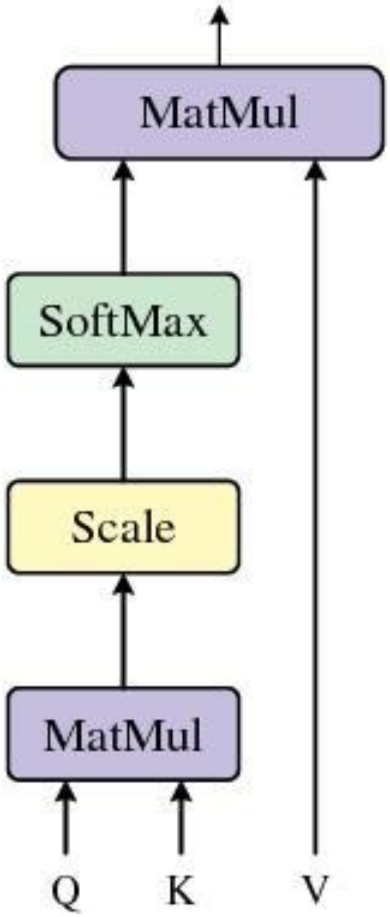
Polarized Self-Attention

Towards High-quality Pixel-wise Regression



$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Attention is all you need

Polarized Self Attention (PSA)

- Plug & Play module
- Boosts state of the arts by 1-2 points on
  2D pose estimation & semantic segmentation

Idea of PSA:

- Polarized filtering: Keep high res both in channel & spatial dim.,
  while completely collapsing their counterpart dim.
- Enhancement: Non linearity that fits output dist.
  of typical fine grained regression

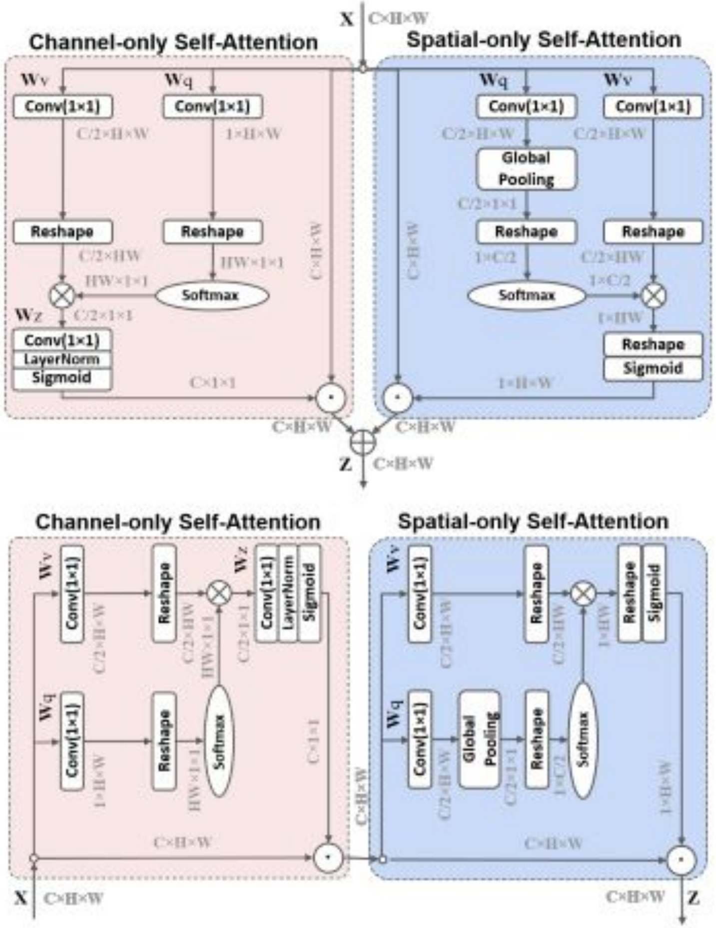
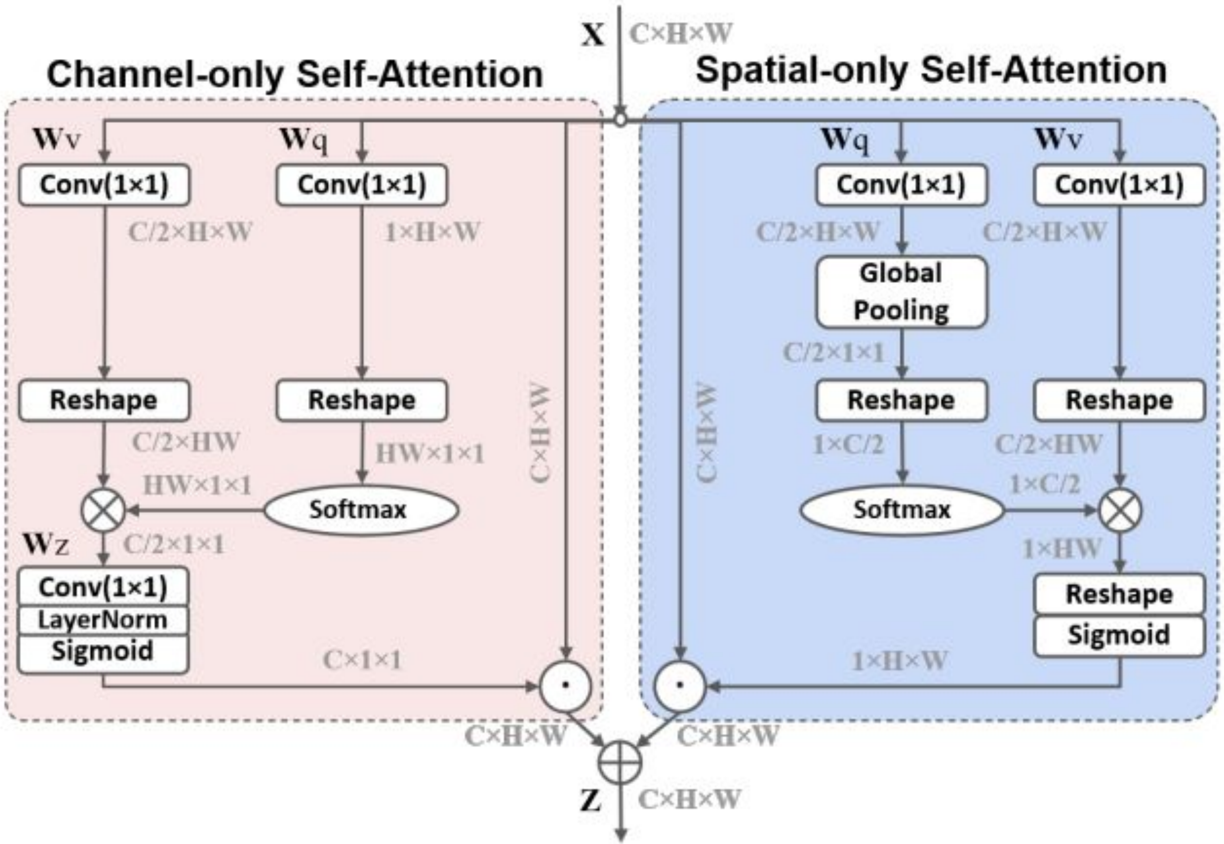


Figure 2. The Polarized Self-Attention (PSA) block under **(upper)** the parallel layout, and **(lower)** the sequential layout.

PS


X
C×H×W
Channel-only Self-Attention
Spatial-only Self-Attention
Wv
Wq
Conv(1×1)
Conv(1×1)
C/2×H×W
1×H×W
Reshape
Reshape
C/2×HW
HW×1×1
HW×1×1
Softmax
Wz
C/2×1×1
Conv(1×1)
LayerNorm
Sigmoid
C×1×1
C×H×W
Wq
Wv
Conv(1×1)
Conv(1×1)
C/2×H×W
C/2×H×W
Global Pooling
C/2×1×1
Reshape
Reshape
1×C/2
C/2×HW
1×C/2
Softmax
1×HW
Reshape
Sigmoid
1×H×W
C×H×W
C×H×W
C×H×W
Z
C×H×W

PS

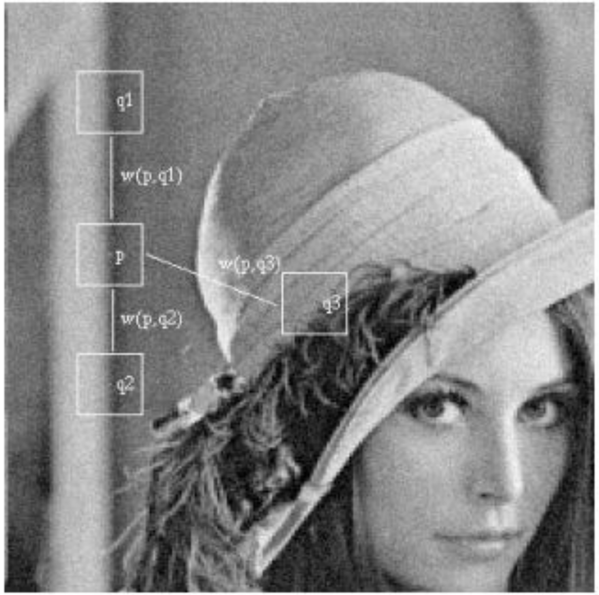


Figure 1. Scheme of NL-means strategy. Similar pixel neighborhoods give a large weight, w(p,q1) and w(p,q2), while much different neighborhoods give a small weight w(p,q3).

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j).$$

A non-local algorithm for image

Non-local Neural

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j).$$

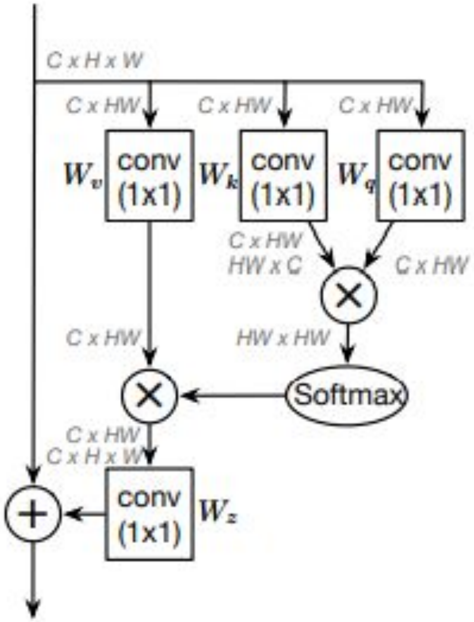
Embedded Gaussian. A simple extension of the Gaussian function is to compute similarity in an embedding space. In this paper we consider:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}. \tag{3}$$

Here $\theta(\mathbf{x}_i) = W_\theta \mathbf{x}_i$ and $\phi(\mathbf{x}_j) = W_\phi \mathbf{x}_j$ are two embeddings. As above, we set $\mathcal{C}(\mathbf{x}) = \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j)$.

We note that *the self-attention module [49] recently presented for machine translation is a special case of non-local operations in the embedded Gaussian version.* This can be seen from the fact that for a given i, $\frac{1}{\mathcal{C}(\mathbf{x})} f(\mathbf{x}_i, \mathbf{x}_j)$ becomes the *softmax* computation along the dimension j.

Non-local Neural



(a) NL block

$$\mathbf{z}_i = W_z \mathbf{y}_i + \mathbf{x}_i,$$

$$\mathbf{z}_i = \mathbf{x}_i + W_z \sum_{j=1}^{N_p} \frac{f\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\mathcal{C}\left(\mathbf{x}\right)} \left(W_v \cdot \mathbf{x}_j\right),$$

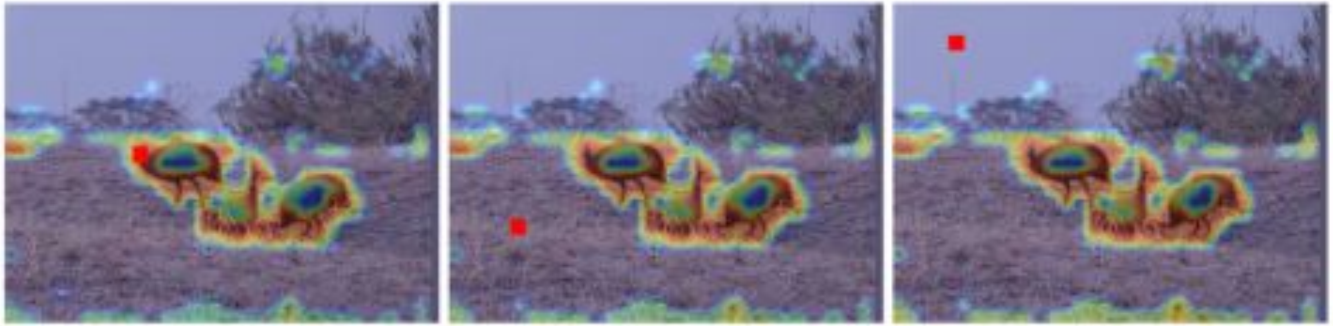
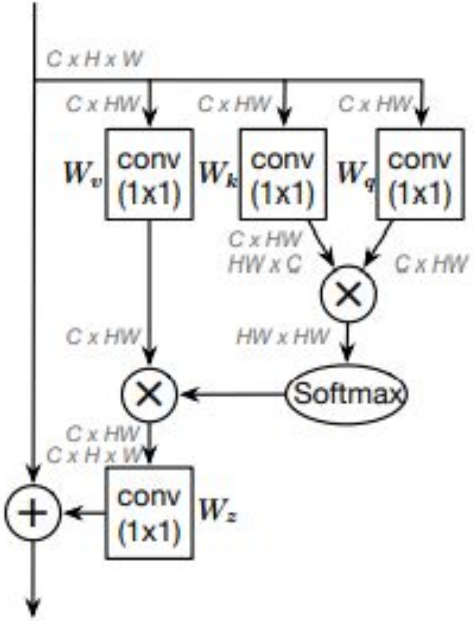


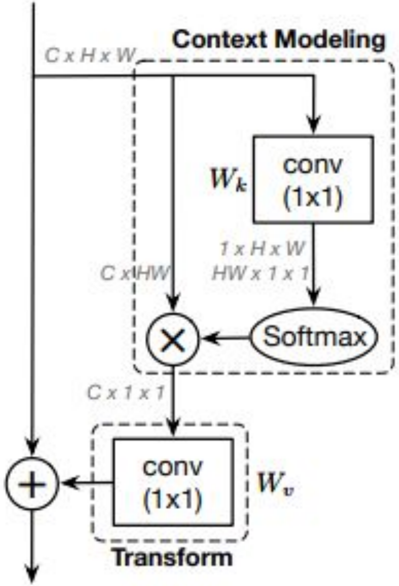
Figure 1: Visualization of attention maps (heatmaps) for different query positions (red points) in a non-local block on COCO object detection. The three attention maps are all almost the same. More examples are in Figure 2.

$$\mathbf{z}_i = \mathbf{x}_i + W_z \sum_{j=1}^{N_p} \frac{f\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\mathcal{C}\left(\mathbf{x}\right)} \left(W_v \cdot \mathbf{x}_j\right), \qquad \mathbf{z}_i = \mathbf{x}_i + \sum_{j=1}^{N_p} \frac{\exp\left(W_k \mathbf{x}_j\right)}{\sum_{m=1}^{N_p} \exp\left(W_k \mathbf{x}_m\right)} \left(W_v \cdot \mathbf{x}_j\right),$$

GCNet: Non-local Networks Meet Squeeze-Excitation Networks and



(a) NL block

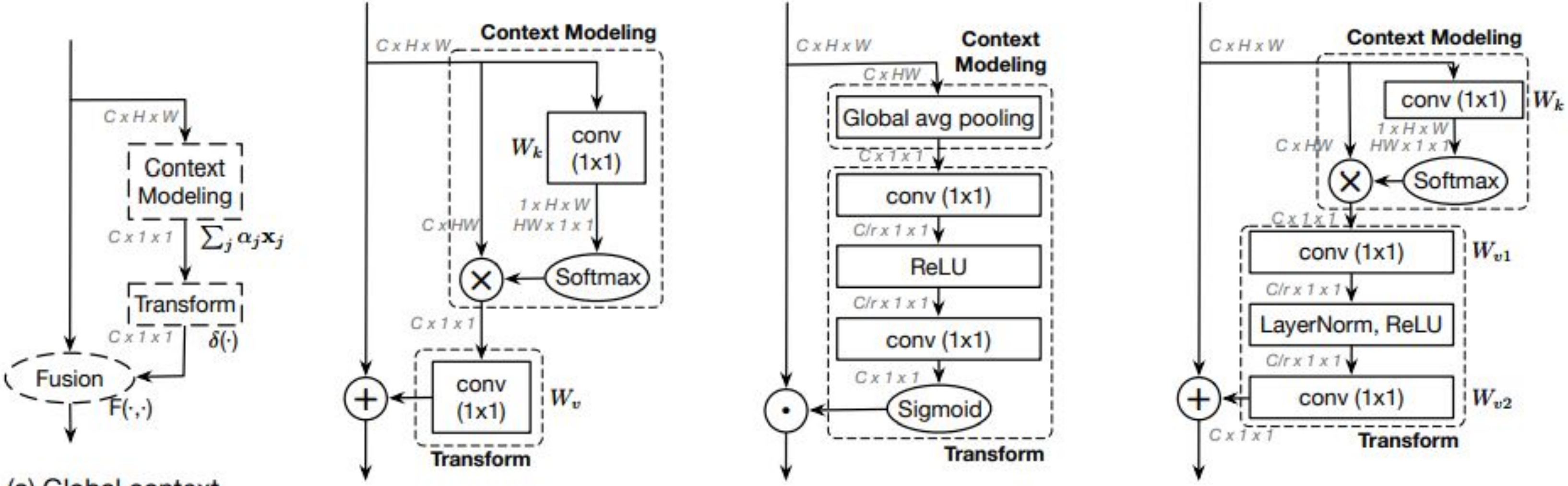


(b) Simplified NL block (Eqn 3)

$$\mathbf{z}_i = \mathbf{x}_i + \sum_{j=1}^{N_p} \frac{\exp\left(W_k \mathbf{x}_j\right)}{\sum_{m=1}^{N_p} \exp\left(W_k \mathbf{x}_m\right)} \left(W_v \cdot \mathbf{x}_j\right),$$

$$\mathbf{z}_i = \mathbf{x}_i + W_v \sum_{j=1}^{N_p} \frac{\exp\left(W_k \mathbf{x}_j\right)}{\sum_{m=1}^{N_p} \exp\left(W_k \mathbf{x}_m\right)} \mathbf{x}_j.$$

GCNet: Non-local Networks Meet Squeeze-Excitation Networks and



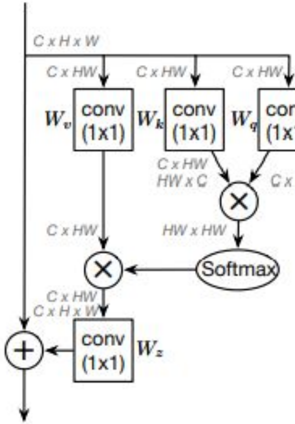
(a) Global context modeling framework

(b) Simplified NL block (Eqn 3)

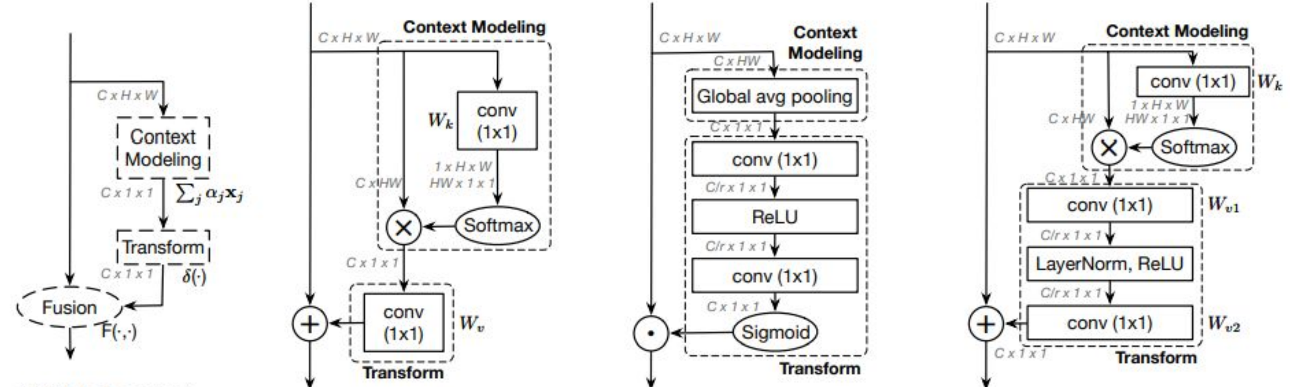
(c) SE block

(d) Global context (GC) block

GCNet: Non-local Networks Meet Squeeze-Excitation Networks and



(a) NL block

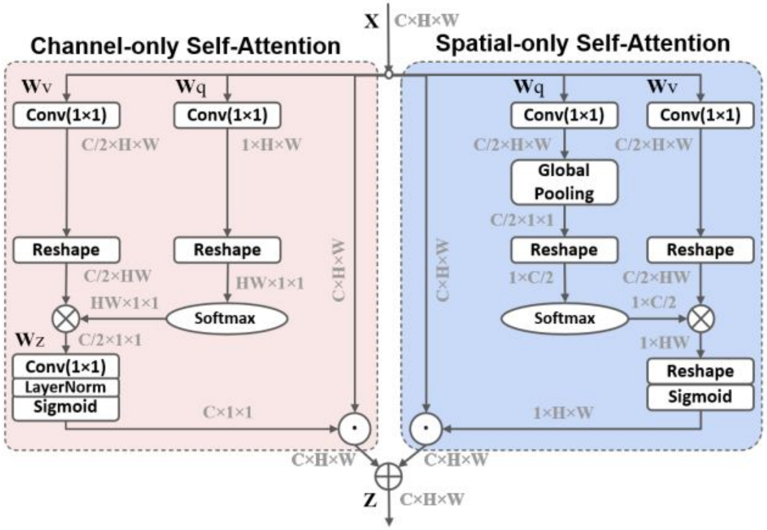


(a) Global context modeling framework

(b) Simplified NL block (Eqn 3)

(c) SE block

(d) Global context (GC) block



Method	ch. resolution	sp. resolution	non-linearity	complexity $O(\cdot)$
NL[47]	C	$[W, H]$	SM	$C^2WH + CW^2H^2$
GC [3]	$C/4$	-	SM+ReLU	CWH
SE [19]	$C/4$	-	ReLU+SD	CWH
CBAM [48]	$C/16$	$[W, H]$	SD	CWH
DA [14]	$C/8$	$[W, H]$	SM	$C^2WH + CW^2H^2$
EA [39]	d_k ($\ll C$)	d_v ($\ll min(W, H)$)	SM	CWH
PSA(ours)	$C/2$	$[W, H]$	SM+SD	CWH

Table 1. Re-visit critical design aspects in existing attention blocks. All the attention blocks are compared in their top-performance configurations. SM: SoftMax, SD: Sigmoid. Complexity is estimated assuming $C < WH$.

Method	Backbone	mIoU ↑	Flops	mPara
DeepLabV3Plus [4]	MobileNet	71.1	16.9G	5.22M
+PSA	MobileNet	**73.7(+2.6)**	17.1G	5.22M
DeepLabV3Plus [4]	Res50	77.2	62.5G	39.8M
+PSA	Res50	**79.0(+1.8)**	65.2G	42.3M
DeepLabV3Plus [4]	Res101	78.3	83.2G	58.8M
+PSA	Res101	**80.3(+2.0)**	87.7G	63.5M

Table 3. PSA vs. Baselines for semantic segmentation on the Pascal VOC2012 Aug database.

Method	Backbone	ImageNet Pretrain	AP ↑	AP_{50}↑	AP_{75}↑	AP_M ↑	AP_L↑	AR ↑	Flops	mPara
Simple-Baseline [51]	Res50	Y	72.2	89.3	78.9	68.1	79.7	77.6	20.0G	34.0M
+PSA	Res50	N	**76.5(+4.3)**	**93.6**	**83.6**	**73.2**	**81.0**	**79.0**	20.9G	36.1M
Simple-Baseline [51]	Res152	Y	74.3	89.6	81.1	70.5	81.6	79.7	35.3G	68.6M
+PSA	Res152	N	**78.0(+3.7)**	**93.6**	**84.8**	**75.2**	**82.3**	**80.5**	37.5G	75.2M
HRNet [40]	HRNet-W32	Y	75.8	90.6	82.5	72.0	82.7	80.9	16.0G	28.5M
+PSA	HRNet-W32	Y	**78.7(+2.9)**	**93.6**	**85.9**	**75.6**	**83.5**	**81.1**	17.1G	31.4M
HRNet [40]	HRNet-W48	Y	76.3	90.8	82.9	72.3	83.4	81.2	32.9G	63.6M
+PSA	HRNet-W48	Y	**78.9(+2.6)**	**93.6**	**85.7**	**75.8**	**83.8**	**81.4**	35.2G	70.0M

Table 2. PSA vs. Baselines for top-down human pose estimation on the MS-COCO val2017 dataset. All results were computed with an human detector [51] of 56.4 AP on COCO val2017 dataset. All detected human image patches were resized to 384×288.

PS

Method	Backbone	mIoU	iIoU cla.	IoU cat.	iIoU cat.
GridNet [13]	-	69.5	44.1	87.9	71.1
LRR-4x	-	69.7	48.0	88.2	74.7
DeepLab [4]	D-ResNet-101	70.4	42.6	86.4	67.7
LC	-	71.1	-	-	-
Piecewise [27]	VGG-16	71.6	51.7	87.3	74.1
FRRN [36]	-	71.8	45.5	88.9	75.1
RefineNet [26]	ResNet-101	73.6	47.2	87.9	70.6
PEARL [23]	D-ResNet-101	75.4	51.6	89.2	75.1
DSSPN [25]	D-ResNet-101	76.6	56.2	89.6	77.8
LKM [34]	ResNet-152	76.9	-	-	-
DUC-HDC [45]	-	77.6	53.6	90.1	75.2
SAC [58]	D-ResNet-101	78.1	-	-	-
DepthSeg [24]	D-ResNet-101	78.2	-	-	-
ResNet38 [49]	WResNet-38	78.4	59.1	90.9	78.1
BiSeNet [53]	ResNet-101	78.9	-	-	-
DFN [54]	ResNet-101	79.3	-	-	-
PSANet [61]	D-ResNet-101	80.1	-	-	-
PADNet [52]	D-ResNet-101	80.3	58.8	90.8	78.5
CFNet [57]	D-ResNet-101	79.6	-	-	-
Auto-DeepLab [30]	-	80.4	-	-	-
DenseASPP [60]	WDenseNet-161	80.6	59.1	90.9	78.1
SVCNet [11]	ResNet-101	81.0	-	-	-
ANN [65]	D-ResNet-101	81.3	-	-	-
CCNet [22]	D-ResNet-101	81.4	-	-	-
DANet [14]	D-ResNet-101	81.5	-	-	-
HRNetV2 [44]	HRNetV2-W48	81.6	61.8	92.1	82.2
HRNetV2+OCR [55]	HRNetV2-W48	84.9	-	-	-
HRNetV2+OCR(MA) [41] (***Strong Baseline***)	HRNetV2-W48	85.4	-	-	-
Ours					
HRNetV2-OCR+PSA(p)	HRNetV2-W48	**86.95**	**71.6**	**92.8**	**85.0**
HRNetV2-OCR+PSA(s)	HRNetV2-W48	**86.72**	**71.3**	92.3	82.8

Table 5. Comparison with State-of-the-Art semantic segmentation approaches on the Cityscapes validation set.

Method	Backbone	Input Size	AP	AP_{50}	AP_{75}	AP_{M}	AP_{L}	AR	Flops	mPara
8-stage Hourglass [32]	8-stage Hourglass	256 × 192	66.9	-	-	-	-	-	14.3G	25.1M
CPN [5]	ResNet50	256 × 192	68.6	-	-	-	-	-	6.2G	27.0M
CPN + OHKM [5]	ResNet50	256 × 192	69.4	-	-	-	-	-	6.2G	27.0M
SimpleBaseline [51]	ResNet50	256 × 192	70.4	88.6	78.3	67.1	77.2	76.3	8.90G	34.0M
SimpleBaseline [51]	ResNet101	256 × 192	71.4	89.3	79.3	68.1	78.1	77.1	12.4G	53.0M
SimpleBaseline [51]	ResNet152	256 × 192	72.0	89.3	79.8	68.7	78.9	77.8	15.7G	72.0M
HRNet-W32 [40]	HRNet	256 × 192	74.4	90.5	81.9	70.8	81.0	78.9	7.10G	28.9M
HRNet-W48 [40]	HRNet	256 × 192	75.1	90.6	82.2	71.5	81.8	80.4	14.6G	63.6M
Dark-Pose [56]	HRNet-W32	256 × 192	75.6	90.5	82.1	71.8	82.8	80.8	7.1G	28.5M
UDP-Pose [21]	HRNet-W48	256 × 192	77.2	91.8	83.7	73.8	83.7	82.0	14.7G	63.8M
SimpleBaseline [51]	ResNet152	384 × 288	74.3	89.6	81.1	70.5	79.7	79.7	35.6G	68.6M
HRNet-W32 [40]	HRNet	384 × 288	75.8	90.6	82.7	71.9	82.8	81.0	16.0G	28.5M
HRNet-W48 [40]	HRNet	384 × 288	76.3	90.8	82.9	72.3	83.4	81.2	32.9G	63.6M
Dark-Pose [56]	HRNet-W48	384 × 288	76.8	90.6	83.2	72.8	84.0	81.7	32.9G	63.6M
UDP-Pose [21]	HRNet-W48	384 × 288	76.2	92.5	83.6	72.5	82.4	81.1	33.0G	63.8M
UDP-Pose [21] (***Strong Baseline***)	HRNet-W48	384 × 288	77.8	92.0	84.3	74.2	**84.5**	**82.5**	33.0G	63.8M
Ours										
UDP-Pose-PSA(p)	HRNet-W48	256 × 192	78.9	**93.6**	**85.8**	**76.1**	83.6	81.4	15.7G	70.1M
UDP-Pose-PSA(p)	HRNet-W48	384 × 288	**79.5**	**93.6**	**85.9**	**76.3**	**84.3**	81.9	35.4G	70.1M
UDP-Pose-PSA(s)	HRNet-W48	384 × 288	**79.4**	**93.6**	**85.8**	**76.1**	84.1	81.7	35.4G	69.1M

Table 4. Comparison with State-of-the-Art top-down 2D pose estimation approaches on the MS-COCO keypoint testdev set. Note that only [21]***Strong Baseline*** used extra training data.

PS

Credits & Further Reading:

1. [2107.00782] Polarized Self-Attention: Towards High-quality Pixel-wise Regression (arxiv.org)

2. [1904.11492] GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond (arxiv.org)

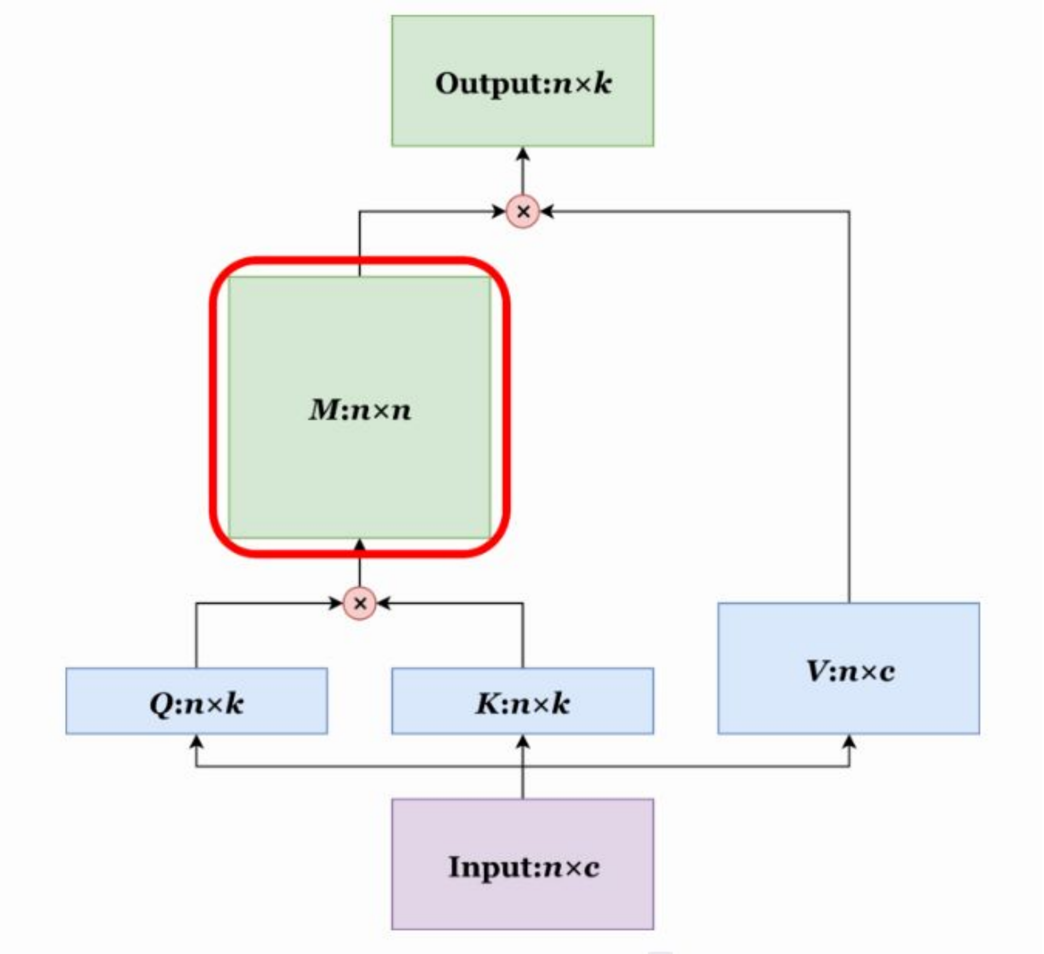
3. [1711.07971] Non-local Neural Networks (arxiv.org)

4. [1709.01507] Squeeze-and-Excitation Networks (arxiv.org)

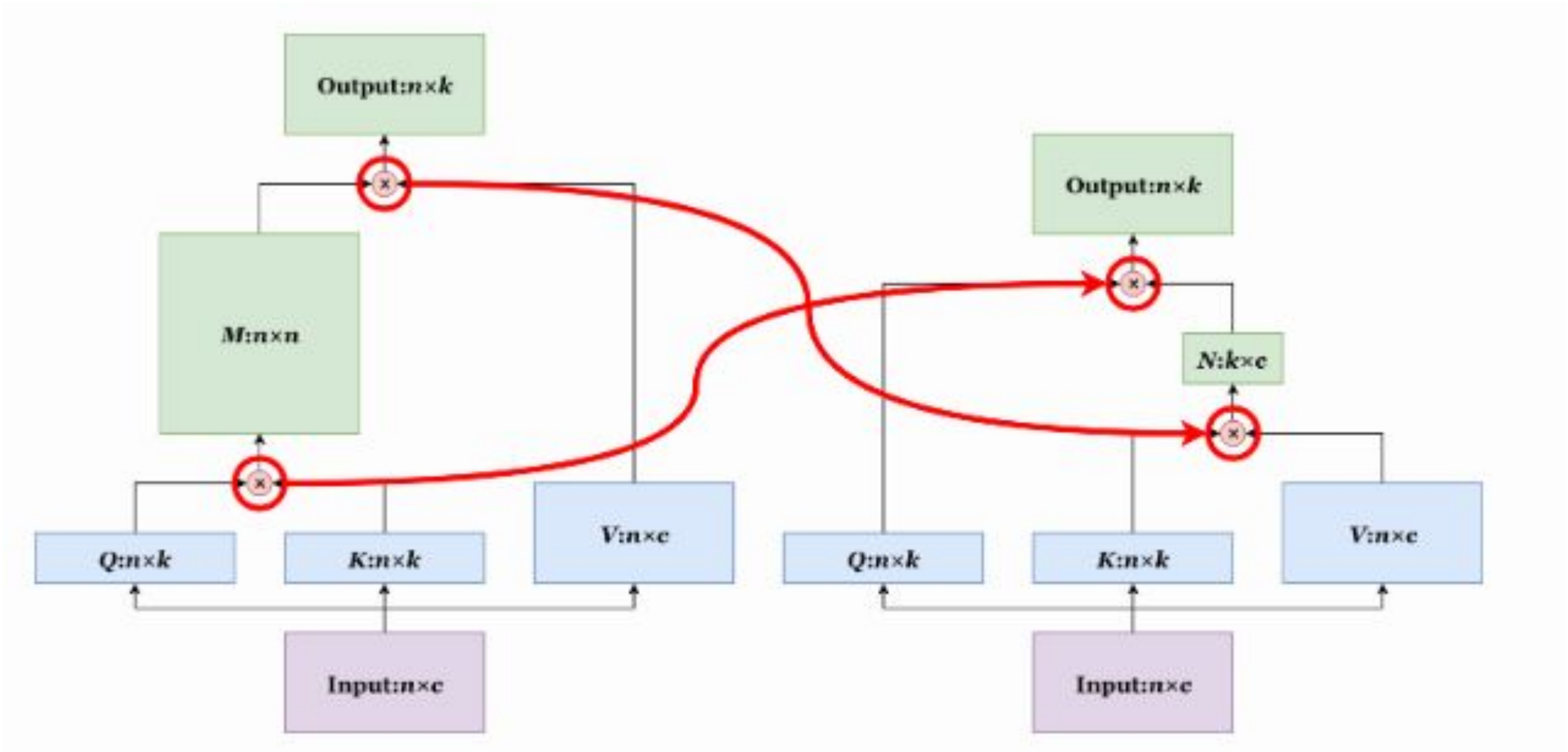
5. [1812.01243] Efficient Attention: Attention with Linear Complexities (arxiv.org)

6. [1809.02983] Dual Attention Network for Scene Segmentation (arxiv.org)

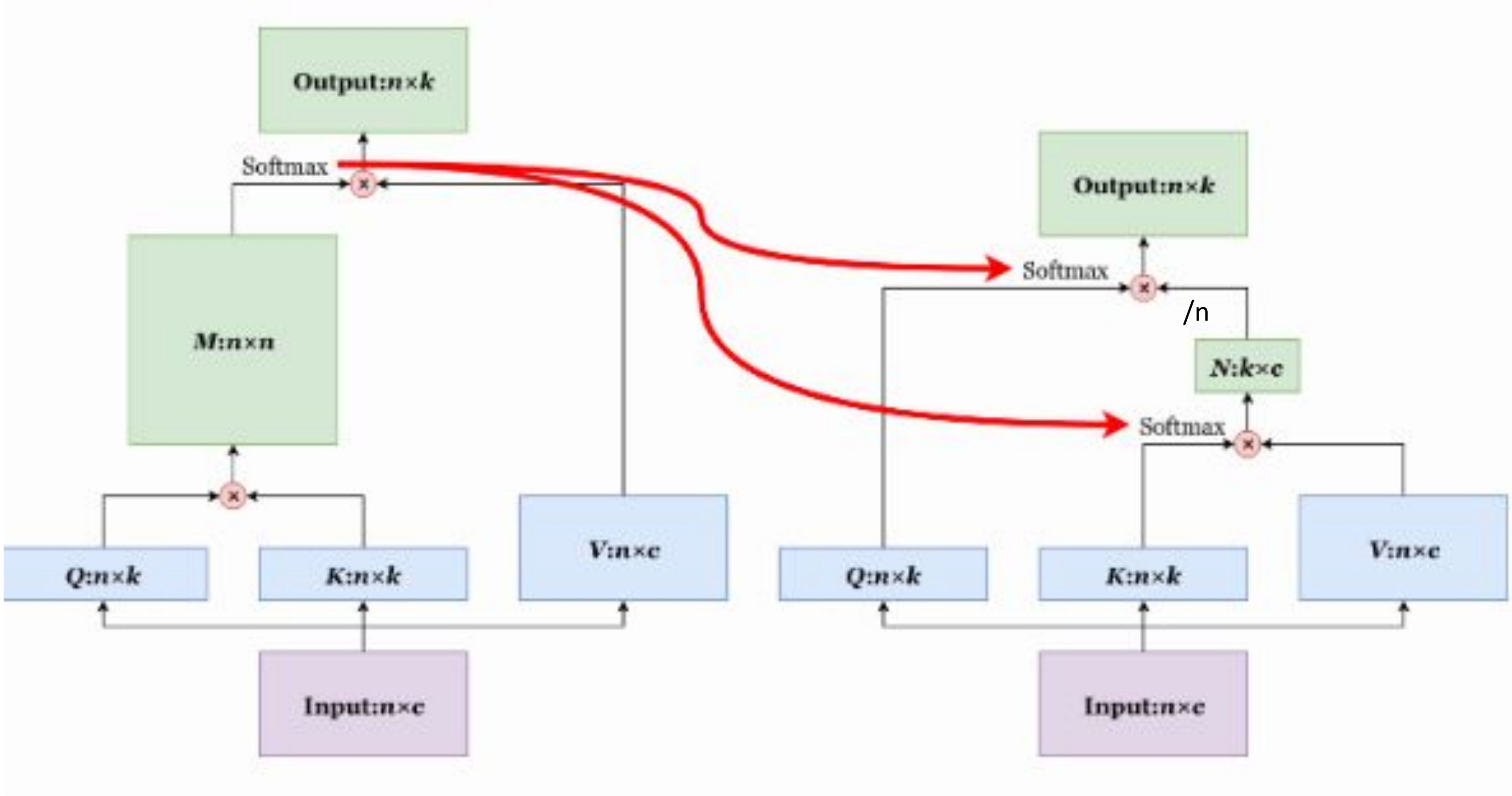
7. [1807.06521] CBAM: Convolutional Block Attention Module (arxiv.org)



Efficient
Attention



Efficient Attention



Efficient Attention

Backbone	Baseline AP	With EA modules
ResNet-50	39.4/35.1	41.2/36.7
ResNet-101	41.3/36.6	43.1/37.9
ResNeXt-101	43.5/38.5	44.9/39.5

Next, we explore the effect of efficient attention on different backbone networks. The table shows that efficient attention is consistently effective on a diversity of backbones. It provides a considerable gain (+1.4 box AP and +1.0 mask AP) even on a highly competitive, ResNeXt-101 baseline.

Stereo Depth Estimation

For stereo depth estimation, we used a PSMNet with optimized hyperparameters as the baseline. The dataset used was Scene Flow. We only experimented with adding a single DA module.

Model	EPE
iResNet-i2	1.40
PSMNet	1.09
EdgeStereo	1.12
CSPN	0.78
EA-PSMNet	**0.48**

As the table shows, EA-PSMNet has set a record of end-point error (EPE) on the Scene Flow dataset by a large margin.

Image Classification

For image classification, we used ResNet-50 as our baseline and tested on the ImageNet dataset.

Number of modules	Top-1 accuracy (%)	Improvement
0	76.052	0.000
1	76.932	0.880
2	77.312	1.260

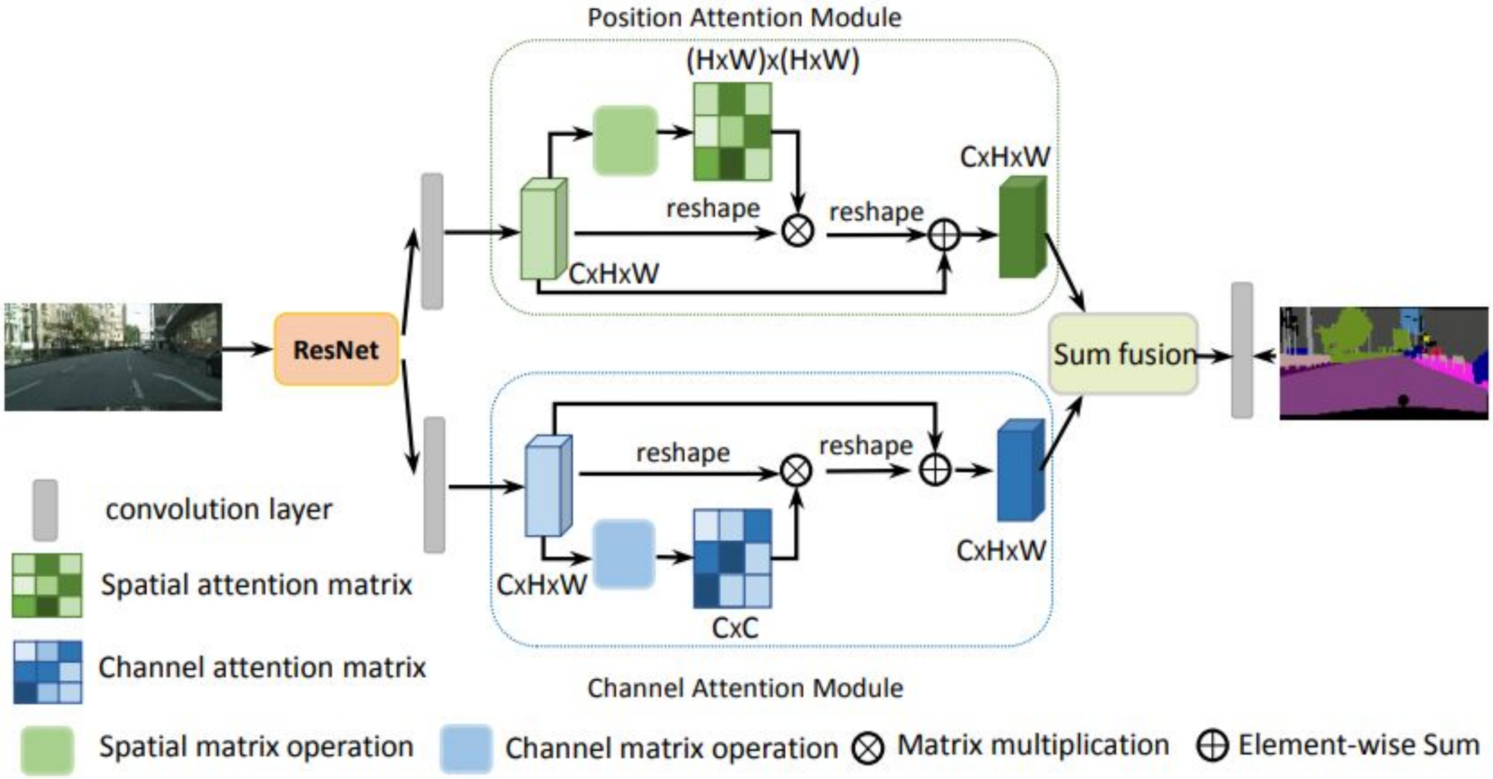
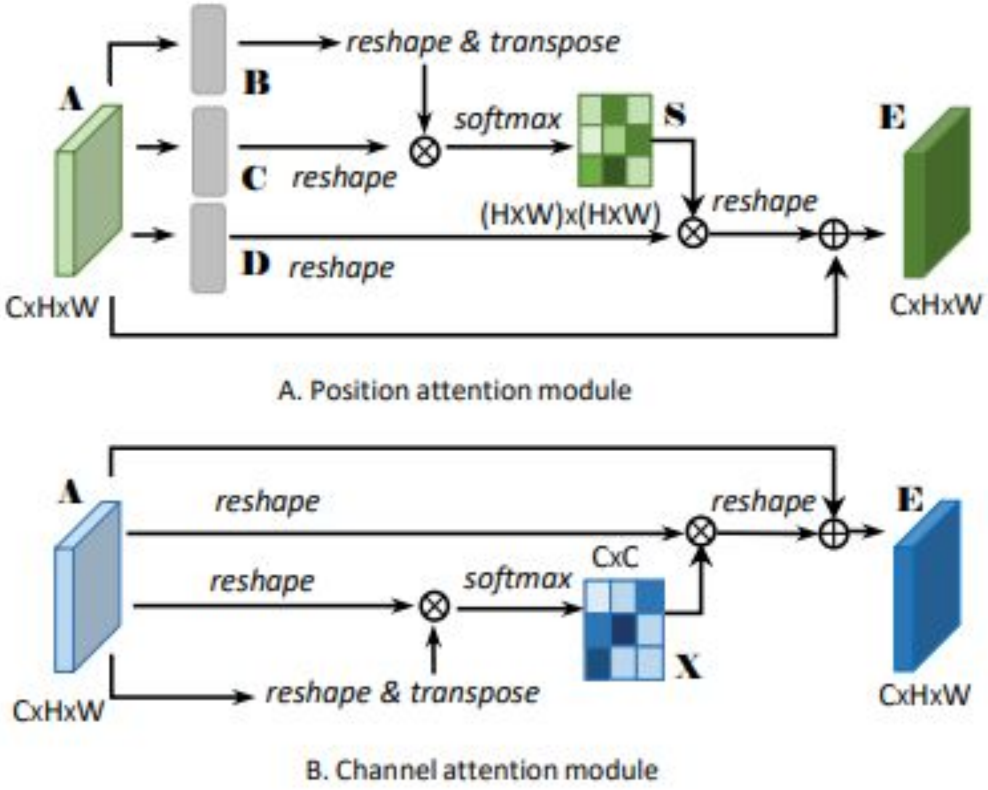


Figure 2: An overview of the Dual Attention Network. (Best viewed in color)

Dual Attention



Dual Attention Network

Methods	Mean IoU	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
DeepLab-v2 [3]	70.4	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8
RefineNet [10]	73.6	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	70
GCN [15]	76.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DUC [22]	77.6	98.5	85.5	92.8	58.6	55.5	65	73.5	77.9	93.3	72	95.2	84.8	68.5	95.4	70.9	78.8	68.7	65.9	73.8
ResNet-38 [24]	78.4	98.5	85.7	93.1	55.5	59.1	67.1	74.8	78.7	93.7	72.6	95.5	86.6	69.2	95.7	64.5	78.8	74.1	69	76.7
PSPNet [30]	78.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BiSeNet [26]	78.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PSANet [31]	80.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DenseASPP [25]	80.6	**98.7**	**87.1**	93.4	**60.7**	62.7	65.6	74.6	78.5	93.6	72.5	95.4	86.2	71.9	96.0	**78.0**	**90.3**	80.7	69.7	76.8
DANet	**81.5**	98.6	86.1	**93.5**	56.1	**63.3**	**69.7**	**77.3**	**81.3**	**93.9**	**72.9**	**95.7**	**87.3**	**72.9**	**96.2**	76.8	89.4	**86.5**	**72.2**	**78.2**

Table 3: Per-class results on Cityscapes testing set. DANet outperforms existing approaches and achieves 81.5% in Mean IoU.

Method	Mean IoU%
FCN [13]	62.2
DeepLab-v2(Res101-COCO) [3]	71.6
Piecewise [11]	75.3
ResNet38 [10]	82.5
PSPNet(Res101) [30]	82.6
EncNet (Res101) [28]	**82.9**
DANet(Res101)	82.6

Table 5: Segmentation results on PASCAL VOC 2012 testing set.

Method	Mean IoU%
FCN-8s [13]	22.7
DeepLab-v2(Res101) [3]	26.9
DAG-RNN [18]	31.2
RefineNet (Res101) [10]	33.6
Ding et al.(Res101) [6]	35.7
Dilated FCN (Res50)	31.9
DANet (Res50)	37.2
DANet (Res101)	**39.7**

Table 7: Segmentation results on COCO Stuff testing set.

Dual Attention Network