Decomposed Attention: Self-Attention with Linear Complexities

Shen Zhuoran  Zhang Mingyuan  Yi Shuai  Yan Junjie  Zhao Haiyu
shenzhuoran  zhangmingyuan  yishuai  yanjunjie  zhaohaiyu
SenseTime
@sensetime.com

Abstract

Recent works have been applying self-attention to various fields in computer vision and natural language processing. However, the memory and computational demands of existing self-attention operations grow quadratically with the spatiotemporal size of the input. This prohibits the application of self-attention on large inputs, e.g., long sequences, high-definition images, or large videos. To remedy this drawback, this paper proposes a novel decomposed attention (DA) module with substantially less memory and computational consumption. The resource-efficiency allows more widespread and flexible application. Empirical evaluations on object recognition demonstrated the effectiveness of these advantages. DA-augmented models achieved state-of-the-art performance for object recognition on MS-COCO 2017 and significant improvement for image classification on ImageNet. Further, the resource-efficiency of DA democratizes self-attention to fields where the prohibitively high costs have been preventing its application. The state-of-the-art result for stereo depth estimation on the Scene Flow dataset exemplified this.

1. Introduction

Long-range dependency modeling has been a central challenge for deep learning. The most successful architectures, e.g., convolutional and recurrent neural networks, process a local neighborhood at a time. To model long-range dependencies, the operations must apply repeatedly to increase the receptive field. For example, a 256x256 input image requires 128 consecutive 3x3 convolutions to reach a global receptive field and ensure the ability to model dependencies between any two pixels. Stacking many layers leads to computational inefficiency. In addition, propagating information through a deep network between two distant points might cause optimization difficulties. Furthermore, multi-hop dependency modeling multiplies the resource requirements. For example, modeling 3-hop dependencies on a 256x256 image demands 384 3x3 convolutions. This further reinforces the computational inefficiency and optimization difficulties.

To mitigate this problem, [1] introduced the self-attention mechanism, which computes the response at every location as a weighted average of features at all locations in the previous layer. In contrast to convolution and recurrence, a single self-attention module expands the receptive field to the entire input. Using self-attention to efficiently model long-range dependencies allows convolution and recurrence to focus on local dependency modeling, in which they specialize. Since the advent of self-attention, the method has been remarkably successful. Self-attention-based models now hold state-of-the-art records on virtually all tasks in natural language processing (NLP) [33, 24, 9, 25]. The non-local module [35], an adaptation of self-attention for vision networks, achieved state-of-the-art performance on video classification [35] and generative adversarial image modeling [38, 3] and demonstrated significant improvements on object detection [35], instance segmentation [35], person re-identification [18], and image de-
puts, e.g. long sequences, high-definition images, and large videos, remains an open problem. The quadratic\(^1\) memory and computational complexities of existing self-attention modules inhibit their application on such inputs. For instance, a non-local module uses over 1 GB of memory and over 50 GMACC\(^2\) of computation for a 128x128 feature map or over 68 GB and over 3.2 TMACC for a 64x64x32 video feature volume. This constrains the application of self-attention to the low-resolution parts of models [35, 38, 3] and prohibits its use for resolution-sensitive and resource-hungry tasks.

The need for global dependency modeling on large inputs greatly motivates the exploration for a resource-efficient self-attention algorithm. An investigation into the non-local module revealed an intriguing phenomenon. The attention masks at each location, despite generated independently, are correlated. As [35] and [38] analyzed, the attention mask of a location mainly focuses on semantically related regions. Figure 2 shows the learned attention masks in a non-local module. When generating the image of a bird before a bush, pixels on the bird’s legs tend to attend to other leg pixels for structural consistency. Similarly, body pixels mainly attend to the body, and background pixels focus on the bush.

This observation inspired the design of the decomposed attention module. Given an input, the module generates a set of basis masks, a coefficient for each mask at every location, and a value feature map. Figure 1 shows the conceptual structure of the module. Scaling each mask by its coefficient at a location and summing the scaled masks gives the attention mask for that location. In practice, the module first aggregates the value map by each basis mask and then combines the aggregated contextual features at each location by the coefficients. This procedure avoids the generation of the huge pairwise attention matrix containing the attention mask for every location and hence removes the quadratic terms in the module’s memory and computational complexities.

The principal contribution of this paper is the decomposed attention (DA) module, which:

1. has linear memory and computational complexities in terms of the spatiotemporal size of the input, as section 3.3 proves;
2. possesses similar expressive power as conventional attention modules, as section 4.1.8 shows;
3. allows the insertion of significantly more attention modules that brings substantial performance boosts over conventional attention-based models, as the state-of-the-art results for object detection and instance segmentation on MS-COCO 2017 in section 4.1 and significant improvement for image classification on ImageNet in section 4.3 demonstrate; and
4. facilitates application of self-attention on resource-hungry tasks, as the state-of-the-art result for stereo depth estimation on the Scene Flow dataset in section 4.2 exemplifies.

2. Related Works

2.1. Self-Attention Mechanism

[1] proposed the initial formulation of the self-attention mechanism to improve word alignment in machine translation. Successively, [33] proposed to completely replace recurrence with self-attention and named the resultant architecture the Transformer. They achieved state-of-the-art performance on machine translation and constituency parsing. Following this work, [24] pretrained a deep Transformer on a large corpus for language modeling and achieved state-of-the-art performance on a broad set of NLP tasks, including natural language inference, semantic similarity, and reading comprehension. Subsequently, [9] proposed to use masked language modeling and next sentence prediction for pretraining to incorporate bidirectional information. With this work, Transformers now hold the state-of-the-art records on virtually all NLP tasks and outperform humans on many.

[35] first adapted the self-attention mechanism for computer vision. They achieved state-of-the-art performance on video classification and demonstrated significant improvements on object detection, instance segmentation, and pose estimation. The paper named the proposed component the non-local module. Subsequent works applied it to various fields in computer vision, including image restoration.

---

\(^1\)The complexities are quadratic with respect to the spatiotemporal size of the input, or quartically w.r.t. the side length of an input image, or sextically w.r.t. the dimension of an input video.

\(^2\)MACC stands for multiply-accumulation. 1 MACC means 1 multiplication or addition operation.
[21], de-raining [16], video person re-identification [18], salient segmentation [15], and cardiac motion scoring [37]. Notably, SAGAN [38] and BigGAN [3] incorporated the non-local module to generative adversarial networks and achieved substantial advancements of the state-of-the-art.

2.2. Other Types of Visual Attention

Besides self-attention, there are a separate set of techniques the literature refer to as attention. This section refers to them as scaling attention. While self-attention is effective for global dependency modeling, scaling attention focuses on emphasizing important features and suppressing uninformative ones. [34] proposed to replace the ResBlock with a complex attention module as the backbone component of convolutional networks. The module consists of a convolutional branch and an hourglass attention branch. It multiplies the attention map generated by the hourglass branch to the output of the convolutional branch to achieve effective scaling. This architecture outperformed ResNet with less resource consumption. In comparison to this method, the squeeze-and-excitation (SE) module [14] is a significantly more economical approach. An SE block uses global average pooling and a linear layer to compute a scaling factor for each channel. The low cost allows the insertion of an SE block at every layer of a network, which allowed SE-enhanced models to achieve state-of-the-art performance on image classification and substantial improvements on scene segmentation and object detection. CBAM [36] built upon the SE module by adding global max pooling beside global average pooling and introducing an extra spatial attention submodule. CBAM further improved SE’s performance on image classification and object detection.

2.3. Object Detection

Since the introduction of R-CNN [11], deep learning methods have achieved remarkable progress on object detection. Most CNN detectors belong to one of the two categories: two-stage detectors and single-stage detectors. Two-stage models first generate proposals that likely contain objects and then refine the box coordinates and perform classification. Notable two-stage methods after the R-CNN include Fast R-CNN [10], Faster R-CNN [29], and R-FCN [7]. In contrast, single-stage detectors directly output detection boxes from the features extracted by the backbone network. SSD [22] and the YOLO family [26, 27, 28] are popular architectures in this category.

Various works have attempted to improve the detection pipeline. [19] proposed feature pyramid networks, which aggregate features across scales and network depths. Mask R-CNN [12] incorporated a segmentation head to perform instance segmentation in parallel with detection. [8] introduced deformable convolution that replaces the fixed square receptive field of standard convolution by a learnable receptive field. Cascade R-CNN [4] stacks multiple detectors with increasing IoU thresholds to adapt to different IoU requirements at test time. RetinaNet [20] introduced the focal loss to address class imbalance and tackle hard examples.

3. Method

3.1. Decomposed Attention

For an input tensor $x$, it first passes through three separate functions $b, c, v$, which generate the basis $b(x)$, the coefficients $c(x)$, and the value $v(x)$, respectively. Then, a context extraction function $n$ takes in $b(x)$ and $v(x)$ and gives the attention context $n(b(x), v(x))$. Finally, an attention application function $m$ uses the coefficients and the context to produce the attended output. In summary, a mathematical representation of the DA module is

$$d(x) = m(c(x), n(b(x), v(x))).$$

In equation (1), the functions $c, b, v, n, m$ are abstract. In the baseline instantiation, $c, b, v$ are each a flattening operation followed by a pointwise linear layer. The outputs $c(x), b(x)$ have the same channel count. The context extraction function $n$ transposes $b(x)$, divides $v(x)$ by the size of the input\(^3\) for normalization, and multiplies them. The attention application function $m$ returns the matrix product of $c(x)$ and $n(b(x), v(x))$. A mathematical formulation of this instantiation is

$$d_{\text{baseline}}(x) = CX \left( (BX)^T \frac{VX}{s} \right),$$

for $X$ the flattened version of $x$, $s$ the size of $x$, and $C, B, V$ parameter matrices.

There are several aspects of the baseline instantiation that are flexible to change. Following subsections introduce a set of design variables that lead to different instantiations of DA.

3.2. Design Variables

3.2.1 Attention Normalization

The baseline instantiation implements attention normalization as the scaling operation on $v(x)$. This version is scaling normalization. Scaling normalization does not involve any non-linearities. Therefore, it is easy to implement and facilitates smooth flow of gradients.

An alternative implementation is softmax normalization. This version applies a channel-wise softmax on $b(x)$ and a location-wise softmax on $c(x)$, before they enter subsequent computation. Softmax normalization provides better interpretability. Each channel in $b(x)$ sums up to one and

\(^3\)The spatiotemporal size, i.e. $t$ for sequence features, $h \times w$ for 3D feature maps, and $h \times w \times d$ for 4D feature volumes.
represents a distribution of attention over the entire input. Similarly, the feature vector at each location in \( c(x) \) sums up to one and represents a distribution of attention over the basis masks.

### 3.2.2 Multi-Head Mechanism

The multi-head mechanism is a method to augment an attention module [33]. A DA module accepts a hyperparameter \( h \) as the number of heads. The channel depth of the outputs of \( c, b, v \) each reduces by a factor of \( h \). Then, \( h \) DA modules with identical structures but separate weights process the input in parallel. A concatenation along the channel dimension of their results makes the final output. The concatenation restores the channel count to the same as in a single-head DA module. Following is a formal representation of a multi-head DA module:

\[
d_{\text{multi}}(x) = (h-1) \bigoplus_{i=0}^{h-1} d_i(x),
\]

### 3.2.3 Other Design Choices

**Basis, coefficients, and value functions.** The baseline instantiation uses pointwise linear layers to implement \( b, c, v \). An alternative to pointwise layers is convolution. Functions \( b, c, v \) can instantiate as \( 3 \times 3 \) convolutional layers. Convolution with larger kernels or more complex modules are possible. However, their high costs defeat the purpose of DA. Therefore, the experiments did not test such instantiations of \( b, c, v \).

**Reprojection and skip connection.** A DA module can have an additional linear reprojection of the output. Such an operation allows changing the number of output channels. For instance, this feature enables the bottleneck structure, where \( v \) reduces the channel count and the reprojection restores it. This structure with appropriate hyperparameters further saves computation and memory. In addition, an optional skip connection might ease gradient flow and improve performance.

### 3.3. Complexity Analysis

This section analyzes the memory and computational complexities of DA and conventional attention modules. In the following analysis, \( b, k, m, c \) represent the channel counts of the basis, the key, the value, and the input, respectively; \( s \) stands for the spatiotemporal size of the input; and \( h \) denotes the number of attention heads.

As Table 1 shows, the memory and computational complexities of a DA module are both \( O(s) \), same as for a Res-Block and significantly less than the \( O(s^2) \) complexities of a conventional attention module. Figure 3 shows the calculation for the DA module.

![Diagram of DA module](Image)

**Figure 3. Architectural illustration of a DA module with annotations for resource usage.** \( M \) represents the memory consumption in number of floats. \( C \) is the computational consumption in MACC. \( \otimes \) and \( \oplus \) stand for matrix multiplication and element-wise addition, respectively. The top-left box shows the total resource usage of the module.

![Resource requirements](Image)

**Figure 4. Resource requirements under different input sizes.** The blue and red bars depict the resource requirements of DA and conventional attention modules, respectively. The calculation assumes 64 input channels. The figure is in log scale.

To exemplify the difference, Figure 4 compares the resource consumption of DA and conventional attention for different input sizes. Directly substituting the conventional attention module on the 64x64 feature map in SAGAN [38] yields a 17x saving of memory and 32x saving of computation. The gap widens rapidly with the increase of the input size. For a 256x256 feature map, a conventional attention module would require impractical amounts of memory (17.2 GB) and computation (412 GMACC). With the same input size, a DA module uses 260x less memory and 515x less computation, which are less than the consumption of a conventional attention module on a 64x64 input.

The difference is more prominent for videos. Replacing the conventional attention module on the tiny 28x28x4 feature volume in \( \text{res}_3 \) of the non-local I3D-ResNet-50 net-
work [35] results in 2x memory and computational saving. On a larger 64x64x32 feature volume, a DA module requires 32x less memory and 1025x less computation.

The multi-head mechanism further enlarges the efficiency disparity between DA and conventional attention. A conventional module with \( h \) heads generates \( h \) attention maps of size \( s \times s \) in a forward pass. When \( s \) is large, the multi-head architecture is prohibitively costly. In contrast, a multi-head DA module does not compute any \( s \times s \) attention map. Counter-intuitively, it demands less computation and memory with a larger \( h \). DA thereby enables the application of the multi-head mechanism on large inputs. Rows 1 and 2 of Table 1 show the different patterns of complexity change with the increase of \( h \).

Figure 5 provides a concrete comparison between conventional and DA Transformers. Transformers make extensive use of multi-head attention. The calculations follow the settings of BERT [9], which holds the state-of-the-art records in virtually all NLP tasks. As the figure demonstrates, when the length of the sequence increases, the advantage of the DA Transformer grows quadratically. On a hyper-long sequence with 8192 entries, the memory consumption of the conventional Transformer is impractical, while the demand of the DA Transformer is still manageable.

### Table 1. Comparison of resource usage of ResBlock, the DA module, and the conventional attention module.

| Module                  | ResBlock | DA Module | Conventional Attention |
|-------------------------|----------|-----------|-----------------------|
| Memory (floats)/Computation (MACC) | \( 5cs \) | \((2b + 3c)s + \frac{c}{K} \) | \((2k + 3c)s + hs^2 \) |
| Assuming \( b = k = \frac{s}{2}, h = \frac{c}{K} \) | \( 36c^2s + cs \) | \((6bc + c^2 + 2\frac{bc}{K})s \) | \((4kc + 2c^2 + c)s + (2hk + 2c)s^2 \) |
| Complexities in \( s \) | \( O(s) \) | \( O(s) \) | \( O(s^2) \) |

Table 2. Experiments for attention normalization methods. All experiments inserted a single DA module after res3 and used \( b = 64, h = 1 \).

### 4. Experimental Results

#### 4.1. MS-COCO Task Suite

##### 4.1.1 Experimental Setup

This section evaluates the proposed method on the MS-COCO 2017 dataset for object detection and instance segmentation. The baseline is a ResNet-50 Mask R-CNN with a 5-level feature pyramid [19]. All models trained for 24 epochs on 32 NVIDIA GTX 1080 Ti GPUs. The batch size is 64 for ResNet-50 backbones and 32 for ResNet-101 and ResNeXt-101. The learning rate is 1.25e-4 at the beginning of training and drops by a factor of 10 at the start of the 18th and 21st epochs. All experiments used the baseline instantiation with skip connection and set the basis dimensionality to 64 unless otherwise specified.

##### 4.1.2 Attention Normalization

These experiments empirically validated the necessity of attention normalization and compared the two methods Section 3.2.1 specified, namely scaling and softmax normalization. Table 2 reports the experimental outcomes. The results demonstrate that while attention normalization is critical for the training of DA-augmented models. The effectiveness does not depend on the specific normalization technique. Following [35], subsequent experiments used softmax normalization.

#### 4.1.3 Basis, Coefficients, and Value Functions

This section presents experiments contrasting different basis, coefficients, and value functions. Table 3 shows the
results using pointwise linear layers and 3x3 convolutions. As the table shows, 3x3 convolution gives slightly better performance. However, it also results in significantly more computational costs. Therefore, subsequent experiments continued to use pointwise linear layers.

4.1.4 Number of Attention Heads

These experiments explored the effect of the number of attention heads on the performance of DA. Table 4 reports the experimental outcomes. The results demonstrate, in contrary to the findings in [33], the multi-head mechanism did not improvement performance. We hypothesize that this is due to the small channel count of the basis and coefficient matrices in our settings. Further splitting the 64 channels might have made the dimensionality of the basis not enough to capture sufficient semantic information. Therefore, we suppose the multi-head mechanism will demonstrate its effectiveness with bases of higher dimensionalities.

4.1.5 Position of Insertion

Table 5 shows experiments contrasting models with DA modules inserted at different positions. When adding a single module, inserting it after $\text{res}_3$ lead to the largest improvement, 0.8 in box AP and 0.9 in mask AP. The high resolution and the small effective receptive field might explain the effectiveness of self-attention there. The insertion position $\text{res}_3$ is the only that lead to degraded performance. A hypothesis is that the low-resolution and large gap between the feature map channel count (2048) and the chosen basis dimensionality (64) might have been the cause. A defect in the implementation caused the experiments for $\text{res}_1$ and $\text{res}_2$ to fail, so the results are not available for report in this paper.

Among all FPN levels, insertion after $\text{fpn}_1$ resulted in the best performance. The rightmost column of Table 5 lists the sizes of the feature maps in $\text{res}_3$, $\text{res}_4$, and all FPN levels. Contrasting the column with other columns shows a strong positive correlation between the size of the feature map and the performance gain when inserting a DA module there. This confirmed the theoretical analysis in Section 1 that high resolution magnifies the advantage of self-attention over convolution. Note that the resource-efficiency of DA enabled these explorations. It allowed the flexibility of inserting the module at different positions of the model. In contrast, the high resource demands of conventional attention modules confines the insertions to the low-resolution parts of a model.

Subsequent experiments explored adding multiple DA modules to a network. Incorporating a DA module at each FPN level gave a 1.2 increase in box AP and a 1.1 increase in mask AP. Adding a DA block to $\text{res}_3$, $\text{res}_4$, and every FPN level resulted in an improvement of 1.8 box AP and 1.6 mask AP. The results back the hypothesis in Section 1 that a core advantage of self-attention is the ability to model multi-hop long-range dependencies.

Interestingly, although a single insertion of a DA module at $\text{res}_4$ decreased performance, removing the module at $\text{res}_4$ from the $\text{fpn}_{all}\&\text{res}_3$ scheme caused a significant decrease in performance. This result further reinforces the importance of multi-hop dependency modeling, as even DA modules which cannot improve performance independently can make contributions when collaborating with other DA modules. The low complexities of the DA module enable multiple insertion and realize this previously hypothetical advantage of self-attention.

4.1.6 Dimensionality of the Basis

These experiments tested the impact of the basis dimensionality on the effect of DA. As in Table 6, decreasing the basis dimensionality from 128 to 32 caused minimal performance change. This result reinforces the hypothesis in Section 1 that most masks can be expressed as linear combinations of a limited set of basis masks. Therefore, when using DA modules, researchers can reduce the dimensionality of the
### Experiments with different basis dimensionalities

All experiments used the fpn$_{all}$&res$_{3,4}$ insertion scheme, softmax normalization, and $h = 1$.

| $b$   | Box AP | Mask AP |
|-------|--------|---------|
| 32    | 40.4   | 36.1    |
| 64    | 40.6   | 36.2    |
| 128   | 40.3   | 36.1    |

Table 6.

### Experiments with different backbone architectures

The +DA variants used the fpn$_{all}$&res$_{3,4}$ insertion scheme, softmax normalization, and $b = 64, h = 1$.

| Model          | Box AP | Mask AP |
|----------------|--------|---------|
| ResNet-50      | 39.4   | 35.1    |
| +DA            | +1.8   | +1.6    |
| ResNet-101     | 41.3   | 36.6    |
| +DA            | +1.8   | +1.3    |
| ResNeXt-101    | 43.5   | 38.5    |
| +DA            | +1.4   | +1.0    |

Table 7.

basis to further save resources. Combining this result with the findings in Section 4.1.5 leads to the conclusion that it is more effective to distribute computational and memory resources across multiple DA modules spread in the network than to concentrate the resources into a single DA module.

### 4.1.7 Backbone Architecture

This section reports experiments with different backbone networks. Table 7 reports the results using three selected architectures, ResNet-50, ResNet-101 and ResNeXt-101. The significant and robust performance gains across all three backbone architectures demonstrate the generalizability of DA.

### 4.1.8 Comparison with Conventional Attention

These experiments empirically compared DA and conventional attention. Table 8 reports the results. As rows res$_3$ and fpn$_3$ show, inserting a DA module or a conventional attention module at the same position in a network has nearly identical effects on the performance. However, at positions where the input in large, inserting a conventional attention module lead to out-of-memory errors, while inserting a DA module improved performance significantly and consumed little additional resources. Consequently, using the best insertion schemes for DA and conventional attention respectively, DA outperforms conventional attention by 0.9 box AP and 0.8 mask AP. The best insertion scheme for DA was fpn$_{all}$&res$_{3,4}$, the best among all schemes that experiments in Table 5 tried. Another set of experiments tried all the

---

4Table 8 did not show input sizes for space constraints. Refer to Table 5 for the input size for each layer.

| Position(s) | DA         | Conventional |
|-------------|------------|--------------|
| res$_3$     | 40.2/36.0  | 40.3/35.9    |
| fpn$_1$     | 39.9/35.8  | OOM          |
| fpn$_2$     | 39.7/35.7  | OOM          |
| fpn$_3$     | 39.7/35.5  | 39.8/35.5    |
| Best        | 41.2/36.7  | 40.3/35.9    |

Table 8.

### Comparison between DA and conventional attention

Results in the table are in the format box AP/mask AP. All experiments used softmax normalization and $b = 64, h = 1$. The best insertion schemes for DA and conventional attention are fpn$_{all}$&res$_{3,4}$ and res$_3$, respectively. Details on the way the experiments searched for the best insertion schemes are in Section 4.1.8.

### 4.1.9 Comparison with the State-of-the-Art

Table 9 compares DA-augmented detectors with other state-of-the-art methods on the MS-COCO 2017 dataset. DA Mask R-CNN with a ResNeXt-101 backbone set a new state-of-the-art. The version with a ResNet-101 backbone also outperformed all other ResNet-101-based models.

| Model          | Backbone     | AP  |
|----------------|--------------|-----|
| Single-model methods |
| TDM Faster R-CNN [30] | Inc.-ResNet-v2 | 36.8 |
| Mask R-CNN [12] | ResNeXt-101 | 39.8 |
| Soft-NMS [2] | Aligned-Inc.-ResNet | 40.9 |
| LH R-CNN [17] | ResNet-101 | 41.5 |
| Fitness-NMS [32] | ResNet-101 | 41.8 |
| DA Mask R-CNN | ResNet-101 | 43.1 |
| DA Mask R-CNN | ResNeXt-101 | 45.9 |
| Non-single-model method |
| Cascade R-CNN [4] | ResNet-101 | 42.8 |

Table 9.

### Comparison with state-of-the-art on MS-COCO 2017

Inc.-ResNet-v2 and Aligned-Inc.-ResNet represent Inception-ResNet-v2 and Aligned-Inception-ResNet, respectively.

settings in Table 5 using conventional attention. The best scheme was res$_3$.

### 4.1.10 Visualization

Figure 6 shows visualization of the basis masks generated for various examples by the DA module after fpn$_1$ in the fpn$_{all}$&res$_{3,4}$ model. The figure illustrates 3 set of basis masks each with a distinct and meaningful focus. Column 2 tends to capture the foreground, column 3 tends to capture the core parts of objects, and column 4 tends to capture the peripheral of objects. The semantic distinctiveness of each set of basis masks supports the analysis in Section 1 that the attention masks are linear combinations of a set of basis masks each focusing on a semantically significant area.
Figure 6. Visualization of basis masks. The left-most column displays 4 images from the MS-COCO 2017 dataset. The other three columns show three of the corresponding basis masks extracted from the DA module after fpn1 for each respective example.

Table 10. Experiments on the Scene Flow dataset. DA-PSMNet is the proposed approach. OOM indicates the model could not fit into the memory.

| Model                  | EPE   |
|------------------------|-------|
| PSMNet (original)      | 1.09  |
| PSMNet (baseline)      | 0.513 |
| DA-PSMNet              | 0.477 |
| Nonlocal-PSMNet        | OOM   |

Table 11. Comparison with state-of-the-art on the Scene Flow dataset. EPE stands for end-point error and is lower the better. The table rounded EPE for DA-PSMNet to the second decimal place following the format of previous works.

| Position(s) | Accuracy (%) | Improvement (%) |
|-------------|--------------|-----------------|
| Baseline    | 76.052/92.952 | 0.000/0.000     |
| res1        | 76.932/93.252 | 0.880/0.300     |
| res2        | 76.532/93.274 | 0.480/0.322     |
| res1, res2  | 77.312/93.650 | 1.260/0.698     |

Table 12. Experiments on ImageNet. All results are in the form top-1/top-5. All experiments used single-crop testing.

### 4.2. Stereo Depth Estimation

This section will present the experimental results of DA-augmented models for stereo depth estimation. The experiments used the Scene Flow dataset. The proposed model set a new state-of-the-art on the dataset.

**Experimental setup.** The baseline is the state-of-the-art published model, PSMNet [5]. Some exploratory experiments suggested that the hyperparameter settings had considerable room for improvement. The optimal settings of hyperparameters use a batch size of 24, a learning rate of 2e-3, and 100 training epochs. Other hyperparameters remain the same as in [5]. Optimizing the hyperparameters lead to a significant increase in performance, reducing the previous state-of-the-art end-point error (EPE) from 1.09 to 0.513.

**Network architecture.** The experiment inserted a single DA module after the final residual block in the monocular backbone. The team also attempted to use conventional attention. However, inserting a single conventional attention module anywhere in the network leads to out-of-memory errors, even when using a batch size of 1.

**Experimental results.** As Table 10 shows, the proposed DA-PSMNet demonstrated significant improvement over the highly competitive baseline, which already surpassed the former state-of-the-art by a substantial margin.

### 4.3. Image Classification

This section presents the experimental results of DA-augmented models on the ImageNet dataset for image classification. The baseline of the experiments is ResNet-50 [13]. As in Table 12, insertions at res1 and res2 both lead to significant improvements. Inserting simultaneously to both positions resulted in the largest gain of performance.

### 5. Conclusion

This paper has presented the decomposed attention mechanism, a quadratically more memory- and computationally-efficient algorithm for self-attention. By dramatically reducing the resource usage of self-attention, DA enables a large number of new use cases of the mechanism, particularly in domains with tight resource constraints or large inputs.

The experiments verified its effectiveness on four distinct tasks, object detection, instance segmentation, stereo depth estimation, and image classification. DA brought significant improvement for each task. On object detection and stereo depth estimation, DA-augmented models have set new states-of-the-art. Besides the tasks this paper evaluated DA on, it has promising potential in other fields where self-attention has demonstrated effectiveness. These fields include generative adversarial image modeling [38, 3] and most tasks in natural language processing [33, 24, 9, 25]. Future plans include generalizing DA to these fields, as well as other fields where the prohibitive costs have been preventing the application of self-attention.
References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In ICLR, 2015. 1, 2
[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nmsimproving object detection with one line of code. In ICCV, 2017. 7
[3] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018. 1, 2, 3, 8
[4] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In CVPR, 2018. 3, 7
[5] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In CVPR, 2018. 8
[6] X. Cheng, P. Wang, and R. Yang. Learning depth with convolutional spatial propagation network. arXiv preprint arXiv:1810.02695, 2018. 8
[7] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In NIPS, 2016. 3
[8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In ICCV, 2017. 3
[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1, 2, 5, 8
[10] R. Girshick. Fast r-cnn. In ICCV, 2015. 3
[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 3
[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In ICCV, 2017. 3, 7
[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016. 8
[14] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In CVPR, 2018. 3
[15] M. Kampffmeyer, N. Dong, X. Liang, Y. Zhang, and E. P. Xing. Connet: A long-range relation-aware pixel-connectivity network for salient segmentation. arXiv preprint arXiv:1804.07836, 2018. 3
[16] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In ACM Multimedia Conference, 2018. 1, 3
[17] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264, 2017. 7
[18] X. Liao, L. He, and Z. Yang. Video-based person re-identification via 3d convolutional networks and non-local attention. arXiv preprint arXiv:1807.05073, 2018. 1, 3
[19] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In CVPR, 2017. 3, 5
[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In ICCV, 2017. 3
[21] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. arXiv preprint arXiv:1806.02919, 2018. 3
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In ECCV, 2016. 3
[23] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In ICCV Workshops, 2017. 8
[24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. OpenAI Blog, 2018. 1, 2, 8
[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 2019. 1, 8
[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016. 3
[27] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In CVPR, 2017. 3
[28] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018. 3
[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015. 3
[30] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. arXiv preprint arXiv:1612.06851, 2016. 7
[31] X. Song, X. Zhao, H. Hu, and L. Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In ACCV, 2018. 8
[32] L. Tychsen-Smith and L. Petersson. Improving object localization with fitness nms and bounded iou loss. In CVPR, 2018. 7
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In NIPS, 2017. 1, 2, 4, 6, 8
[34] 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
[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In CVPR, 2018. 1, 2, 5
[36] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cham: Convolutional block attention module. In ECCV, 2018. 3
[37] W. Xue, G. Brahm, S. Leung, O. Shmuilovich, and S. Li. Cardiac motion scoring with segment-and subject-level non-local modeling. In MICCAI, 2018. 3
[38] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318, 2018. 1, 2, 3, 4, 8