Spatially-Aware Context Neural Networks

空间感知上下文神经网络

Dongsheng Ruan, Yu Shi, Jun Wen, Nenggan Zheng, *Senior Member, IEEE*, and Min Zheng

***Abstract*— A variety of computer vision tasks benefit significantly from increasingly powerful deep convolutional neural networks. However, the inherently local property of convolution operations prevents most existing models from capturing long-range feature interactions for improved performances. In this paper, we propose a novel module, called Spatially-Aware Context (SAC) block, to learn spatially-aware contexts by capturing multi-mode global contextual semantics for sophisticated long-range dependencies modeling. We enable customized non-local feature interactions for each spatial position through re-weighted global context fusion in a non-normalized way. SAC is very lightweight and can be easily plugged into popular backbone models. Extensive experiments on COCO, ImageNet, and HICO-DET benchmarks show that our SAC block achieves significant performance improvements over existing baseline architectures while with a negligible computational burden increase. The results also demonstrate the exceptional effectiveness and scalability of the proposed approach on capturing long-range dependencies for object detection, segmentation, and image classification, outperforming a bank of state-of-the-art attention blocks.**

卷积运算固有的局部特性阻止了大多数现有模型捕获远程特征交互以提高性能。

***Index Terms*— Computer vision, attention mechanism, object detection, image classification, context modeling.**

I. Introduction

With the booming development of deep convolutional neural networks (CNNs), significant progress has been achieved with increasingly robust backbone neural networks in a variety of computer vision tasks [1]–[7]. However, most existing approaches suffer from the inherently local nature of convolutional operations that operate on local feature regions, preventing them from effectively capturing long-range interactions. Although stacking multiple layers with larger-scope convolutional filters or dilated and deformable convolutions alleviates this limitation, these techniques lead to many intractable problems, e.g., optimization difficulty, receptive-field degradation, significantly increased computational burden [8]–[11].

扩张的

顽固的

Recently, attention-based modules have been popularly employed to model long-range dependencies. These modules typically learn global spatial interactions to facilitate perceiving information beyond the local receptive field of convolutional filters [12], [13]. A most outstanding work is the non-local network (NLNet) which models long-range interactions among different spatial positions via a self-attention mechanism [13], [14]. Despite the encouraging results, the prohibitively expensive computation of each pixel-pair interactions and memory footprint prevents its broad application to various visual tasks, e.g., image segmentation.

令人望而却步

Motivated by the observation that different query positions tend to share almost the same attention map, the Global Context Network (GCNet) [12] simplifies the NLNet by learning a query-independent attention map to aggregate global context from all positions and achieves better detection performances with a negligible computational burden. However, the GCNet learns only a single-mode global context, which prevents it from fully capturing multi-mode contextual statistics of an entire image for detection and classification of different objects. Further, the GCNet applies the same global contextual information to all spatial positions, failing to respect spatially semantic variations which are critical for determining 'what' and 'where' global contexts pay attention to in each position to accurately detect different objects in an image [15]–[17]. As shown in Fig. 1, to detect different objects at different positions, we may refer to different global context information for varied non-local feature interactions modeling, e.g., the contours and pedestrian body parts in Fig. 1 (b) and (c), respectively.

未能够遵循空间语义的变化

轮廓和行人身体部位

To address the above limitations, in this paper, we propose to capture spatially-aware global contexts for each spatial position with a novel Spatially-Aware Context (SAC) attention block. Instead of fusing the same global context to all spatial positions, the SAC block customizes global contextual semantics of each position by fusing multi-mode global contexts. It is mainly composed of two parts, Multi-mode Context Aggregation (MCA) and Spatially-aware Context Customization (SCC). Specifically, MCA aggregates features from all positions to obtain a compact multi-mode contextual basis set representing different contextual information of objects. Based on each location's characteristics, SCC fuses the aggregated contextual bases via weighted sum to obtain customized global contexts for each spatial position. Further, to ensure

获得紧凑的多模式上下文基集

Manuscript received August 28, 2020; revised January 26, 2021 and March 25, 2021; accepted June 15, 2021. Date of publication July 26, 2021; date of current version August 6, 2021. This work was supported in part by the 13-5 State S&T Projects of China under Grant 2018ZX10302-206, in part by the Natural Science Foundation of China under Grant 61572433 and Grant 61972347, and in part by the Zhejiang Provincial Natural Science Foundation under Grant LR19F020005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weiyao Lin. *(Corresponding authors: Nenggan Zheng; Min Zheng.)*

Dongsheng Ruan and Jun Wen are with the Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, Zhejiang 310007, China, and also with the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310007, China (e-mail: dongshengruan@zju.edu.cn; junwen@zju.edu.cn).

Yu Shi and Min Zheng are with the State Key Laboratory for Diagnosis and Treatment of Infectious Diseases, National Clinical Research Center for Infectious Diseases, Collaborative Innovation Center for Diagnosis and Treatment of Infectious Diseases, The First Affiliated Hospital, College of Medicine, Zhejiang University, Hangzhou, Zhejiang 310006, China (e-mail: zjushiyu@zju.edu.cn; minzheng@zju.edu.cn).

Nenggan Zheng is with the Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, Zhejiang 310007, China, also with the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310007, China, and also with the Zhejiang Laboratory, Hangzhou 311121, China (e-mail: zng@cs.zju.edu.cn).

Digital Object Identifier 10.1109/TIP.2021.3097917

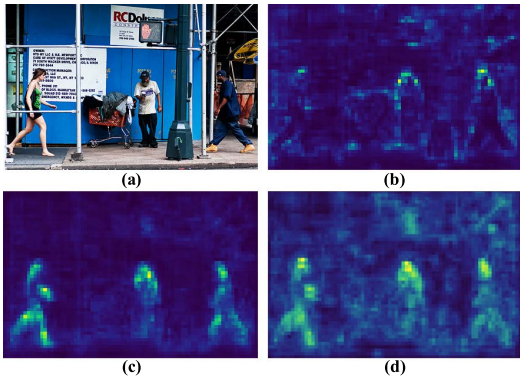


Fig. 1. (a) input image; (b-c) different global attention maps in SAC that capture varied contextual information for object detection, such as the contours (b) and human body parts (c); (d) the semantic feature map of 3×3 conv in the residual block suggesting that the spatially varied semantics are important. More visualizations are shown in Fig. 4 and Fig. 5.

a lightweight property, we incorporate the SAC block right after the 3×3 convolution operation inside each residual block, empirically shown to facilitate long-range dependencies modeling. With visualization, we show that SAC successfully captures the multi-mode global contexts of objects and learns non-local interactions complementary to the local patterns learned from typical convolutions. Our main contributions can be summarized as follows:

- We propose to learn compact multi-mode global contexts to represent different global contextual information in images, which dramatically reduces context redundancy while effectively capturing essential contextual semantics of objects.
- To obtain spatially-aware contexts, we propose to customize each spatial position to pay attention to different global contexts through re-weighted aggregation.
- We propose a novel attention module, the SAC block, which is lightweight and can be easily incorporated into popular backbone models to capture long-range discriminative features with a negligibly increased computational burden.
- With extensive experiments on the COCO, ImageNet, and HICO-DET benchmarks, the significantly improved performances over state-of-the-art attention blocks demonstrate the effectiveness of the proposed block on learning complex non-local interactions for precise object detection, segmentation and classification.

II. Related Work

A. Deep Convolutional Networks

Various computer vision tasks have benefited significantly from well-designed convolutional networks, which enable powerful capability for object categorization, localization, and segmentation. The family of GoogleNets uses different filter sizes to extract multi-scale features, which shows that it is beneficial for multi-scale object detection [18]–[21]. Motivated by ResNet [3], several deep residual architectures, such as WideResNet [4], DenseNet [5], and PyramidNet [22], have been developed with improved performances [23]. Another direction of network design is to strike a balance between accuracy and performance by limiting the scope of aggregated input channels. The lightweight property is achieved via group convolution [6], [24] or depthwise convolution [25]–[27]. Some works attempt to change the spatial scope of aggregation by dilated [8], [9] and deformable convolutions [10], [11] to enhance geometric modeling ability. Recently, network design benefits from reinforcement learning [28]–[30], evolutionary search [31], or other learning algorithms [32], [33], which allows for a shift from manual design to automated architecture search. However, these convolutional networks are still inefficient in modeling long-range feature dependencies because convolutional operators are inherently local. In contrast, as a lightweight and efficient add-on module, our SAC block aims to empower these networks to effectively capture complementary non-local feature interactions to improve performances while with a negligibly increased computational burden.

B. Attention Models

Attention mechanism is widely adopted in different tasks for capturing long-range dependencies, from natural language process [14], [34] to visual recognition [16], [35]–[37]. SENet [38] and GENet [39] perform feature re-calibration in the channel dimension by learning channel-wise dependencies from entire feature maps. Further, CBAM [15] and BAM [40] introduce spatial and channel attentions to refine features via re-scaling. AANet [41] proposes to augment convolutional operators with self-attention to capture long-range interactions, indicating the competitiveness of self-attention mechanism to replace convolutions. SAN [42] introduces pairwise and patchwise self-attention models to improve image recognition performance. ReCoR [43] progressively models more contexts through a recursive structure, providing a more feasible approach to capture contextual relationships for object detection. Relation Networks [44] learns object-wise relationship with a scaled dot-product attention mechanism. NLNet [13] employs a self-attention mechanism to model the interactions between spatial position pairs via additive fusion in the residual form. Despite significant improvements, the prohibitively expensive computational burden and GPU memory footprint limit its wide applications. A substantial amount of works have been proposed to simplify the NLNet [45]–[47]. Among them, A^2-Net [47] proposes double attention (DA) block to gather and distribute long-range features. LatentGNN [48] introduces a latent space to encode non-local relations with efficient message passing. CCNet [35] proposes to harvest the contextual information of pixels on the criss-cross path with a further recurrent operation, significantly improving computational efficiency. GCNet [12] learns one global context and adopts a bottleneck transform structure to fuse the learned global context, achieving improved performances while with a simplified structure. However, the GCNet fails to consider multi-mode global contexts and spatially semantic variations. In contrast, the proposed SAC block can effectively model spatially-aware contexts by customizing spatial contexts for each position while with the lightweight property.

III. Methods

In this section, we first briefly review attention-based global context modeling [12] and then present the proposed spatially-aware context block in detail.

A. Revisiting Global Context Modeling

Global context modeling methods aim to effectively capture long-range feature dependencies to overcome the limitation of the inherently local property of convolution operators, which can be typically abstracted into the following three procedures: (a) global attention pooling for global context aggregation; (b) feature transform to capture channel-wise dependencies; and (c) feature fusion to aggregate global context features into each position.

Consider an input feature $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, where C, W, and H denote channel number, spatial width and height, respectively. The output $\mathbf{Z} \in \mathbb{R}^{C\times H\times W}$ of the global context modeling framework can be formulated as:

$$\mathbf{Z} = F(\mathbf{X}, \delta(T(\mathbf{X}, \boldsymbol{\alpha}))), \tag{1}$$

where $\boldsymbol{\alpha}$ denotes attention maps. $T(\cdot)$, $\delta(\cdot)$ and $F(\cdot)$ denote context modeling module, feature transform module, fusion function, respectively.

Some instantiations of this framework have been proposed, such as SE [38], NL [13], and GC [12]. In particular, GC combines the advantages of both NL and SE by learning a single-mode global context and employing a bottleneck structure to transform the global context, showing stronger global context modeling capability than NL and SE. However, there are still two apparent limitations for GC. First, a single-mode global attention map is insufficient to capture multi-mode global contexts in an image for detection or classification of different objects. Second, GC fails to consider semantic variations spatially and applies the same global information to all positions, while different positions may need different global contexts.

B. Spatially-Aware Context Block

In this section, we present a novel light-weight spatially-aware context block to capture multi-mode global contexts and learn spatially-aware feature interactions for various vision tasks. Fig. 2 shows the detailed operations of the SAC block. Following the global context modeling framework described above, the SAC block is mainly composed of two parts: context modeling and feature transform. For context modeling, we define two operations: Multi-mode Context Aggregation (MCA) and Spatially-aware Context Customization (SCC).

1) Multi-Mode Context Aggregation: MCA aims to learn multi-mode global contexts to represent different channel-wise statistics. Given the input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, we perform a linear function that transforms the input $\mathbf{X}$ to K attention maps $\boldsymbol{\alpha} \in \mathbb{R}^{K\times H\times W}$:

$$\boldsymbol{\alpha} = softmax(\mathbf{W}_\phi^T\mathbf{X}) = e^{\mathbf{W}_\phi^T\mathbf{X}} \oslash \textstyle\sum_{m=1}^{N} e^{\mathbf{W}_\phi^T\mathbf{X}_m}, \tag{2}$$

where $\mathbf{W}_\phi \in \mathbb{R}^{C\times K}$ is a learned weight matrix. $\oslash$ denotes broadcast element-wise division. N is the number of spatial positions in the feature map (e.g., $N = H \times W$ for image). m denotes the spatial position m. $softmax$ function normalizes the transformed features over the space to form valid attention weights. For simplicity, we ignore the flatten operation for the spatial dimension (i.e., $H \times W \rightarrow HW$) in Eq. (2). Then, to obtain multi-mode global contexts, we perform global attention pooling to aggregate the features of all positions via weighted averaging with the attention maps $\boldsymbol{\alpha}$. These global contexts form a compact contextual basis set $\boldsymbol{\mu} \in \mathbb{R}^{K\times C}$ that represents different global semantics of the feature space. Note that K is much smaller than C ($K \ll C$), which greatly reduces context redundancy while retaining essential context information. The aggregation process can be formulated as:

$$\boldsymbol{\mu} = e^{\mathbf{W}_\phi^T\mathbf{X}} \oslash \textstyle\sum_{m=1}^{N} e^{\mathbf{W}_\phi^T\mathbf{X}_m}\mathbf{X}^T. \tag{3}$$

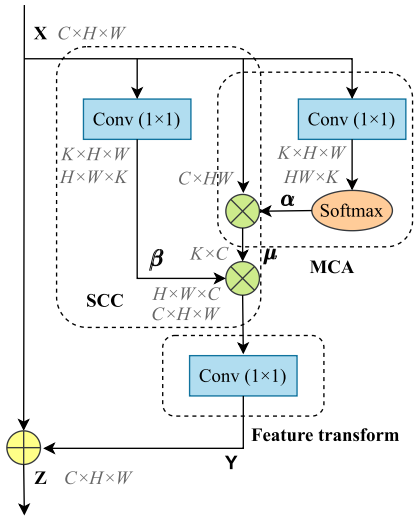


Fig. 2. An illustration of the SAC module. H, W and C denote the height, width, and channel number of the input feature map. K denotes the number of the global contexts and is C/l, where l is the reduction ratio and is set to 32. Thus K is typically much smaller than C. $\oplus$ denotes element-wise addition. $\otimes$ denotes matrix multiplication.

2) Spatially-Aware Context Customization: The SCC operation is proposed to customize spatially-aware global contexts for each position based on the learned contextual basis set $\boldsymbol{\mu}$. Specifically, it learns a set of weight vectors, and then fuses multi-mode global contexts via weighted summation to obtain the customized context features. More formally, we define the output $\mathbf{Y} \in \mathbb{R}^{C\times H\times W}$ of the SCC operation as:

$$\mathbf{Y} = (e^{\mathbf{W}_\phi^T\mathbf{X}} \oslash \textstyle\sum_{m=1}^{N} e^{\mathbf{W}_\phi^T\mathbf{X}_m}\mathbf{X}^T)^T\mathbf{W}_\theta^T\mathbf{X}, \tag{4}$$

where $\mathbf{W}_\theta \in \mathbb{R}^{C\times K}$ is a weight matrix to obtain the weight tensor $\boldsymbol{\beta} \in \mathbb{R}^{K\times H\times W}$.

It is noteworthy that we don not apply a $softmax$ function to normalize the weight tensor $\boldsymbol{\beta}$ on the channel dimension since it was found that $softmax$ function lost the diversity of context features, which will be discussed in our experiments.

3) Feature Transform: To obtain lightweight property, GC adopts a bottleneck transform module to capture the inter-dependencies between channels and is integrated after the last 1×1 convolution inside the residual block. The design has

been shown to be useful for capturing long-range dependencies. Our approach can follow the GC block. However, it leads to the following problems: 1. the number of parameters is significantly increased due to the introduction of additional convolutions into the context modeling module; 2. the computation costs are also significantly increased, as the spatial dimension is considered in the entire SAC block.

To alleviate the above problems, we move the SAC block after the 3×3 convolution inside the residual block, which significantly reduces the model complexity by about three quarters. We further observe that both the feature transform module and the residual block adopt the same bottleneck structure, which motivates us to share their last 1×1 convolution layer. This modification slightly increases the number of parameters and computational costs, but the depth of the feature transform module reduces from 3 to 1, which is conducive to ease optimization.

Finally, as practiced in [12], [13], we fuse the transformed context features into the original features $\mathbf{X}$ via a residual connection. The final output $\mathbf{Z} \in \mathbb{R}^{C \times H \times W}$ of the SAC block is defined as:

$$\mathbf{Z} = \mathbf{X} + \mathbf{W}_\rho \mathbf{Y}, \tag{5}$$

where $\mathbf{W}_\rho \in \mathbb{R}^{C \times C}$ is the feature transform matrix. $\mathbf{Y}$ is given in Eq. 4.

4) Implementation of the SAC Block: To obtain lightweight property, $\mathbf{W}_\rho$, $\mathbf{W}_\phi$, and $\mathbf{W}_\theta$ are implemented by 1×1 convolutions. To ensure at least one global context in SAC, we set the number of channels K of $\boldsymbol{\alpha}$ to be $max(1, C/l)$, where l is the reduction ratio. Since our feature transform module has only one convolution layer, our SAC block does not need any normalization methods, such as batch normalization [19] and layer normalization [49], to ease optimization and activation functions to increase the non-linearity of the block.

C. Comparisons to Other Attention Models

1) Comparison to Non-Local Network: NLNet [13] presents the non-local (NL) block for capturing long-range dependencies. The NL block learns the attention maps for all positions by computing the interactions between two positions, resulting in an expensive computational burden. As observed in GCNet [12], the attention maps in the NL block are almost the same, indicating much redundancy in the attention maps. Unlike the NL block, our SAC block does not calculate the interactions between spatial positions but learns a compact contextual basis set $\boldsymbol{\mu}$ ($K \ll HW$) to reconstruct global contextual features for each position. It effectively reduces redundancy while preserving important contextual semantics to detect or segment objects in the image.

2) Comparison to Double Attention Network: A^2-Net [47] proposes the double attention (DA) block to gather and distribute informative global features, which presents a similar understanding to ours in capturing long-range dependencies. However, the SAC block offers the following advantages. First, SAC learns a more low-rank basis set, significantly reducing the redundancy of global contexts. Second, we reconstruct global contextual features of all positions in a non-normalized manner, which provides more space for context composition, as shown in Sec. IV-A.2.g. Third, SAC is more lightweight, allowing us to apply it to the entire backbone network with only a slight increase in parameters and calculations, while better modeling long-range dependencies. The DA block can be treated as a special case of the SAC block when the weight $\mathbf{W}_\phi$ in the DA block is an identity matrix. Last but not least, the results in Table VII show that SAC can better capture long-range dependencies than DA.

TABLE I

ABLATION STUDY OF REDUCTION RATIO l. THE RESULTS ARE BASED ON MASK RCNN WITH RESNET50 AS BACKBONE. PARAMS DENOTES THE NUMBER OF PARAMETERS. FLOPS DENOTES THE NUMBER OF MULTIPLY-ADDS

Ratio l	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$	params	FLOPs
baseline	37.3	59.1	40.3	34.2	55.9	36.3	44.4M	279.4G
1	39.2	61.7	42.7	35.7	58.2	37.8	48.1M	290.2G
2	39.2	61.8	42.7	35.8	58.3	37.7	46.9M	288.6G
4	39.2	61.4	42.7	35.7	58.0	37.9	46.3M	284.8G
8	39.2	61.6	42.9	35.9	58.3	38.3	45.9M	283.9G
16	39.1	61.5	42.4	35.7	57.9	38.0	45.8M	283.4G
32	**39.6**	61.9	43.1	35.9	58.5	38.2	45.7M	283.1G
64	39.4	62.0	42.6	**36.0**	58.4	38.6	45.7M	283.1G
128	39.4	61.7	42.8	35.8	58.2	38.1	45.7M	282.9G
256	39.2	61.5	42.5	35.8	57.9	38.0	45.6M	282.9G
512	39.2	61.4	42.9	35.7	57.8	38.0	45.6M	282.9G

TABLE II

ABLATION STUDY OF $softmax$ NORMALIZATION

	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
baseline	37.3	59.1	40.3	34.2	55.9	36.3
softmax	39.2	61.7	42.5	35.7	58.1	38.0
w/o softmax	**39.6**	61.9	43.1	**35.9**	58.5	38.2

TABLE III

ABLATION STUDY OF POSITIONS. BEFORE 1×1 AND AFTER 1×1 DENOTE THAT ATTENTION BLOCK IS INSERTED BEFORE AND AFTER THE LAST 1×1 CONVOLUTION INSIDE THE RESIDUAL BLOCK, RESPECTIVELY

Backbone	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
after 1×1						
baseline	37.3	59.1	40.3	34.2	55.9	36.3
+GC	38.6	60.8	41.6	35.2	57.4	37.2
+SAC	**39.0**	61.3	42.5	**35.6**	57.8	37.9
before 1×1						
baseline	37.3	59.1	40.3	34.2	55.9	36.3
+GC	38.7	61.0	42.2	35.4	57.4	37.4
+SAC	**39.6**	61.9	43.1	**35.9**	58.5	38.2

TABLE IV

ABLATION STUDY OF EFFECTIVE GAIN OF SAC. THE RESULTS SHOW THAT THE PROPOSED MCA AND SCC OPERATIONS CAN EFFECTIVELY MODEL LONG-RANGE DEPENDENCIES

Backbone	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
baseline	37.3	59.1	40.3	34.2	55.9	36.3
+1x1 conv	37.5	59.1	40.7	34.4	55.8	46.3
+SAC	**39.6**	61.9	43.1	**35.9**	58.5	38.2

TABLE V

COMPARISONS BETWEEN GC AND SAC BASED ON MASK RCNN AND CASCADE MASK RCNN. THE NUMBERS IN BRACKETS DENOTE THE IMPROVEMENTS OVER THE BASELINE BACKBONE. FPS VALUES ARE MEASURED ON THE SAME MACHINE WITH A SINGLE GEFORCE RTX 2080TI GPU UNDER THE SAME MMDETECTION FRAMEWORK, USING A BATCH SIZE OF 1. THE RESULTS SHOW THAT SAC CONSISTENTLY OUTPERFORMS GC ACROSS A VARIETY OF BACKBONES AND DETECTORS

Detector	Backbone	FPS	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
Mask RCNN [50]	ResNet50 [3]	9.9	37.3	59.1	40.3	34.2	55.9	36.3
	+GC [12]	9.6	38.6(+1.3)	60.8	41.6	35.2(+1.0)	57.4	37.2
	+SAC	9.4	**39.6**(+**2.3**)	61.9	43.1	**35.9**(+**1.7**)	58.5	38.2
	ResNet101 [3]	8.9	39.4	61.0	43.3	35.9	57.7	38.4
	+GC [12]	8.0	40.8(+1.4)	63.1	44.7	37.0 (+1.1)	59.6	39.5
	+SAC	7.9	**41.0**(+**1.6**)	63.0	44.7	**37.2**(+**1.3**)	59.8	39.7
Cascade Mask RCNN [51]	ResNet50 [3]	7.9	41.2	59.1	45.1	35.7	56.3	38.6
	+GC [12]	7.7	42.6(+1.4)	60.8	46.6	36.7(+1.0)	57.8	39.6
	+SAC	7.4	**43.1**(+**1.9**)	61.5	47.2	**37.3**(+**1.6**)	58.4	40.4
	ResNet101 [3]	7.0	42.6	60.7	46.7	37.0	58.0	39.9
	+GC [12]	6.7	43.9(+1.3)	62.2	48.0	37.9(+0.9)	59.5	40.9
	+SAC	6.6	**44.5**(+**1.9**)	62.9	48.6	**38.4** (+**1.4**)	59.9	41.5

TABLE VI

COMPARISONS BETWEEN GC AND SAC ON MULTIPLE POPULAR DETECTORS. THE NUMBERS IN BRACKETS DENOTE THE IMPROVEMENTS OVER THE BASELINE BACKBONE. FPS VALUES ARE MEASURED ON THE SAME MACHINE WITH A SINGLE GEFORCE RTX 2080Ti GPU USING A BATCH SIZE OF 1. THE RESULTS SHOW THAT SAC WORKS WELL ON OTHER DETECTORS

Detector	Backbone	FPS	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Faster R-CNN [52]	ResNet50 [3]	12.6	36.4	58.4	39.1	21.6	40.1	46.6
	+GC [12]	12.1	37.5(+1.1)	59.9	40.8	22.6	41.8	47.4
	+SAC	11.9	**38.4**(+**2.0**)	61.1	41.6	23.5	42.6	48.2
	ResNet101 [3]	10.3	38.5	60.5	41.8	22.3	43.2	49.8
	+GC [12]	9.4	39.8(+1.3)	62.3	43.0	23.9	44.3	50.8
	+SAC	9.3	**40.0**(+**1.5**)	62.4	43.6	24.1	44.8	51.4
Cascade R-CNN [53]	ResNet50 [3]	9.8	40.5	58.7	44.1	21.7	43.8	53.8
	+GC [12]	9.4	42.0(+1.5)	60.9	45.8	24.1	46.0	54.6
	+SAC	9.3	**42.7**(+**2.2**)	61.8	46.5	24.8	46.7	55.7
	ResNet101 [3]	8.2	42.0	60.3	45.9	23.2	46.0	56.3
	+GC [12]	7.7	43.1(+1.1)	62.0	47.0	24.6	47.2	57.2
	+SAC	7.6	**43.9**(+**1.9**)	62.8	47.8	25.6	47.8	57.8
RetinaNet [54]	ResNet50 [3]	12.9	35.6	55.5	38.3	20.0	39.6	46.8
	+GC [12]	12.3	37.1(+1.5)	57.4	39.7	21.8	41.1	48.8
	+SAC	12.1	**37.8**(+**2.2**)	58.4	40.4	22.0	41.8	49.2
	ResNet101 [3]	9.9	37.7	57.5	40.4	21.1	42.2	49.5
	+GC [12]	9.6	39.0(+1.3)	59.5	42.1	23.0	43.7	51.2
	+SAC	9.5	**39.3**(+**1.6**)	60.0	42.3	22.3	43.8	51.8

3) Comparison to Global Context Network: Instead of just learning single-mode global context and fusing the same global contextual feature into all positions, our SAC block captures multi-mode global contexts and customizes contextual features for each position. Each location receives the context that best matches the input image by perceiving the global context information. Hence, the SAC block can capture long-range dependencies more effectively than the GC block. Furthermore, if we set K to 1 and adopt the bottleneck transform module, GC also can be seen as a special form of SAC.

Compared with GC, SAC can achieve better performance with fewer parameters and GPU memory footprint. The FLOPs of SAC are slightly larger than those of GC due to the introduction of spatial dimension, but which is almost negligible compared to the backbone networks, as shown in Table VII and X.

IV. EXPERIMENTS

In this section, we evaluate our approach on two popular benchmarks with three vision tasks, object detection/segmentation on COCO 2017 [55] and image classification on ImageNet [56].

目标检测、目标分割、图像分类

A. Object Detection/Segmentation on COCO

We investigate our SAC block on object detection and instance segmentation on COCO 2017, which has 80 object categories. All models are trained using 115k images *train* split for training and 5k *validation* split for validation. We report the standard COCO-style average precisions (AP) at different IoU thresholds ($AP_{0.5}$ and $AP_{0.75}$) or object scales (AP_S, AP_M, and AP_L). For Mask RCNN and Cascade Mask RCNN, both box AP (AP^{bbox}) and mask AP (AP^{mask}) are evaluated.

TABLE VII

COMPARISONS WITH STATE-OF-THE-ART ATTENTION BLOCKS. PARAMS DENOTES THE NUMBER OF PARAMETERS. FLOPS DENOTES THE NUMBER OF MULTIPLY-ADDS. FOR A FAIR COMPARISON, WE RETRAIN OTHER ATTENTION MODELS IN THE SAME TRAINING SETTING AS SAC. FPS VALUES ARE MEASURED ON THE SAME MACHINE WITH A SINGLE GEFORCE RTX 2080Ti GPU USING A BATCH SIZE OF 1. THE BEST RESULTS ARE MARKED AS **BOLD**

Backbone	Params	FLOPs	FPS	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{mask}	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$
ResNet50 [3]	44.4M	279.4G	9.9	37.3	59.1	40.3	34.2	55.9	36.3
+SE [38]	46.9M	279.5G	9.7	38.3(+1.0)	60.3	41.3	35.0(+0.7)	57.0	36.8
+CBAM [15]	46.9M	279.7G	9.6	38.9(+1.6)	61.3	42.1	35.4(+1.2)	57.8	37.7
+BAM [40]	44.8M	280.2G	9.5	37.3(+0.0)	58.9	40.4	34.2(+0.0)	55.6	36.4
+DA [47]	64.3M	336.6G	5.3	38.9(+1.6)	61.2	41.9	35.5(+1.3)	57.8	37.5
+GC [12]	46.9M	279.6G	9.6	38.6(+1.3)	60.8	41.6	35.2(+1.0)	57.4	37.2
+SAC	45.7M	283.1G	9.4	**39.6**(+**2.3**)	61.9	43.1	**35.9**(+**1.7**)	58.5	38.2

1) Network Architectures: To validate the effectiveness and generality of our approach, we experiment with different combination of popular backbone network ResNet [3], and state-of-the-art detection architectures including Faster RCNN [52], Mask RCNN [50], Cascade RCNN [53], Cascade Mask RCNN [51] and RetinaNet [54]. By default, our SAC block is integrated into stage c3-c5 of the backbone network. The reduction ratio l is set as 32.

2) Training: All experiments are implemented with *mmdetection V1* [51] framework. The short edge of the input image is resized to 800, and the long edge is limited to 1333. We train on 4 GPUs with two images per each for 12 epochs. Following the conventional finetuning setting [50], **we use frozen BathNorm instead of synchronized BathNorm** due to limited 4 GPUs. According to the linear scaling rule [57], the initial learning rate is set to 0.01, which is decreased by 10 at the 9th and 12th epochs. All models are trained using synchronized SGD with a weight decay of $1e$-4 and momentum of 0.9. All backbone networks are pre-trained on ImageNet. Note that **all attention modules are trained from scratch**. We finetune all layers except for c1 and c2 with FPN [58], detection and segmentation heads. Other hyper-parameters follow the default settings of the *mmdetection* framework. The reduction ratio in GC is set to 16. For a fair comparison, we retrain other attention models in the same training setting as SAC.

所有的注意力模块都是从头开始训练的

a) Object detection and instance segmentation: We evaluate our SAC block on the tasks of object detection and instance segmentation. We select two popular detectors: Mask RCNN [50] and Cascade Mask RCNN [51] with ResNet backbone. Table V shows that SAC consistently performs better than GC across various backbones and detectors, indicating that our proposed approach is more effective in capturing long-range dependencies. In particular, with Mask RCNN based on ResNet50 backbone, our approach improves AP^{bbox} over GC by 1.0%. Compared to the baseline, significant gains (1.6-2.3% $\uparrow$ on AP^{bbox} and 1.3-1.7% $\uparrow$ on AP^{mask}) are observed with just a slight increase in parameters and computation costs. In the case of stronger detector Cascade Mask RCNN, the SAC block still improves the performance by 1.9% $\uparrow$ AP^{bbox} and 1.5% $\uparrow$ AP^{mask} regardless of the backbone depth, suggesting that our approach is complementary to the capacity of stronger detectors.

b) Experiments on other detectors: To evaluate whether our approach can be generalized to other detectors, we conduct experiments on three state-of-the-art detection frameworks, including Faster RCNN [52], Cascade RCNN [53] and RetinaNet [54]. The results are reported in Table VI. We make the following four observations. First, SAC significantly improves the baseline performance and outperforms GC, which is consistent with the results in Sec. IV-A.2.a. Second, SAC can yield substantial performance gains for both one-stage and multi-stage detectors. Third, for ResNet50 backbone, the gain of detection performance is more noticeable with basically more than 2.0% AP^{bbox} across a variety of detectors. The possible reason is that shallow network is not good at sensing the global image information due to the small receptive field, which is well alleviated by SAC. Finally, we observe that SAC based on ResNet50 backbone achieves performance comparable to the ResNet101 baseline, with significantly fewer parameters and less computation costs. This comparison shows that the gain brought by SAC is complementary to going deeper. These observations demonstrate the effectiveness and generalization of our approach.

c) Comparisons with state-of-the-art models: Table VII compares our approach with other attention blocks. In addition to BAM, we insert each attention block into all residual blocks (c3 + c4 + c5). For BAM, we follow the practice in original work [40]. We note that our SAC block outperforms other attention blocks by a large margin, with fewer or comparable parameters and computations. Furthermore, DA and CBAM achieve the second-highest performance gain due to the introduction of spatial dimension, which confirms the importance of spatial semantics for object detection. We also observe that BAM does not bring performance gain, possibly because only three BAMs are added.

d) Experiments on stronger backbones: Next, we access the ability of SAC to generalize to stronger backbones. To this end, we insert our SAC blocks into all residual blocks (c3-5) of the backbone ResNeXt [6]. The results are shown in Table VIII. It can be observed that SAC is still able to significantly improve the baseline performance and outperforms GC across various detectors. For the strongest backbone, with deformable convolution and cascade mask RCNN in ResNeXt101, our SAC block can boost performance by 1.0% $\uparrow$ on AP^{bbox}. Further, with multi-scale training

TABLE VIII

RESULTS OF SAC WITH RESNEXT101 BACKBONE ON MULTIPLE POPULAR DETECTORS. "*" DENOTES MULTI-SCALE TRAINING AND SINGLE-SCALE TESTING. **X:** RESNEXT. **DCN:** DEFORMABLE CONVOLUTIONAL NETWORK. THE RESULTS SHOW SAC OUTPERFORMS GC ON STRONGER BACKBONES

Detector	Backbone	AP^{bbox}	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$
Faster RCNN	X101 [6]	40.1	62.0	43.8
	+GC [12]	41.3(+1.2)	63.7	44.7
	+SAC	**41.8**(+**1.7**)	64.4	45.7
Mask RCNN	X101 [6]	41.1	62.8	45.0
	+GC [12]	42.1(+1.0)	64.4	46.2
	+SAC	**42.6**(+**1.5**)	64.9	46.4
Cascade RCNN	X101 [6]	43.6	62.2	47.4
	+GC [12]	44.5(+0.9)	63.8	48.2
	+SAC	**45.0**(+**1.4**)	64.1	49.1
Cascade Mask RCNN	X101 [6]	44.4	62.6	48.6
	+GC [12]	45.1(+0.7)	63.9	49.2
	+SAC	**45.6**(+**1.2**)	64.2	49.9
Cascade Mask RCNN	X101-DCN [10]	47.1	65.9	51.6
	+GC [12]	47.7(+0.6)	66.8	52.0
	+SAC	48.1(+1.0)	67.2	52.3
	+GC* [12]	48.0(+0.9)	67.1	52.3
	+SAC*	**48.3**(+**1.2**)	67.2	52.7

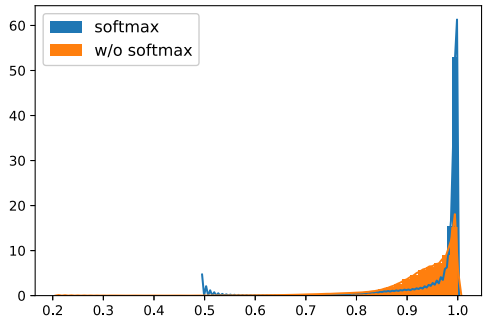


Fig. 3. Similarity distributions with and without $softmax$ function. The x-axis represents similarity. The y-axis represents counting.

strategy, SAC achieves 48.3% AP^{bbox}. These results demonstrate the effectiveness and generalization of our approach.

e) Runtime: We report the runtime of SAC and other attention methods on the same machine with a single GeForce RTX 2080Ti GPU using a batch size of 1. The results in Table V, VI and VII show that the FPS of our method is comparable to that of other attention modules except DA, with a slight decrease of about 0.1-0.2 FPS. However, our method outperforms other methods by large margins. For example, SAC outperforms GC by 1.0% on AP^{bbox} and 0.7% on AP^{mask} gains. We think that this is affordable compared to the performance gain from SAC.

f) Visualization and interpretation:

i) Visualization of Attention Map: To verify whether our SAC block can capture different contextual information, we randomly select 4 images from COCO and visualize the attention maps $\boldsymbol{\alpha}$ of the SAC block in the last residual block of stage c4, as shown in Fig. 4. We can see that different attention maps are indeed responsible for extracting varied contextual information from the images. Some focus on the foreground or background, while others focus on the contours of objects. However, this is not available to GC because GC is a channel-based attention block that loses spatial semantics.

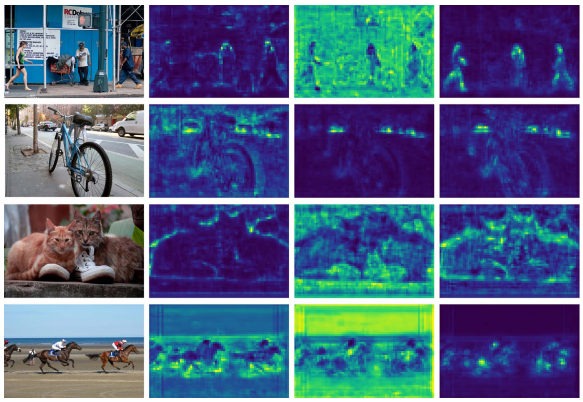


Fig. 4. Visualizations of the attention maps $\boldsymbol{\alpha}$ of the SAC block in the last residual block of stage c4. The first column denotes input images. Columns 2-4 denote attention maps which mainly focus on the contours of objects, background, and foreground, respectively.

ii) Visualization of Semantic Feature: Further, we explore how our SAC block plays a role in the backbone network. To this end, we examine the semantic features of the 3×3 conv and the SAC block before addition. Specifically, we compute the vector lengths of all positions in their feature maps as their semantic features (i.e., $\|\mathbf{X}\|$ and $\|\mathbf{Z} - \mathbf{X}\|$). The visualization results are shown in Fig. 5. We observe that SAC did focus on different semantic information. For example, in the second image on the last row, SAC captures not only the cat's head, but also its body. Moreover, the contours of the objects, such as bike, horse, and giraffe, are also well captured. However, these semantics are not learned by 3×3 convolution. Through addition fusion, SAC enriches and complements the semantic information of 3×3 convolution, thus improving the backbone network's representation ability. In contrast, GC fuses the same semantic information into each spatial position, missing these semantics. The visualization results explain why SAC is more effective than GC in global context modeling.

g) Ablation study: We report the ablation studies based on Mask RCNN detection framework with ResNet50 as backbone.

i) Reduction Ratio: The reduction ratio is a vital hyper-parameter which provides a tradeoff between capacity and model complexity. To investigate this relationship, we conduct a series of experiments for a range of different l values. The results in Table I show that the performance does not improve monotonically with the increased capacity. We found that SAC with $l \leqslant 16$ yields a similar gain. The results suggest that too much global contextual information does not better capture long-range dependencies. We speculate that it is the result of global contextual saturation. With the ratio $16 < l < 128$, the performance of SAC is significantly improved ($\geqslant$2.2% $\uparrow$ on AP^{bbox} and $\geqslant$1.5% $\uparrow$ on AP^{mask}). In particular, the setting $l = 32$ makes a good tradeoff

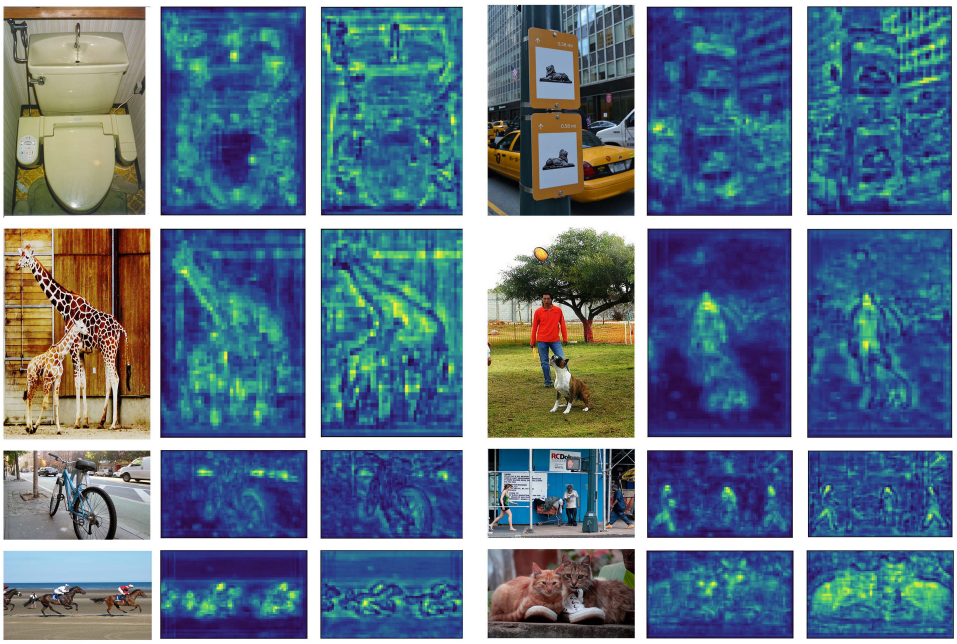


Fig. 5. Visualizations of semantic features learned by 3×3 convolution and SAC before fusion in the last residual block of stage c4. First column: input images. Second column: 3×3 convolution. Third column: SAC block.

between performance and complexity. Thus, this value is used for all experiments by default. In the case of $l \geqslant 128$, the performance is similar to the case of $l \leqslant 16$ with a similar gain, indicating too few global contexts cannot adequately capture long-range dependencies. It is worth noting that when l is 512, i.e., $K = 1$, SAC only captures the single-mode global context but retains the spatially-aware context customization operation, in which case the performance drops by 0.4% AP^{bbox} compared to the case of $l = 32$. However, we find that this case is still superior to GC since the SCC operation learns different weights for each position. These findings suggest that our proposed multi-mode global contexts and spatially-aware context customization are critical to help the network better capture long-range dependencies.

ii) Normalization: Table II shows the results of the SAC block with and without $softmax$ normalization, indicating that it is more effective for global context modeling to fuse multi-mode global contexts in a non-normalized manner. To further explore it, we calculate the cosine similarity $cos(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)$ between any two positions i and j, where $\boldsymbol{\beta}_i \in \mathbb{R}^K$ denotes the weight vector for position i in the weight tensor $\boldsymbol{\beta}$. Then we plot their similarity distributions, as shown in Fig. 3. We observe that in the case of adopting $softmax$ function, the weight tensor $\boldsymbol{\beta}$ has a very high similarity, leading to similar context features in the fusion process. In contrast, the case without $softmax$ effectively alleviates this phenomenon, improving global context modeling capability. It is worth noting that the distribution of cosine similarity is concentrated in >0.5. We argue that there are two possible reasons: 1) in the feature map, most regions are background, and the spatial positions of these regions tend to learn similar contexts; 2) the lack of extra supervision for SAC leads to the learning of redundant contexts. Although this phenomenon is alleviated by not using $softmax$ function, redundancy is not completely eliminated.

iii) Positions: To investigate the impact of different positions, we insert our SAC block before and after the last 1×1 convolution inside the residual block. Note that, in the case of the after 1×1 position, SAC also uses the bottleneck transform module to reduce the complexity, where layer normalization is replaced by batch normalization due to the introduction of spatial dimension. The results are shown in Table III. It is clear that our SAC block is superior to the GC block in both cases, with significant gains of 0.9% AP^{bbox} and 0.5% AP^{mask} in the before 1×1 case. Furthermore, before 1×1 achieves higher performance than that of after 1×1 for both GC and SAC, demonstrating that it is more effective to fuse global contextual features into local features of 3×3 convolution. Consequently, we adopt before 1×1 as the default.

iv) Effective Gain of SAC: Since the SAC blocks increase the depth of the backbone network and may yield partial gain, we examine how much effective gain the context 部分增益 modeling module can bring. For this purpose, we remove the MCA and SCC operations from SAC and only retain the feature transform module, i.e., insert one 1×1 convolution after the 3×3 convolution of each residual block (stage c3 + c4 + c5). From Table IV, it can be seen that adding 1×1 convolution slightly improves performance by 0.2% AP^{bbox} and 0.2% AP^{mask}. In contrast, our SAC block brings significant gains of 2.3% AP^{bbox} and 1.7% AP^{mask}. This comparison shows that the performance improvement comes primarily from the MCA and SCC operations, rather than the increase in depth, indicating that multi-mode global context aggregation and

TABLE IX
RESULTS OF SAC AND GC INSERTED INTO DIFFERENT STAGES OF RESNET50. THE NUMBERS IN BRACKETS DENOTE THE IMPROVEMENTS OVER THE BASELINE

Stage	Backbone	Params	FLOPs	Top-1	Top-5
baseline	ResNet50	25.56M	4.122G	76.426	93.088
stage 2	+GC	25.58M	4.124G	76.898(+0.372)	93.286(+0.198)
	+SAC	25.57M	4.164G	**76.904**(+**0.478**)	**93.376**(+**0.288**)
stage 3	+GC	25.69M	4.124G	76.838(+0.412)	93.242(+0.154)
	+SAC	25.63M	4.177G	**77.260**(+**0.834**)	**93.624**(+**0.536**)
stage 4	+GC	26.36M	4.124G	76.922(+0.496)	93.222(+0.134)
	+SAC	25.98M	4.204G	**77.236**(+**0.810**)	**93.428**(+**0.340**)
stage 5	+GC	27.14M	4.124G	**75.914**(−**0.512**)	**92.982**(−**0.106**)
	+SAC	26.39M	4.163G	75.814(−0.612)	92.830(−0.258)

spatially-aware context customization are crucial to capture long-range dependencies.

B. Image Classification on ImageNet

In this section, we evaluate our approach on ImageNet. The ImageNet 2012 dataset contains 1.28 million training images and 50K validation images with 1000 classes. We do standard data augmentation for training: a 224 × 224 crop is randomly sampled from a 256 × 256 image or its horizontal flip using the scale and aspect ratio augmentation. During the test, 224 × 224 pixels are cropped from the center of an image whose shorter side is 256. During training and testing, data is normalized using channel means and standard deviations. We train our models for 100 epochs on 4 GPUs with 64 images per GPU (effective batch size 256) with 5 epochs of linear warm up. The initial learning rate starts from 0.1 and drops every 30 epochs. For a fair comparison, we retrain other attention models with the same training strategy as SAC.

In object detection task, through pre-training on ImageNet, the backbone network usually has good weight initialization, making it easy to capture global contextual information even if attention modules are inserted in all stages. However, we empirically found that in image classification task, too many SAC and GC blocks seriously hinder the optimization of the network if trained from scratch, which is consistent with the finding of this study [59]. Therefore, we need to explore the most suitable insertion position to maximize the performance gain.

1) Stages: Here we investigate which stage to add block. Table IX shows the results of inserting the SAC block at different stages. We find that for stage 2-4, SAC provides an excellent performance gain for baselines and is superior to GC with fewer parameters. For stage 5, both the SAC and GC blocks experience performance degradation, probably because deeper layers have sufficiently large receptive fields. The insertion of SAC and GC hinders the gradient flow of the network. We notice that the FLOPs of SAC is slightly higher than that of GC due to the space attention introduced by SAC, but the increased FLOPs are marginal over baseline (about 0.99% ↑). Inspired by the above observation, we insert SAC in the last residual block in stage 2-4 to maintain the lightweight property.

TABLE X
COMPARISONS WITH STATE-OF-THE-ART ATTENTION BLOCKS ON IMAGENET. DA, BAM, AND GC ARE INSERTED AFTER THE LAST RESIDUAL BLOCK IN STAGE 2-4. SE AND CBAM ARE INSERTED INTO EACH RESIDUAL BLOCK IN STAGE 2-5

Backbone	Params	FLOPs	Top-1	Top-5
ResNet50 [3]	25.56M	4.122G	76.43	93.09
SE-ResNet50 [38]	28.09M	4.130G	77.18	93.67
BAM-ResNet50 [40]	25.92M	4.205G	76.90	93.40
CBAM-ResNet50 [15]	28.09M	4.139G	77.34	93.69
DA-ResNet50 [47]	26.94M	4.740G	77.33	93.59
GC-ResNet50 [12]	25.73M	4.124G	76.92	93.34
SAC-ResNet50	25.65M	4.163G	**77.61**	**93.86**
ResNet101 [3]	44.55M	7.849G	77.70	93.80
SE-ResNet101 [38]	49.33M	7.863G	78.47	94.19
BAM-ResNet101 [40]	44.91M	7.933G	78.22	94.02
CBAM-ResNet101 [15]	49.33M	7.879G	78.49	94.23
DA-ResNet101 [47]	45.93M	8.465G	78.43	94.14
GC-ResNet101 [12]	44.73M	7.851G	78.39	94.14
SAC-ResNet101	44.64M	7.891G	**78.63**	**94.30**

2) Comparisons With State-of-the-Arts: The results are shown in Table X. With only a slight increase in parameters and calculations, our SAC-ResNet achieves a significant performance improvement over the baseline (⩾0.93% Top-1 accuracy ↑). Regardless of network depth, SAC performs better than GC, and in particular, SAC-ResNet50 yields a 0.69% Top-1 gain with a parameter decrease of 0.31%. Moreover, our approach is also superior to other attention methods with fewer parameters and comparable computational costs. These results demonstrate that the superiority of our approach in image classification task.

C. Human Object Interactions on HICO-DET

We further evaluate our proposed method on human object interactions task. We adopt the widely-used HICO-DET [61] dataset, which has 38118 training and 9658 testing images. HICO-DET annotates the images for 600 human-object interactions over 80 object categories. Following the previous works [60], [62], [63], we report mean Average Precision (mAP) score in Full, Rare, and Non-Rare Categories.

1) Implementation Details: We select VSGNet [60] (CVPR2020) as the detector. SAC-ResNet is used as the backbone feature extractor, where SAC is inserted into the

TABLE XI
RESULTS (mAP) OF SAC AND GC BASED ON VSGNET [60] USING RESNET [3] AS BACKBONE ON HICO-DET VALIDATION SET. SAC AND GC ARE INSERTED INTO THE LAST RESIDUAL BLOCK IN STAGE 2-4 OF RESNET.

Detector	Backbone	Full	Rare	Non-Rare
VSGNet	ResNet50	18.34	12.84	19.97
+GC	ResNet50	18.68	13.34	20.27
+SAC	ResNet50	**19.44**	**15.17**	**20.72**
VSGNet	ResNet101	19.75	16.02	20.87
+GC	ResNet101	20.07	15.82	21.32
+SAC	ResNet101	**20.56**	**18.01**	**21.34**

last residual block in stage 2-4 of ResNet. We pre-train this backbone on ImageNet. Following [60], [62] we do not fine-tune the backbone SAC-ResNet and Faster-RCNN during the training process. Our initial learning rate is set to 0.001 with a batch size of 16. Stochastic Gradient Descent (SGD) is used with a weight decay of 0.0001 and a momentum of 0.9. We increase the learning rate to 0.01 for all the layers except for the spatial attention branch in VSGNet between epoch 10 to epoch 28. All models are trained for 50 epochs. For more implementation details, please refer to the paper of VSGNet [60]. For a fair comparison, we also pre-train other models on ImageNet and then re-train them on HICO-DET in the same training setting as SAC.

2) Results: The results are shown in Table XI. Compared to the baseline VSGNet, our SAC block yields significant gains, especially for the Rare category, with only a slight parameter increase. Specifically, we show a 6.0% relative performance gain on ResNet50 backbone and a 4.1% relative gain on ResNet101 backbone. We can also see that SAC consistently is better than GC at different network depths with fewer parameters. Furthermore, with ResNet101 backbone, our SAC-VSGNet achieves a performance of 20.56% mAP, outperforming the best result of 19.80% mAP published in [60]. These results demonstrate that our method can effectively capture long-range dependencies.

V. CONCLUSION

In this paper, we present a novel spatial attention module, the SAC block, that can effectively capture long-range dependencies by aggregating multi-mode global contexts and customizing spatially-aware contextual information. Based on its lightweight property, the SAC block can be inserted into various backbone networks with a negligible parameter and computation increase. Comprehensive experiments on COCO, ImageNet, and HICO-DET benchmarks are conducted to evaluate our approach. The results show that our SAC block can provide a solid improvement over baselines across various detection frameworks and backbones. Meanwhile, we also demonstrate the superiority of SAC over other attention networks. We hope SAC will become an indispensable element of deep network architecture.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[4] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–15.

[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[6] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[7] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[9] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122

[10] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. ICCV*, 2017, pp. 764–773.

[11] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets V2: More deformable, better results," in *Proc. CVPR*, 2019, pp. 9308–9316.

[12] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. ICCV*, 2019, pp. 1–10.

[13] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.

[14] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[15] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, 2018, pp. 3–19.

[16] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. CVPR*, 2017, pp. 5659–5667.

[17] Y. Wu *et al.*, "Rethinking classification and localization for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10186–10195.

[18] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-V4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, 2017, vol. 31, no. 1, pp. 4278–4284.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016, pp. 2818–2826.

[22] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. CVPR*, 2017, pp. 5927–5935.

[23] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[24] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*, 2018, pp. 122–138.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[27] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proc. ICML*, 2018, pp. 4095–4104.

[28] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. ICLR*, 2016, pp. 1–16.

[29] B. Zoph, V. K. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. CVPR*, 2018, pp. 8697–8710.

[30] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. ICLR*, 2017, pp. 1–18.

[31] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI*, 2019, pp. 4780–4789.

[32] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with Bayesian optimisation and optimal transport," in *Proc. NeurIPS*, 2018, pp. 1–32.

[33] R. Luo, F. Tian, T. Qin, E. Chen, and T. Liu, "Neural architecture optimization," in *Proc. NeurIPS*, 2018, pp. 1–16.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[35] Z. Huang *et al.*, "CCNet: Criss-cross attention for semantic segmentation," 2018, *arXiv:1811.11721*. [Online]. Available: http://arxiv.org/abs/1811.11721

[36] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. CVPR*, 2019, pp. 3146–3154.

[37] H. Zhao *et al.*, "PSANet: Point-wise spatial attention network for scene parsing," in *Proc. ECCV*, 2018, pp. 267–283.

[38] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[39] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. NeurIPS*, 2018, pp. 9401–9411.

[40] J. Park, S. Woo, J. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *BMCV*, 2018, p. 147.

[41] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. ICCV*, 2019, pp. 3286–3295.

[42] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. CVPR*, 2020, pp. 10076–10085.

[43] Z. Chen, J. Zhang, and D. Tao, "Recursive context routing for object detection," in *Proc. IJCV*, 2020, pp. 1–19.

[44] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. CVPR*, 2018, pp. 3588–3597.

[45] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. ICCV*, 2019, pp. 593–602.

[46] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. ICCV*, 2019, pp. 9167–9176.

[47] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-nets: Double attention networks," in *Proc. NeurIPS*, 2018, pp. 352–361.

[48] S. Zhang, X. He, and S. Yan, "LatentGNN: Learning efficient non-local relations for visual recognition," in *Proc. ICML*, 2019, pp. 7374–7383.

[49] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: https://arxiv.org/abs/1607.06450

[50] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[51] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*. [Online]. Available: http://arxiv.org/abs/1906.07155

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[53] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[55] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[56] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Feifei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[57] P. Goyal *et al.*, "Accurate, large minibatch SGD: Training imagenet in 1 hour," 2017, *arXiv:1706.02677*. [Online]. Available: http://arxiv.org/abs/1706.02677

[58] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[59] X. Li, X. Hu, and J. Yang, "Spatial group-wise enhance: Improving semantic feature learning in convolutional networks," 2019, *arXiv:1905.09646*. [Online]. Available: http://arxiv.org/abs/1905.09646

[60] O. Ulutan, A. S. M. Iftekhar, and B. S. Manjunath, "VSGNet: Spatial attention network for detecting human object interactions using graph convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13617–13626.

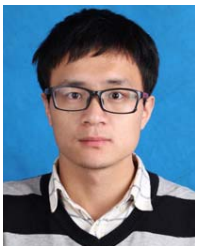
[61] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to detect human-object interactions," in *Proc. WACV*, 2018, pp. 381–389.

[62] G. Gkioxari, R. Girshick, P. Dollar, and K. He, "Detecting and recognizing human-object interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8359–8367.

[63] C. Gao, Y. Zou, and J.-B. Huang, "ICAN: Instance-centric attention network for human-object interaction detection," in *Proc. BMVC*, 2018, pp. 1–13.

Dongsheng Ruan is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Zhejiang, China. His current research interests include computer vision and deep learning.

Yu Shi is currently an Attending Physician with the Department of Infectious Diseases, The First Affiliated Hospital, Zhejiang University. He is also a Research Scientist with the State Key Laboratory for Diagnosis and Treatment of Infectious Diseases, Collaborative Innovation Center for Diagnosis and Treatment of Infectious Diseases and the National Clinical Research Center for Infectious Diseases. His major interests of studies are the natural history and pathogenesis of viral hepatitis, liver cirrhosis, and liver failure, especially related to hepatitis B virus.



Jun Wen is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University. His research interests include transfer learning and healthcare data mining.

Nenggan Zheng (Senior Member, IEEE) received the bachelor's and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2002 and 2009, respectively. He is currently a Full Professor in computer science with the Academy for Advanced Studies, Zhejiang University. His research interests include artificial intelligence, brain–computer interface, and embedded systems.

Min Zheng is currently the Deputy Director of the State Key Laboratory for Diagnosis and Treatment of Infectious Diseases and the Vice President of The First Affiliated Hospital, Zhejiang University, Zhejiang, China. Her current research interests include pathogenesis of viral hepatitis B, and precision diagnosis and treatment, including the applications of AI in clinical diagnosis.