 and conv5 stages, respectively.

Experiments are implemented based on the open source mmdetection [5] code base. The hyper-parameter setting strictly follows FPN [27]. Anchors of 5 scales and 3 aspect ratios are utilized. 2k and 1k region proposals are generated at a non-maximum suppression threshold of 0.7 at training and inference respectively. In SGD training, 256 anchor boxes (of positive-negative ratio 1:1) and 512 region proposals (of positive-negative ratio 1:3) are sampled for backpropagating their gradients. In our experiments, the networks are trained on 8 GPUs with 2 images per GPU for 12 epochs. The learning rate is initialized to 0.02 and is divided by 10 at the 8-th and the 11-th epochs. The weight decay and the momentum parameters are set to 10^{-4} and 0.9, respectively.

Image Semantic Segmentation

Models are trained on the 5,000 finely annotated images of the Cityscapes [9] train set. Evaluation is done on the 500 images of the validation set. The standard mean IoU score (mIoU) is used to measure semantic segmentation accuracy.

The CCNet [24] for semantic segmentation is utilized, with ImageNet pre-trained ResNet-50 and without the criss-cross attention module proposed in [24], which is a variant of Transformer attention. As done for object detection, the attended residual blocks in Fig. 2 (a) are applied in the last two stages. An additional Transformer attention / dynamic convolution module is placed after the ResNet-50 output following the practice in [24] for improving performance.

The hyper-parameter setting strictly follows that in the CCNet paper [24]. In SGD training, the training images are augmented by randomly scaling (from 0.7 to 2.0), randomly cropping (size of 769×769 pixels) and random flipping horizontally. In our experiments, the networks are trained on 8 GPUs with 1 image per GPU for 60k iterations. The "poly" learning rate policy is employed, where the initial learning rate is set as 0.005 and multiplied by $\left(1 - \frac{\text{iter}}{\text{iter}_{\max}}\right)^{0.9}$. Synchronized Batch Normalization [35] is placed after every newly added layer with learnable weights. The weight decay and the momentum parameters are set as 10^{-4} and 0.9, respectively.

Neural Machine Translation (NMT)

Model training is conducted on the standard WMT 2014 English-German dataset, consisting of about 4.5 million sentence pairs. Sentences are encoded using byte-pair encoding [4], with a shared source-target vocabulary of about 37k tokens. Evaluation is on the English-to-German new-

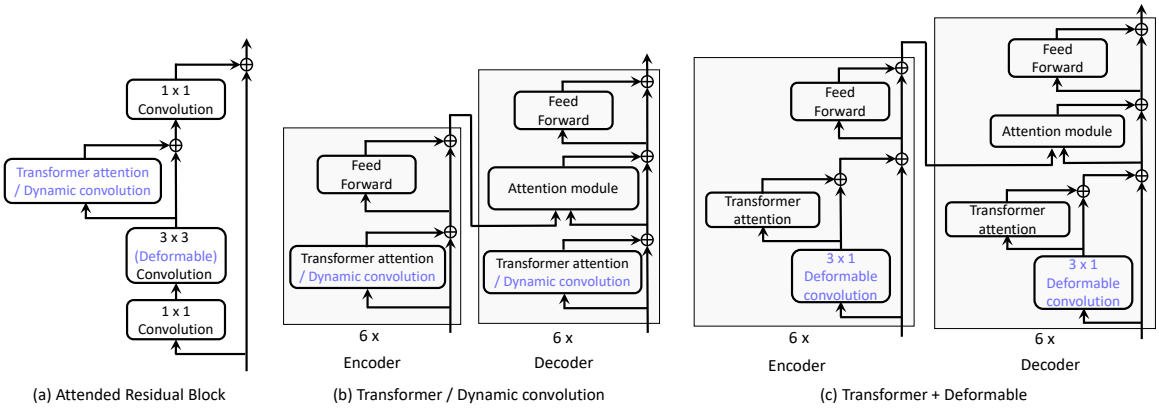


Figure 2. Illustration of attention module configurations for empirical study. The modules in blue color are newly added to existing blocks.

stest2014 set. Accuracy is measured by the standard bilingual evaluation understudy (BLEU) scores [32].

The Transformer base model [41] with relative position encoding [11] is utilized as the backbone. Experiments are implemented based on the open source fairseq [14] code base. The hyper-parameters follows the original setting in [41]. We used the Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. In our experiments, the networks are trained on 8 GPUs for 100k iterations. Each training batch contained a set of sentence pairs containing approximately 30k source tokens and 30k target tokens. The initial learning rate is set as 10^{-7} and linearly increased to 0.001 after $\text{iter}_{\text{warmup}} = 4000$ iterations, and then multiplied by $\left(\frac{\text{iter}}{\text{iter}_{\text{warmup}}}\right)^{-0.5}$. No weight decay is adopted. During training, label smoothing [39] of value 0.1 is employed.

4.2. Effects of various attention-based modules

Disentanglement in Transformer attention

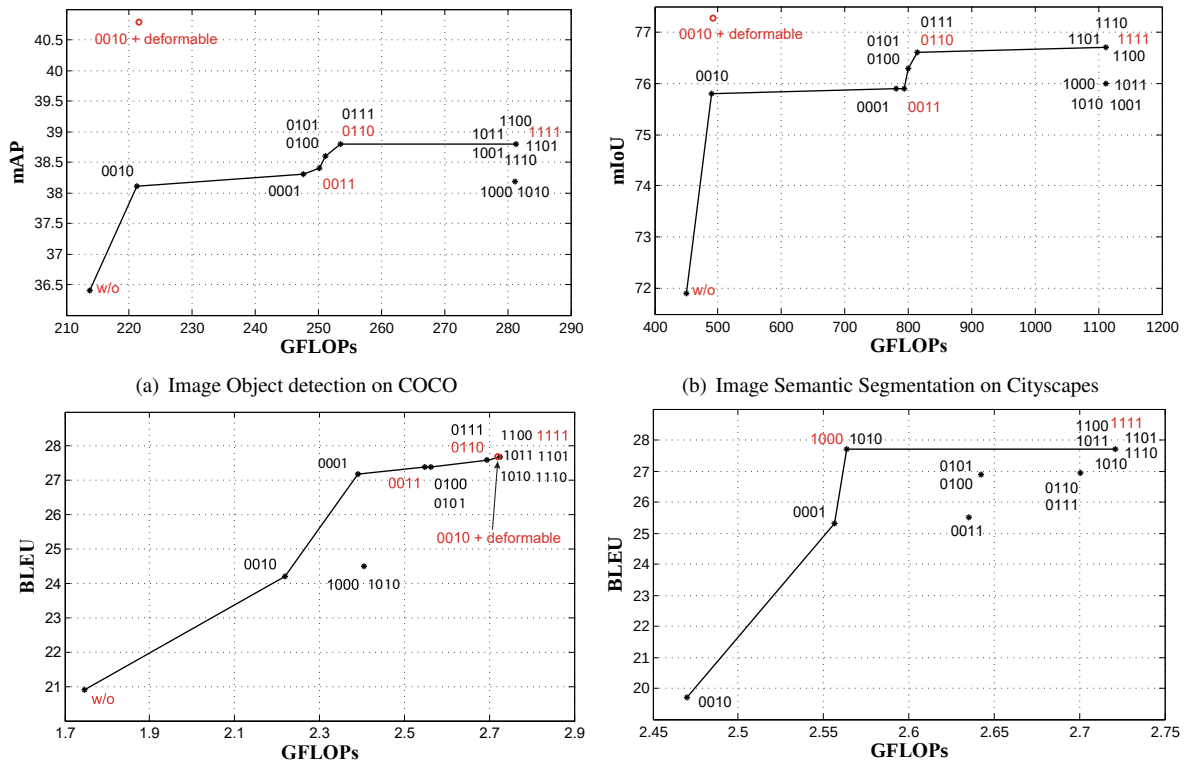
We first seek to disentangle the effects of the four terms in the Transformer attention module. This is achieved by manually setting the $\{\beta_j^{\text{Trans}}\}_{j=1}^4$ values in Eq. (8) to control the activation / deactivation of individual terms. The network is trained and tested for all 16 possible configurations of $\{\beta_j^{\text{Trans}}\}_{j=1}^4$. In this set of experiments, no other attention mechanisms are involved. Thus, for the object detection and semantic segmentation tasks, the 3×3 convolution is of regular convolution in the network of Fig. 2 (a). For the NMT task, the network architecture in Fig. 2 (b) is utilized. Transformer attention is used in the choices of "Transformer attention / Dynamic convolution" in Fig. 2 (a) and (b). Note that for the NMT task, Transformer attention modules are utilized for both self-attention and encoder-decoder attention. To reduce experimental complexity, the Transformer attention modules in encoder-decoder attention are kept as their full version ($\beta_j^{\text{Trans}} = 1, j = 1, \ldots, 4$, abbreviated as configuration "1111" here) when we study self-attention.

Fig. 3 plots the accuracy-efficiency tradeoffs of different $\{\beta_j^{\text{Trans}}\}_{j=1}^4$ configurations, where the accuracy-efficiency envelopes are indicated by connected line segments. Note that only the computational overheads from the Transformer attention modules under study are counted here, without the overheads from other parts of the network. From the plot, we draw the following conclusions:

(1) *In self-attention, the query-sensitive terms play a minor role compared to the query-irrelevant terms.* Especially, the query and key content term have a negligible effect on accuracy, while being computationally heavy in image recognition tasks. Overall, the accuracy gain brought by the Transformer attention module is large (from the configuration where the Transformer attention module is removed ("w/o") to that where the full version of Transformer attention is utilized ("1111")). It can be seen that the gain brought by the query-irrelevant terms (from configuration "w/o" to "0011") is much larger than that brought by the query-sensitive terms (from configuration "0011" to "1111"). Particularly, the performance gain brought by the query and key content term (controlled by β_1^{Trans}) is negligible. Removing it (from configuration "1111" to "0111") incurs only a tiny drop in accuracy, while considerably reducing the computational overhead in image recognition tasks.

(2) *In encoder-decoder attention, the query and key content term is vital.* Deactivation of it (controlled by β_1^{Trans}) incurs a noticeable drop in accuracy, while only utilizing the query and key content term (configuration "1000") delivers accuracy almost the same as the full version (configuration "1111"). This is because the key step in NMT is to align the words in the source and the target sentences. A traversal of the query and key content is essential for such alignment.

(3) *In self-attention, the attention factors of query content & relative position and the key content only are most important.* The corresponding configuration "0110" delivers accuracy very close to the full version (configuration "1111"), while saving a considerable amount of computational overhead in image recognition tasks. It is also worth noting that the key content only term, which captures saliency information, can effectively improve the per-



(a) Image Object detection on COCO

(b) Image Semantic Segmentation on Cityscapes

(c) Translation on newstest2014 (self-attention)

(d) Translation on newstest2014 (encoder-decoder attention)

Figure 3. Accuracy-efficiency tradeoffs of the four terms in Transformer attention ($\mathcal{E}_1$ for query and key content, $\mathcal{E}_2$ for query content and relative position, $\mathcal{E}_3$ for key content only, and $\mathcal{E}_4$ for relative position only). The activation and deactivation of particular terms is set by configuration $\{\beta_j^{\text{Trans}}\}_{j=1}^4$ (*e.g.*, "0011" denotes the activation of $\mathcal{E}_3$ and $\mathcal{E}_4$). Because the encoder-decoder attention mechanism is indispensable for NMT, there is no "w/o" setting in (d). The results of some configurations overlap in the plots because they are of the same accuracy and computational overhead. The key configurations under study are highlighted in red. The recommended configuration of "0010 + deformable" for self-attention in Tab. 2 is also plotted here.

$\beta_{1,2,3,4}^{\text{Trans}} \rightarrow \beta_{1,2,3,4}^{\text{Trans}}$ + deformable	Object Detection (self-attention)				Semantic Segmentation (self-attention)				Neural Machine Translation (self-attention)			
	mAP	Δ mAP	GFLOPs	Δ% FLOPs	mIoU	Δ mIoU	GFLOPs	Δ% FLOPs	BLEU	Δ BLEU	GFLOPs	Δ% FLOPs
w/o → 1111 + deformable	36.4 → **41.0**	+4.6	**213.7** → 281.4	+31.7%	71.9 → **77.8**	+5.9	**449.5** → 1112.1	+147.4%	20.9 → **28.0**	+7.1	**1.7** → 3.2	+88.2%
1111 → 1011 + deformable	38.8 → 41.0	+2.2	281.4 → 281.4	-0.0%	76.7 → 77.8	+1.1	1112.1 → 1112.1	-0.0%	27.7 → 28.0	+0.3	2.7 → 3.2	+17.3%
1110 → 1010 + deformable	38.8 → 40.9	+2.1	281.4 → 281.2	-0.1%	76.7 → 77.7	+1.0	1112.1 → 1111.2	-0.1%	27.7 → 28.0	+0.3	2.7 → 2.9	+5.8%
1101 → 1001 + deformable	38.8 → 41.0	+2.2	281.4 → 281.4	-0.0%	76.7 → 77.8	+1.1	1112.1 → 1112.1	-0.0%	27.7 → 28.0	+0.3	2.7 → 3.2	+17.3%
1100 → 1000 + deformable	38.8 → 40.9	+2.1	281.4 → 281.2	-0.1%	76.7 → 77.7	+1.0	1112.1 → 1111.2	-0.1%	27.7 → 28.0	+0.3	2.7 → 2.9	+5.8%
0111 → 0011 + deformable	38.8 → 41.0	+2.2	253.6 → 250.1	-1.4%	76.6 → 77.5	+0.9	814.0 → 794.4	-2.4%	27.6 → 27.7	+0.1	2.7 → 3.0	+10.9%
0110 → <u>0010 + deformable</u>	38.8 → <u>40.8</u>	+2.0	253.6 → <u>221.1</u>	-12.8%	76.6 → <u>77.3</u>	+0.7	814.0 → <u>489.5</u>	-39.9%	27.6 → <u>27.7</u>	+0.1	2.7 → <u>2.7</u>	-1.1%
0101 → 0001 + deformable	38.6 → 40.7	+2.1	251.1 → 247.6	-1.4%	76.3 → 77.3	+1.0	800.7 → 781.1	-2.5%	27.4 → 27.6	+0.2	2.6 → 2.9	+11.6%
0100 → w/o + deformable	38.6 → 39.9	+1.3	251.1 → 213.7	-14.9%	76.3 → 77.2	+0.9	800.7 → 449.5	-43.9%	27.4 → 27.3	-0.1	2.6 → 2.2	-13.5%

Table 2. Deformable convolution vs. $\mathcal{E}_2$ in Transformer attention, where both exploit query content and relative position information. The underlined configuration of "0010 + deformable" is recommended for an optimal accuracy-efficiency tradeoff.

formance with little additional overhead.

Our findings contradict the widespread belief that query-sensitive terms, especially the query and key content term, are crucial for the success of Transformer attention. The experimental results suggest that this is only true for the encoder-decoder attention scenario. In self-attention scenarios, the query and key content term is even removable.

Deformable convolution vs. $\mathcal{E}_2$ in Transformer attention

Here, we compare deformable convolution and the $\mathcal{E}_2$ term from Transformer attention in Eq. (2). Because deformable convolution is designed for capturing self-attention, we restrict the experiments to self-attention scenarios only. Note that when deformable convolution is utilized in the NMT task, the network architecture is of "Transformer + Deformable" in Fig. 2 (c).

Tab. 2 compares deformable convolution and the $\mathcal{E}_2$ term in a variety of settings. We find that:

(1) *For object detection and semantic segmentation, de-*

$\beta^{\text{Trans}}_{1,2,3,4} \rightarrow$ dynamic	Object Detection (self-attention)				Semantic Segmentation (self-attention)				Neural Machine Translation (self-attention)			
	mAP	Δ mAP	GFLOPs	Δ% FLOPs	mIoU	Δ mIoU	GFLOPs	Δ% FLOPs	BLEU	Δ BLEU	GFLOPs	Δ% FLOPs
0100	**38.6**	-	**251.1**	-	**76.3**	-	**800.7**	-	**27.4**	-	**2.6**	-
0100 ($n_k = 31$) $\rightarrow$ dynamic ($n_k = 31$)	38.6 $\rightarrow$ 37.9	-0.7	229.4 $\rightarrow$ 352.9	+53.8%	75.5 $\rightarrow$ 74.2	-1.3	523.3 $\rightarrow$ 1029.0	+96.6%	27.4 $\rightarrow$ 27.6	+0.2	2.4 $\rightarrow$ 2.4	+1.8%
0100 ($n_k = 25$) $\rightarrow$ dynamic ($n_k = 25$)	38.6 $\rightarrow$ 37.8	-0.8	226.6 $\rightarrow$ 306.8	+35.4%	75.5 $\rightarrow$ 74.2	-1.3	511.8 $\rightarrow$ 840.4	+64.2%	27.4 $\rightarrow$ 27.6	+0.2	2.3 $\rightarrow$ 2.3	+1.4%
0100 ($n_k = 19$) $\rightarrow$ dynamic ($n_k = 19$)	38.6 $\rightarrow$ 37.6	-1.0	224.4 $\rightarrow$ 270.6	+20.6%	75.4 $\rightarrow$ 73.7	-1.7	502.6 $\rightarrow$ 692.1	+37.7%	27.4 $\rightarrow$ 27.5	+0.1	2.3 $\rightarrow$ 2.3	+1.1%
0100 ($n_k = 13$) $\rightarrow$ dynamic ($n_k = 13$)	38.5 $\rightarrow$ 37.5	-1.0	222.7 $\rightarrow$ 244.3	+9.7%	74.4 $\rightarrow$ 71.9	-2.5	495.9 $\rightarrow$ 584.3	+17.8%	27.3 $\rightarrow$ 27.4	+0.1	2.3 $\rightarrow$ 2.3	+0.7%

Table 3. Dynamic convolution vs. $\mathcal{E}_2$ in Transformer attention, where both exploit query content and relative position information. The kernel size of dynamic convolution N_k is n_k^2 for image recognition and n_k for NMT. The spatial range of Transformer attention is also constrained to be the kernel size of dynamic convolution for ablation.

formable convolution considerably surpasses the $\mathcal{E}_2$ term in both accuracy and efficiency. While for NMT, deformable convolution is on par with the $\mathcal{E}_2$ term in both accuracy and efficiency. In terms of efficiency, deformable convolution does not need to traverse all the key elements. This advantage is obvious on images, where numerous pixels are involved. In terms of accuracy, the bilinear sampling in deformable convolution is based on the hypothesis of local linearity of feature maps. This hypothesis holds better on images where local image content changes gradually, than on languages where words change abruptly.

(2) *The combination of deformable convolution and the key content only term ("0010 + deformable") delivers the best accuracy-efficiency tradeoff.* The accuracy is on par with using deformable convolution and the whole attention module ("1111 + deformable"), while the overhead is slightly higher than that of deformable convolution only ("w/o + deformable"). This finding is in line with finding (3) of "Disentanglement in Transformer attention". It further suggests the importance of the query content & relative position and key content only factors in self-attention. The configuration "0010 + deformable" is also plotted in Fig. 3.

Dynamic convolution vs. $\mathcal{E}_2$ in Transformer attention

We compare these two instantiations in self-attention scenarios. The network architectures are of Fig. 2 (a) for image recognition tasks, and of Fig. 2 (b) for NMT, where either the Transformer attention with $\mathcal{E}_2$ only (configuration "0100") or dynamic convolution is utilized.

Tab. 3 presents the results. We can find that for NMT, dynamic convolution achieves accuracy on par with the $\mathcal{E}_2$ term at reduced computational cost. However, dynamic convolution is not effective for object detection and semantic segmentation, delivering considerably lower accuracy. To further study the influence of kernel size in dynamic convolution, we also constrain the spatial range of the $\mathcal{E}_2$ term to be the same as that in dynamic convolution. The accuracy drops as the spatial range shrinks for both dynamic convolution and the $\mathcal{E}_2$ term. But it is worth noting that the $\mathcal{E}_2$ term still surpasses dynamic convolution at the same spatial range in image recognition tasks, with even smaller computational overhead. The inferior accuracy of dynamic convolution in image recognition tasks might be because dynamic convolution is originally designed for NMT, and some design choices may not be suitable for image recognition.

References

[1] T. Alkhouli, G. Bretschner, J.-T. Peter, M. Hethnawi, A. Guta, and H. Ney. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, 2016. 3

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 1, 2

[3] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016. 2

[4] D. Britz, A. Goldie, M.-T. Luong, and Q. Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017. 6

[5] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. mmdetection. https://github.com/open-mmlab/mmdetection, 2018. 6

[6] W. Chen, E. Matusov, S. Khadivi, and J.-T. Peter. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*, 2016. 3

[7] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016. 1, 2

[8] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*, 2016. 3

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6

[10] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 3, 4

[11] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 1, 2, 3, 6, 7

[12] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017. 5

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[14] S. Edunov, M. Ott, and S. Gross. fairseq. https://github.com/pytorch/fairseq, 2017. 7

[15] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018. 1, 2, 3, 4

[16] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017. 1, 2

[17] H. Ghader and C. Monz. What does attention in neural machine translation pay attention to? *arXiv preprint arXiv:1710.03348*, 2017. 3

[18] R. Ghaeini, X. Z. Fern, and P. Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language inference. *arXiv preprint arXiv:1808.03894*, 2018. 3

[19] J. Gu, H. Hu, L. Wang, Y. Wei, and J. Dai. Learning region features for object detection. In *ECCV*, 2018. 1, 2

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[22] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018. 1, 2

[23] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 3

[24] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*, 2018. 1, 2, 4, 6

[25] S. Jain and B. Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019. 3

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[29] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017. 1, 2

[30] L. Liu, M. Utiyama, A. Finch, and E. Sumita. Neural machine translation with supervised attention. *arXiv preprint arXiv:1609.04186*, 2016. 3

[31] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015. 1, 2

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 7

[33] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016. 1, 2

[34] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017. 1, 2

[35] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. In *CVPR*, 2018. 6

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 6

[37] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 2

[38] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 2

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 7

[40] G. Tang, R. Sennrich, and J. Nivre. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. *arXiv preprint arXiv:1810.07595*, 2018. 3

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 2, 4, 6, 7

[42] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *CVPR*, 2017. 3

[43] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 1, 2, 4, 6

[44] F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, and M. Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019. 2, 3, 5, 6

[45] F. Xiao and Y. Jae Lee. Video object detection with an aligned spatial-temporal memory. In *ECCV*, 2018. 2

[46] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[47] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 2

[48] Y. Yuan and J. Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 2, 4

[49] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 3

[50] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 2

[51] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 1, 2

[52] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018. 2, 3, 4

[53] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 2