An Empirical Study of Spatial Attention Mechanisms in Deep Networks

Xizhou Zhu$^{1,2\dagger}$ Dazhi Cheng$^{2\dagger}$ Zheng Zhang$^2$ Stephen Lin$^2$ Jifeng Dai$^2$

$^1$University of Science and Technology of China
$^2$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, 28, 14]. 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 relationships [6, 26, 30, 31, 36], where both the key and query are from the same set of elements. In a milestone paper [36], 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 [20, 38, 17, 22, 46, 13], 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 [10], 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 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 [9, 47]
and dynamic convolution [39], 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.

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 [38, 22, 43, 13], like the non-local block [38] and criss-cross attention module [22], 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-relevant 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, 28, 14, 36, 34, 10]. 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 [28], 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 [14], the weight assignment additionally accounts for the inner product of spatial positions encoded in high-dimensional vectors. The landmark work of Transformer [36] set a new standard, and its latest variants use relative positions instead of absolute positions for better generalization ability [34, 10]. In this paper, we conduct the empirical study on the latest instantiation of Transformer attention [10] from this family of works.

Motivated by their success in NLP tasks [2, 28, 14, 6, 26, 30, 31, 36], attention mechanisms have also been employed in computer vision applications such as relational reasoning among objects [3, 33], image captioning [41], image generation [45, 42], image recognition [20, 38, 17, 22, 46, 13], and video recognition [48, 40]. 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 in-
The key is that the attention weights are normalized within $\Omega_q$ of query-key pairs. To allow the model to attend to key contents from 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 [10], the attention weight of each query-key pair is computed as the sum of four terms $\{E_{ij}\}_{i=1}^4$ that are based on different attention factors, as

$$A^{\text{Trans}}_m(q, k, z_q, x_k) \propto \exp \left( \sum_{j=1}^4 E_j \right),$$

normalized by $\sum_{k \in \Omega_q} A^{\text{Trans}}_m(q, k, z_q, x_k) = 1$ where the supporting key region $\Omega_q$ spans the key elements (e.g., the whole input sentence). By default, 8 attentional heads are utilized in this paper.

The $E_1$ and $E_2$ terms are sensitive to the query content. The $E_1$ term measures the compatibility of the query and key content, as $E_1 = z_q U_m V^C_m x_k$, where $U_m$, $V^C_m$ 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
Table 1. Comparison of different attention mechanisms.  

<table>
<thead>
<tr>
<th>attention mechanism</th>
<th>spatial properties</th>
<th>query content</th>
<th>key content</th>
<th>relative position</th>
<th>complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer attention</td>
<td>dense, global</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>$O(N_k^2C + N_sC^2)$</td>
</tr>
<tr>
<td></td>
<td>sparse, local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regular convolution</td>
<td>sparse, local</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>$O(N_sC^2N_k)$</td>
</tr>
<tr>
<td>Deformable convolution</td>
<td>sparse, global</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>$O(N_sC^2N_k)$</td>
</tr>
<tr>
<td>Dynamic convolution</td>
<td>sparse, local</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>$O(N_sCN_gN_k + N_sC^2)$</td>
</tr>
</tbody>
</table>

The $E_2$ term, it is based on the query content and relative position, as $E_2 = \sum_l U_l^T V_l^R R_{k=q}$, 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\(^1\) [36]. $V_l^R$ 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 $E_3$ and $E_4$ terms are irrelevant to the query content. The $E_3$ term involves key content only, as $E_3 = w_m^T V_m^C x_k$, where $w_m$ 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 $E_4$ term, it involves relative position only, as $E_4 = v_m^T V_m^R R_{k=q}$, 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 $E_1$, is the key to the success of Transformer attention. Thus, in some recent variants\([38, 22, 43, 13]\), only $E_1$ 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 $y_q$.

**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., $3 \times 3$) 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

\[
A_{\text{regular}}(q, k) = \begin{cases} 
1 & \text{if } k = q + p_m \\
0 & \text{else,} 
\end{cases} 
\]  

where each sampled key element is of a separate attention head (e.g., $3 \times 3$ 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 $\Omega_q$ is restricted to a local window centered at the query position and determined by the convolution kernel size.

In deformable convolution\([9, 47]\), 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

\[
A_{\text{deform}}(q, k, x_q) = G(k, q + p_m + w_m^T x_q), 
\]

where $p_m$ also denotes a predetermined offset, and $w_m^T x_q$ 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 $g(a_n, b_n) = \max(0, 1 - |a_n - b_n|)$. 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 $\Omega_q$ 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\([39]\) 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 [19] 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 $1 \times 1$ 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 [39], an orthogonal module called the gated linear unit (GLU) [11] 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

$$y_q = \sum_{c=1}^{C_m} W_c \left[ \sum_{k \in \Omega_q} A_c^{\text{dynamic}}(q, k, x_q) \cdot x_{k,c} \right],$$

where $c$ enumerates the channels of the input features ($C_m$ channels in total), $x_{k,c}$ denotes the feature value at the $c$-th channel of $x_k$, and $W_c$ is of the $1 \times 1$ 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

$$A_c^{\text{dynamic}}(q, k, x_q) = \left\{ \begin{array}{ll} K_{j,c} \cdot K_{j,c} & \text{if } k = q + p_j \\ 0 & \text{else}, \end{array} \right.$$

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

$$K_{j,c} = K_{j,g}^\text{share} \propto \exp \left( d_{j,g}^{T} x_q \right), \quad g = \left\lfloor \frac{c}{C_m/N_g} \right\rfloor.$$  

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 $\Omega_q$ 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 $E_1$, $E_2$ and $E_4$ 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 $E_3$ 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 $E_1$, $E_2$ and $E_4$ are comparable to $E_3$. In image recognition, the key and query elements consist of numerous pixels in an image. The computational overheads of $E_1$, $E_2$ and $E_4$ are thus much heavier than $E_3$. Note that when the four terms are put together, some computational overhead can be shared among them.

Similar to the $E_2$ 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 $E_2$ for image recognition, and is comparable in speed to $E_2$ 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 $E_2$ 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

$$A_m^{\text{Trans}}(q, k, z_q, x_k) \propto \exp \left( \alpha \sum_{j=1}^{4} \beta_j^{\text{Trans}} E_j \right),$$

where $\{\beta_j^{\text{Trans}}\}$ takes values in $\{0, 1\}$ to control the activation of corresponding terms, and $A_m^{\text{Trans}}(q, k, z_q, x_k)$ is normalized by $\sum_{k \in \Omega_q} A_m^{\text{Trans}}(q, k, z_q, x_k) = 1$.

**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.

Detailed descriptions are provided in the Appendix, and a brief introduction is presented here. For the object detection and semantic segmentation tasks, ResNet-50 [18] is chosen as the backbone and just the self-attention mecha-
nism is involved. The three attention mechanisms are incorporated into the residual block. The resulting architecture is called “Attended Residual Block”, shown in Fig. 2 (a).

In the neural machine translation task, the network architecture follows the Transformer base model [36] with relative position encoding [10], where both self-attention and encoder-decoder attention mechanisms are involved. The architecture is shown in Fig. 2 (b). For its deformable convolution counterpart, the deformable convolution unit (with kernel size of 3) is inserted 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 [25] 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 [32] with Feature Pyramid Networks (FPN) [24] is chosen as the baseline system. ImageNet [12] 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. The hyper-parameter setting strictly follows FPN [24]. See the Appendix for details.

Image Semantic Segmentation

Models are trained on the 5,000 finely annotated images of the Cityscapes [8] 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 [22] for semantic segmentation is utilized, with ImageNet pre-trained ResNet-50 and without the criss-cross attention module proposed in [22], 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 [22] for improving performance. The hyper-parameter setting strictly follows that in the CCNet paper [22]. Details are given in the Appendix.

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 newstest2014 set. Accuracy is measured by the standard bilingual evaluation understudy (BLEU) scores [29].

The Transformer base model [36] with relative position encoding [10] is utilized as the backbone. The hyper-parameters follows the original setting in [36]. Further details are in the Appendix.

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 \( \{ \beta_{\text{Trans}}^j \}_{j=1}^4 \) values in Eq. (8) to control the activation / deactivation of individual terms. The network is trained and tested for all 16 possible configurations of \( \{ \beta_{\text{Trans}}^j \}_{j=1}^4 \). In this set of experiments, no other attention mechanisms are involved. Thus, for the object detection and semantic segmentation tasks, the 3 x 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 \((\beta^{\text{Trans}} = 1, j = 1, \ldots, 4, \text{abbreviated as configuration "1111" here})\) when we study self-attention.

Fig. 3 plots the accuracy-efficiency tradeoffs of different \(\{\beta^{\text{Trans}}\}_1^4\) 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 $\beta_{1}^{\text{Trans}}$) 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 $\beta_{1}^{\text{Trans}}$) 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 performance 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, deformable 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] Tamer Alkhouli, Gabriel Bretschnier, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney.


[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes datasets for semantic urban scene understanding. In *CVPR*, 2016. 6


[40] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory. In\textit{ ECCV}, 2018. 2


[42] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In\textit{ CVPR}, 2018. 2


