Spatial Group-wise Enhance: Enhancing Semantic Feature Learning in CNN

Yuxuan Li[1], Xiang Li[1]*, and Jian Yang[1]

Nankai University, 38 Tongyan Road, Jinnan District, Tianjin, P.R.China 300350

Abstract. The success of attention modules in CNN has attracted increasing and widespread attention over the past years. However, most existing attention modules fail to consider two important factors: (1) For images, different semantic entities are located in different areas, thus they should be associated with different spatial attention masks; (2) most existing framework exploits individual local or global information to guide the generation of attention masks, which ignores the joint information of local-global similarities that can be more effective. To explore these two ingredients, we propose the Spatial Group-wise Enhance (SGE) module. SGE explicitly distributes different but accurate spatial attention masks for various semantics, through the guidance of local-global similarities inside each individual semantic feature group. Furthermore, SGE is lightweight with *almost no extra parameters and calculations*. Despite being trained with only category supervisions, SGE is effective in highlighting multiple active areas with various high-level semantics (such as the dog's eyes, nose, etc.). When integrated with popular CNN backbones, SGE can significantly boost their performance on image recognition tasks. Specifically, based on ResNet101 backbones, SGE improves the baseline by 0.7% Top-1 accuracy on ImageNet classification and 1.6∼1.8% AP on COCO detection tasks. The code and pretrained models are available at https://github.com/implus/PytorchInsight.

Keywords: Computer Vision · Backbone · Attention Mechanism.

1 Introduction

Recently, attention mechanisms have become extremely popular in convolutional neural networks. SENet [1] first proposes feature recalibration using the global information in a channel-wise manner. Subsequently, more works [2, 3] extend the recalibration to the spatial dimension, enabling the attention factors to be spatially redistributed. Despite their great success, there are at least two aspects have been ignored by most existing work, which limits the rationality and effectiveness of attention modules:

For Spatial Attention Modeling: The natural image usually contains multiple semantic objects distributed in different image regions. However, almost all the existing spatial attention modules [2–4] only use one single global spatial

* Corresponding author

attention mask, which obviously has no way to reasonably reflect the spatial distribution of different semantic features.

For Attention Mask Generation: Existing attention modules strive to guide the generation of the attention mask by utilizing global [1, 5, 6, 2, 3, 7, 4], or local [7, 8], or local-local pair [9, 10] information, but unfortunately lose the chances of gaining benefits from the joint information of the local-global pairs.

In this paper, we aim to propose a novel attention mechanism by taking into account the two factors:

For the first factor, inspired by the CapsuleNet [11] where the *grouped sub-features* can represent the instantiation parameters of a specific type of entity, we propose a group-wise attention mechanism. To be specific, the feature vector is first divided into groups, which are supposed to be learnt with multiple semantics (similarly as Capsules do). Then different spatial attention masks are designed and generated between different semantic feature groups, in the purpose of achieving a more *reasonable and explainable* spatial attention modeling.

For the second factor, in order to fully utilize both global and local information, and to lighten the complexity of the designed module as much as possible, we propose to use the similarity between the global feature descriptor and the local feature vector to guide the generation of the attention mask, which introduces rich information from local-global pairs.

To combine both factors above, the two solutions are merged naturally and completely into a unified framework by requiring *almost no additional parameters and calculations*, which is termed Spatial Group-wise Enhance (SGE) module.

We show on the ImageNet [12]benchmark that the SGE module performs better or comparable to a series of recently proposed state-of-the-art attention modules, despite its superiority in both model capacity and complexity. Similar trend is also observed on smaller dataset like CIFAR-100 [13]. Meanwhile, based on ResNet101 [14] backbones, SGE improves the baseline by 0.7% Top-1 accuracy on ImageNet classification and 1.6∼1.8% AP on COCO detection tasks, which demonstrates its remarkable advantages in accurate spatial modeling.

In the ablation study, we show that both solutions of the two factors play an important role for improving the final performance. We also examine the changes in the distribution of the semantic feature activations for each group after the SGE module. The results show that SGE significanty improves the spatial distribution of different semantic sub-features within its group, which strengthens the feature learning in semantic regions and compresses the possible noise and interference. The visualization of activation maps by Grad-CAM [15] also shows that SGE is able to make better use of accurate spatial features.

2 Related Work

Spatial Attention Modeling: In this part, we mainly focus on spatial attention mechanism, where the exsiting work mainly generates a *single* spatial mask for the entire tensor. BAM [2] and CBAM [3] utilize the convolutional layers or

Table 1. Summary of major differences among popular lightweight attention modules. The additional costs comprehensively consider the situation of multiple backbones.

Features	**SGE (ours)**	SE	SK	SRM	GE	BAM	CBAM	GC	GCT
Multiple Spatial Attention Mask	✓				✓(local version)				
Spatial Attention on *Feature Vectors*	✓								
Global Feature for Attention Generation	✓	✓	✓	✓	✓(global version)	✓	✓	✓	✓
Local Feature for Attention Generation	✓				✓(local version)				
Additional Parameter Cost $< \sim 1‰$	✓				✓(GE-θ^-)				
Additional FLOPs Cost $< \sim 1‰$	✓	✓		✓	✓(GE-θ^-)		✓	✓	✓

channel-based max/avg pooling layer to produce a unified attention map for spatial refinement. GCNet [4] proposes a context modeling, where a convolutional layer is also utilized to produce one spatial mask. The variants of GENet [7] with local extent ratio can be regarded as that each channel has its own attention spatial mask obtained by local information. However, [11] shows that a single scalar is difficult to characterize a semantic entity well, and local attention is also very limited in terms of semantic enhancement. Conversely, the proposed SGE explicitly assign different spatial attention masks in different semantic feature groups, leading to accurate feature enhancement.

Attention Mask Generation: The existing methods can be mainly attributed into the following three groups:

• **global only**: A series of work like SENet[1], SKNet[5], SRM[6], GCT[16], BAM [2], CBAM[3] and ECANet[17] performs feature recalibration via the guidance of global averaged statistics. The gather operator in GENet[7] aggregates neuron responses over a given spatial extent to guide the production of the refined tensor. Among the different parameter-free versions of GENet (GE-θ^-), the one with global extent ratio achieves the best performance. Different from the global average operator, GCNet[4] utilizes the context modeling block to weighted average the global statistics. FcaNet[18] decomposes channel features in the frequency domain and utilizes multi-frequency components with the selected DCT bases to replace global average pooling. Instead of squeezing a 3D feature tensor into a single feature vector, Coordinate Attention Network[19] and Triplet Attention Network[20] utilize global pooling along height and width dimensions separately to capture fine-grained global spatial attention.

• **local only**: Residual Attention Network[8] constructs a light encoder-decoder architecture between stages to utilize the local spatial information for generating attention masks. The variants of GENet[7] with local extent ratio aggregate the local spatial neuron responses to produce the refined feature tensors. SCNet[21] utilizes a self-calibration branch to allow local spatial information adaptively interact with its surrounding context.

• **local-local pair only**: [9] gives a thorough study of spatial attention mechanisms designed for broad application, where four types of attention terms are investigated in different combinations of context/position encodings of dense key-query pairs[22]. Such a key-query pair essentially reflects the property of local-local pairs. Another representative structure based on local-local pairs is

Non-Local[10] Network, which aims at strengthening the features of the query position via aggregating information from all other positions. However, the time and space complexity of the Non-Local blocks are both quadratic to the number of positions, which are considerably heavy for lightweight modules.

In contrast, our proposed SGE module explores a novel and rich guidance which is generally ignored by the related work: the local-global similarity. Such operator can not only make good use of both global and local information, but also utilize the advantage of the joint statistics between them. Compared to other attention modules, SGE also has fewer parameters, less computational complexity (Table 2), and a clear interpretable mechanism (Figure 3). Table 1 summarizes the essential differences between SGE and other existing lightweight attention modules for better reference.

Grouped Features: Learning and distributing features into groups in convolutional networks has been widely studied recently. AlexNet[23] initially presents the group convolution and divides features into two groups on different GPUs to save computing budgets. ResNeXt[24] examines the importance of grouping in feature transfer and suggests that the number of groups should be increased to obtain higher accuracy under similar model complexity. The MobileNet series[25–27] and Xception[28] treat each channel as a group and model only spatial relationships inside these groups. The ShuffleNet[29, 30] family rearranges the grouped features to produce efficient feature representation. Res2Net[31] uses a hierarchical mode to transfer grouped sub-features, enabling the network to incorporate multi-scale features in a single bottleneck. CapsuleNet[11] models each of the grouped neurons as a capsule, where the activities of the neurons within an active capsule represent the various properties of a particular entity that is present in the image. The overall length of the vector of instantiation parameters is used to represent the existence of the entity and the orientation of the vector is forced to represent the properties of the entity. In SGE, all enhancements are operated inside groups, which saves computational overhead similarly as in group convolution. Conceptually, the SGE module adopts the basic modeling assumptions of CapsuleNet, and believes that the features of each group are able to actively learn various semantic entity representations. At the same time, in the process of visualization of this paper, we also use the length of the sub-feature to measure as its activation value, analogous to the probability of the existence of entities in CapsuleNet.

3 Method

Here we describe the detailed implementation of SGE module, which unifies the above aforementioned two solutions: various semantic spatial attention mask and local-global similarity guidance. We consider a C channel, $H \times W$ convolutional feature map and divide it into G groups along the channel dimension. Without loss of generality, we first examine a certain group separately (see the bottom black box in Figure 1). Then the group has a vector representation at every position in space, namely $\mathcal{X} = \{\mathbf{x}_{1\ldots m}\}, \mathbf{x}_i \in \mathbb{R}^{\frac{C}{G}}, m = H \times W$. Conceptually in-

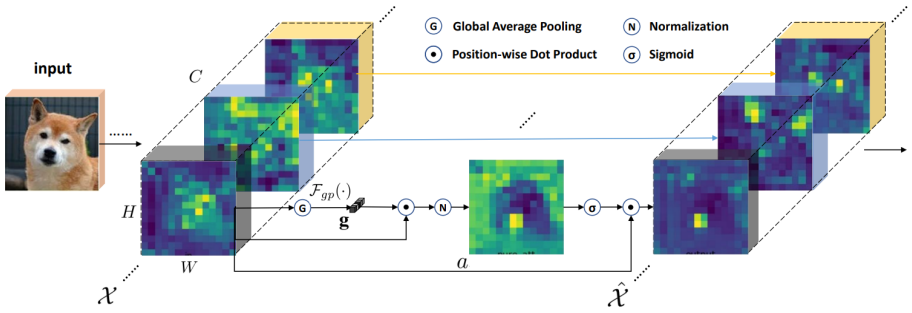

Fig. 1. Illustration of the proposed lightweight SGE module. It processes the sub-features of each group in parallel, and uses the similarity between global statistical feature and local positional features in each group as the attention guidance to enhance the features, thus obtaining well-distributed semantic feature representations in space.

spired by the capsules [11], we further assume that this group gradually captures a specific semantic response (such as the dog's eyes) during the course of network learning. In this group space, ideally we can get features with strong responses at the eye positions (i.e., features with a larger vector length and similar vector directions among multiple eye regions), whilst other positions almost have no activation and become zero vectors. However, due to the unavoidable noise and the existence of similar patterns, it is usually difficult for CNNs to obtain the well-distributed feature responses. We propose to utilize the overall information of the entire group space to further enhance the learning of semantic features in critical regions, given the fact that the features of the entire space are not dominated by noise (otherwise the model learns nothing from this group). Therefore we can use the global statistical feature through spatial averaging function $\mathcal{F}_{gp}(\cdot)$ to approximate the semantic vector that this group learns to represent:

$$\mathbf{g} = \mathcal{F}_{gp}(\mathcal{X}) = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i. \tag{1}$$

Next, using this global feature, we can generate the corresponding importance coefficient for each feature, which is obtained by simple dot product that measures the similarity between the global semantic feature $\mathbf{g}$ and local feature $\mathbf{x}_i$ to some extent. Thereby for each position, we have the following expression:

$$c_i = \mathbf{g} \cdot \mathbf{x}_i. \tag{2}$$

Note that c_i can also be expanded as $\|\mathbf{g}\|\|\mathbf{x}_i\|\cos(\theta_i)$, where θ_i is the angle between $\mathbf{g}$ and $\mathbf{x}_i$. It indicates that features that have a larger vector length (i.e., $\|\mathbf{x}_i\|$) and a direction (i.e., θ_i) closer to $\mathbf{g}$ are more likely to obtain a larger initial coefficient, which is in line with our assumptions. In order to prevent the biased magnitude of coefficients between various samples, we normalize c over

the space, as is widely practiced in [32–34]:

$$\hat{c}_i = \frac{c_i - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \quad \mu_c = \frac{1}{m}\sum_j^m c_j, \quad \sigma_c^2 = \frac{1}{m}\sum_j^m (c_j - \mu_c)^2, \tag{3}$$

where ϵ (e.g., 1e-5) is a constant added for numerical stability. To make sure that the normalization inserted in the network can represent the identity transform, we introduce a pair of parameters γ, β for each coefficient $\hat{c}_i$, which scale and shift the normalized value:

$$a_i = \gamma \hat{c}_i + \beta. \tag{4}$$

Note that γ, β here are the only parameters introduced in our module. In a single SGE unit, the number of γ, β is the same as the number of groups G, and the order of their magnitude is about tens (typically, 32 or 64), which is basically *negligible* compared to the millions of parameters of the entire network. Finally, to obtain the enhanced feature vector $\hat{\mathbf{x}}_i$, the original $\mathbf{x}_i$ is scaled by the generated importance coefficients a_i via a sigmoid function gate $\sigma(\cdot)$ over the space:

$$\hat{\mathbf{x}}_i = \mathbf{x}_i \cdot \sigma(a_i), \tag{5}$$

and all the enhanced features form the resulted feature group as

$$\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_{1\ldots m}\}, \hat{\mathbf{x}}_i \in \mathbb{R}^{\frac{C}{G}}, m = H \times W. \tag{6}$$

4 Experiments

4.1 Image Classification

We first compare SGE with a set of SOTA attention modules on ImageNet benchmark. The ImageNet 2012 dataset[12] comprises 1.28 million training images and 50k validation images from 1k classes. We train networks on the training set and report the Top-1 and Top-5 accuracies on the validation set with single 224×224 central crop. For data augmentation, we follow the standard practice[35] and perform the random-size cropping to 224 × 224 and random horizontal flipping. The practical mean channel subtraction is adopted to normalize the input images. All networks are trained with naive softmax cross entropy without label-smoothing regularization[36]. We train all the architectures from scratch by synchronous SGD with weight decay 0.0001 and momentum 0.9 for 100 epochs, starting from learning rate 0.1 and decreasing it by a factor of 10 every 30 epochs. The total batch size is set as 256 and 8 GPUs (32 images per GPU) are utilized for training, using the weight initialization strategy in[37]. Our codes are implemented in the pytorch[38] framework in which all results are reproduced. Note that in the following tables, Param. denotes the number of parameter and the definition of FLOPs follows[29], i.e., the number of multiply-adds.

Comparisons with state-of-the-art Attention Modules. We select a series of state-of-the-art attention modules, which is considered to be relatively

Table 2. Comparisons between various guidance for spatial attention mask generation on ImageNet validation set, based on ResNet50. The best and the second best records are marked as **red** and **blue**, respectively.

Backbone	Param.	GFLOPs	Top-1 (%)	Top-5 (%)
ResNet50 [14]	25.56M	4.122	76.38	92.91
SE-ResNet50 [1]	28.09M	**4.130**	77.18	**93.67**
SK-ResNet50* [5]	26.15M	4.185	77.54	**93.70**
BAM-ResNet50 [2]	25.92M	4.205	76.90	93.40
CBAM-ResNet50 [3]	28.09M	4.139	**77.63**	93.66
SRM-ResNet50 [6]	**25.62M**	4.139	77.13	93.51
GCT-ResNet50 [16]	25.68M	4.134	77.30	**93.70**
GE-ResNet50 [7]	**25.56M**	**4.127**	76.78	93.22
SGE-ResNet50 **(ours)**	**25.56M**	**4.127**	**77.58**	93.66
ResNet101 [14]	44.55M	7.849	78.20	93.91
SE-ResNet101 [1]	49.33M	**7.863**	78.47	94.10
SK-ResNet101* [5]	45.68M	7.978	**78.79**	**94.27**
BAM-ResNet101 [2]	44.91M	7.933	78.22	94.02
CBAM-ResNet101 [3]	49.33M	7.879	78.35	94.06
SRM-ResNet101 [6]	**44.68M**	7.879	78.47	94.20
GCT-ResNet101 [16]	44.76M	7.869	78.60	94.10
GE-ResNet101 [7]	**44.55M**	**7.858**	78.42	94.14
SGE-ResNet101 **(ours)**	**44.55M**	**7.858**	**78.90**	**94.37**
ResNeXt50 [24]	25.03M	4.273	77.15	93.52
SE-ResNeXt50 [1]	27.56M	**4.281**	78.09	93.96
SK-ResNeXt50 [5]	27.42M	4.505	**78.21**	**94.07**
BAM-ResNeXt50 [2]	25.39M	4.356	77.44	93.60
CBAM-ResNeXt50 [3]	27.56M	4.290	78.08	94.05
GCT-ResNeXt50 [16]	**25.19M**	4.285	78.20	94.00
GE-ResNeXt50 [7]	**25.03M**	**4.279**	77.48	93.69
SGE-ResNeXt50 **(ours)**	**25.03M**	**4.279**	**78.25**	**94.09**
DenseNet121 [39]	7.98M	2.883	75.36	92.60
SE-DenseNet121 [1]	**7.99M**	**2.884**	**76.21**	93.00
SK-DenseNet121* [5]	8.10M	2.930	75.83	92.88
BAM-DenseNet121 [2]	8.07M	2.904	76.20	**93.01**
CBAM-DenseNet121 [3]	**7.99M**	**2.886**	76.10	92.78
GE-DenseNet121 [7]	**7.98M**	**2.884**	76.18	92.88
SGE-DenseNet121 **(ours)**	**7.98M**	**2.884**	**76.45**	**93.06**

lightweight, and demonstrate their performance based on ResNet50, ResNet101 [14, 40], ResNeXt50 [24] and DenseNet121 [39]. For a fair comparison, we implement all the attention modules (partially refer to the official codes[1]) with their respective best settings using a unified pytorch framework. Following [1, 3], these attention modules are placed after the last BatchNorm [32] layer inside each bottleneck except for BAM and SK. BAM [2] is naturally designed between stages. SK [5] is originally designed on ResNeXt-like bottlenecks with multiple large-kernel group convolutions. To transfer it to the ResNet/DenseNet backbones, we make a slight modification and only append one additional 3×3 group ($G = 32$) convolution upon each original 3×3 convolutions, to prevent the parameters and calculations of the corresponding SKNets from being too large or too small. For GE, we select the best performed parameter-free settings with global extent ratio, namely GE-θ^-, for comparisons (the other variants increase the number of parameters too much). From the results of Table 2, we observe that based on ResNet50, SGE is on par with the best entries from CBAM (Top-1) and SK/SE (Top-5) but has much fewer parameters and slightly less calculations. As for ResNet101, it outperforms most other competing modules. The similar trend is also hold for ResNeXt50 [24] and DenseNet121 [39].

The Effectiveness of Local-Global Similarities. To validate the effectiveness of local-global similarities, we conduct extensive experiments by comparing SGE with global-only and local-only variants of the state-of-the-art SE and GE modules. Specifically in Table 3, to keep the comparisons more fair under the settings of multiple spatial semantics in global-only type, we extend SE with group settings (denoted as SE*), where the fc layers are replaced by group conv1x1 layers with group number G. We also extend GE-θ^- as GE-θ^{-*} with groups. Considering the parameter-free settings of GE-θ^-, we simply average the elements in each group of the global pooled vector to reweight the activations. For the modified group versions of SE* and GE-θ^{-*}, we choose the two settings G=32 and G=64 for experiments. In local-only type, we select the GE modules with spatial extent ratio e=8. Furthermore, we validate the importance of local-global similarities by deleting the similarity part but only using the length of each local sub-feature itself to guide the attention generation in SGE, which is denoted as SGE (- similarity). For the comparisons with local-local pairs, four variants of Non-Local [10] blocks are applied. As the module adds a lot of extra complexity, it is forced to place only one instance on the last stage of ResNet50. From the above results, we notice that the joint information of local-global similarities is considerably efficient and beneficial for achieving the best performance.

Group number G. In the SGE module, the number of groups G controls the number of semantic sub-features. Too many groups will result in a reduction in the sub-feature dimension within each group, leading to weaker feature representation for each semantic response; On the contrary, too few groups will make the diversity of semantics limited. It is natural to speculate that there is a moderate hyperparameter G that balances semantic diversity and the ability of representing each semantic to optimize network performance. From Figure 2, we can see

[1] https://github.com/Jongchan/attention-module

Table 3. Comparisons between various guidance for spatial attention mask generation on ImageNet validation set, based on ResNet50. The best records are marked as **bold**.

Type	Backbone	Param.	GFLOPs	Top-1/5 (%)
global-only	+ SE [1]	28.09M	4.130	77.2/93.7
	+ SE* (G=32)	26.20M	4.128	77.2/93.6
	+ SE* (G=64)	25.89M	4.128	77.0/93.5
	+ GE-θ^- [7]	25.56M	4.127	76.8/93.2
	+ GE-θ^{-*} (G=32)	25.56M	4.127	76.6/93.2
	+ GE-θ^{-*} (G=64)	25.56M	4.127	76.7/93.4
local-only	+ GE-θ^- (e=8)	25.56M	4.127	76.5/93.1
	+ GE-θ^{-*} (G=64, e=8)	25.56M	4.127	76.5/93.2
	+ SGE (- similarity)	25.56M	4.125	77.0/93.5
local-local pair	+ Non-Local [10] (Gaussian)	29.76M	4.328	75.8/92.7
	+ Non-Local [10] (Embedded Gaussian)	33.95M	4.534	75.6/92.6
	+ Non-Local [10] (Dot Product)	33.95M	4.534	76.2/92.8
	+ Non-Local [10] (Concatenation)	33.96M	4.534	76.2/92.9
local-global pair	+ SGE **(ours)**	25.56M	4.127	**77.6/93.7**

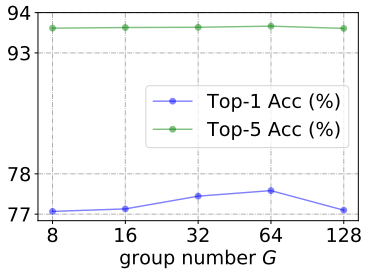

Fig. 2. Performance of SGE-ResNet50 w.r.t. group number G.

that with the increase of G, the performance of the network shows a trend of increasing first and then decreasing (especially in terms of Top-1 accuracy), which is highly consistent with our deduction. Through the experimental results, we recommend the group number G to be 32 or 64. In subsequent experiments, we use $G = 64$ by default.

Initialization of the γ and β. During the experiment, we find that the initialization of the parameter γ and β has a not negligible effect on the result. We use values 0, 1 for grid search to see the effects of the initialization. From Table 4 we find that initializing γ to 0 tends to get better results. We speculate that when the ordinary patterns of semantic learning has not yet been completely formulated in convolutional feature maps during the initial stage of network training, it may be appropriate to temporarily discard the attention mechanism, but let the network learn a basic semantic representation first. After the initial training period, the attention modules then need to be gradually turned in effect.

Table 4. Performance of SGE-ResNet50 as a function of initializations of γ and β.

γ	β	Top-1 (%)	Top-5 (%)
0	0	77.38	**93.71**
0	1	**77.58**	93.66
1	0	77.22	93.58
1	1	77.08	93.70

Table 5. Performance of SGE-ResNet50 with/without the normalization part.

Type	Top-1 (%)	Top-5 (%)
w/ Normalization	**77.58**	**93.66**
w/o Normalization	76.50	93.16

Table 6. Comparisons to the state-of-the-art attention modules on CIFAR-100 test set. The best and the second best records are marked as **red** and **blue**, respectively.

Backbone	Param.	GFLOPs	Top-1 (%)
ResNet50 [14]	23.71M	1.306	78.06
SE [1]	26.24M	1.310	78.96
SK* [5]	24.30M	1.329	79.42
BAM [2]	24.06M	1.335	79.35
CBAM [3]	26.24M	1.317	78.44
SRM [6]	23.77M	1.316	78.62
GCT [16]	23.75M	1.312	79.10
GE [7]	23.71M	1.310	78.83
SGE **(ours)**	23.71M	1.310	79.47

Therefore, in the early moments of network learning, the attention mechanism of SGE is not suggested to participate heavily in training by setting γ to 0. Such an operation is almost equivalent to simulate the learning process of a network without attention modules during the very early training stage, since each sub-feature of each location is linearly multiplied by the same constant (i.e., $\sigma(\beta)$), whose effect can be cancelled by the following BatchNorm layer.

Normalization. To investigate the importance of normalization in SGE modules, we conduct experiments by eliminating the normalization part from SGE (as shown in Table 5) and find that performance is considerably reduced. The central reason is that the variance of the activation values of different samples in the same group can be statistically very different, indicating that normalization is essential for SGE to work.

Image Classification on CIFAR-100. We also compare SGE with a set of SOTA modules on the 32x32 image dataset CIFAR-100 [13] benchmark. We perform random cropping on images with 4-pixel padding, random horizontal flipping and random rotation with 15 degrees. We train networks on the train

set and report the Top-1 accuracy on the test set. We adopt a standard training strategy as stated in [41]. Total batch size is set as 128. From the results in Table 6, we observe that based on the ResNet50 backbones, the SGE outperforms all other competing modules in Top-1 classification accuracy, with minimal parameters and relatively lowest computations. SGE's good performance on small image dataset demonstrates its robustness to the scale of the input images.

4.2 Object Detection

We further evaluate the SGE module on object detection on COCO 2017 [42], whose train set is comprised of 118k images, validation set of 5k images. We follow the standard setting [43] of evaluating object detection via the standard mean Average-Precision (AP) scores at different box IoUs or object scales, respectively. The input images are resized with their shorter side being 800 pixels [44]. We train on 8 GPUs with 2 images per each. The backbones of all models are pretrained on ImageNet [12] (directly borrowed from the models listed in Table 2), then all layers except for the first two stages are jointly finetuned with FPN [44] neck and a set of detector heads. Following the conventional finetuning setting [43], the BatchNorm layers are frozen during finetuning. All models are trained for 24 epochs using synchronized SGD with a weight decay of 0.0001 and momentum of 0.9. The learning rate is initialized to 0.02, and decays by a factor of 10 at the 18th and 22nd epochs. The choice of hyper-parameters follows the latest release of the detection benchmark [45].

Table 7. $AP_{50:95}$ (%) scores via embedding SGE on the backbones of state-of-the-art detectors on COCO [42] dataset. The best records are marked as **bold**.

Backbone	Param.	GFLOPs	Retina [46]	Faster [47]	Mask [43]	Cascade [48]
ResNet50	23.51M	88.0	36.4	37.5	38.6	41.1
+ SGE	23.51M	88.1	**37.5**	**38.7**	**39.6**	**42.6**
ResNet101	42.50M	167.9	38.1	39.4	40.4	42.6
+ SGE	42.50M	168.1	**38.9**	**41.0**	**42.1**	**44.4**

Experiments on state-of-the-art Detectors. We embed the SGE modules into the popular detector framework separately to check if the enhanced feature map helps to detect objects. We select four popular detection frameworks, including RetinaNet [46], Faster RCNN [47], Mask RCNN [43], and Cascade RCNN [48], and choose the widely used FPN [44] as the detection neck. For a fair comparison, we only replace the pretrained backbone model on ImageNet while keeping the other components in the entire detector intact. Table 7 shows the performance of embedding the backbone with the SGE module on these state-of-the-art detectors. We find that although SGE introduces almost no additional parameters and calculations, the gain of detection performance is still very noticeable with basically more than 1% AP point. It is worth noting that

Table 8. Various AP (%) comparisons based on the state-of-the-art detectors (Faster [47]/Mask [43]/Cascade [48] RCNN) and backbone ResNet101 [14] on COCO [42] dataset. The Parm. and GFLOPs are only with the backbone parts, given that all the remaining structures are kept the same. The numbers in brackets denote the improvements over the baseline backbones. The best records are marked as **bold**.

Backbone	Param.	GFLOPs	Detector	$AP_{50:95}$	AP_{50}	AP_{75}	AP_{small}	AP_{media}	AP_{large}
ResNet101 [14]	42.5M	167.9	Faster	39.4	60.7	43.0	22.1	43.6	52.1
+ SE [1]	47.3M	168.3	Faster	40.4(+1.0)	61.9	44.2	23.7(+1.6)	44.5	51.9
+ CBAM [3]	47.3M	168.5	Faster	40.1(+0.7)	61.9	43.6	23.3(+1.2)	44.5	51.2
+ GC(r16) [4]	47.3M	168.3	Faster	40.3(+0.9)	62.1	43.8	23.4(+1.3)	44.8	51.8
+ GE(-θ^-) [7]	**42.5M**	**168.1**	Faster	39.5(+0.1)	61.2	43.4	23.2(+1.1)	44.4	50.5
+ SGE	**42.5M**	**168.1**	Faster	**41.0(+1.6)**	**63.0**	**44.3**	**24.5(+2.4)**	**45.1**	**52.9**
ResNet101 [14]	42.5M	167.9	Mask	40.4	61.6	44.2	22.3	44.8	52.9
+ SE [1]	47.3M	168.3	Mask	41.5(+1.1)	63.0	45.3	23.8(+1.5)	45.5	54.7
+ CBAM [3]	47.3M	168.5	Mask	41.2(+0.8)	62.9	44.8	24.6(+2.3)	45.5	53.1
+ GC(r16) [4]	47.3M	168.3	Mask	41.6(+1.2)	63.2	45.6	24.7(+2.4)	45.8	53.8
+ GE(-θ^-) [7]	**42.5M**	**168.1**	Mask	40.6(+0.2)	62.5	44.0	24.0(+1.7)	45.2	52.8
+ SGE	**42.5M**	**168.1**	Mask	**42.1(+1.7)**	**63.7**	**46.1**	**24.8(+2.5)**	**46.6**	**55.1**
ResNet101 [14]	42.5M	167.9	Cascade	42.6	60.9	46.4	23.7	46.1	56.9
+ SE [1]	47.3M	168.3	Cascade	43.4(+0.8)	62.2	47.2	24.1(+0.4)	47.5	57.9
+ CBAM [3]	47.3M	168.5	Cascade	43.3(+0.7)	62.1	47.1	24.5(+0.8)	47.4	57.7
+ GC(r16) [4]	47.3M	168.3	Cascade	43.4(+0.8)	62.2	47.4	24.8(+1.1)	47.4	57.9
+ GE(-θ^-) [7]	**42.5M**	**168.1**	Cascade	42.8(+0.2)	61.8	46.5	24.1(+0.4)	47.0	57.2
+ SGE	**42.5M**	**168.1**	Cascade	**44.4(+1.8)**	**63.2**	**48.4**	**25.7(+2.0)**	**48.3**	**58.7**

SGE can be more prominently advanced on stronger detectors (**+1.5%** AP on ResNet50 and **+1.8%** on ResNet101 in Cascade RCNN).

Comparisons with state-of-the-art Attention Modules. Next, based on backbone ResNet101, we compare SGE with several representative strong attention modules on various competitive state-of-the-art detectors, and report the detailed AP scores including the metrics over three different scales. The original backbones are replaced with the corresponding attention embedded ResNets, which are pretrained on ImageNet. In Table8, thanks to the enhancement of critical regions, SGE greatly improves the accuracy of detection for small objects ($> 2\%$ absolute AP gain) while its performance of the media and large objects still significantly competitive. This is consistent with our visualization in Figure 3, which demonstrates that the SGE module is able to retain the feature representation of the spatial region well. Conversely for the others, in each channel, they give the same importance coefficient for every single location, resulting in a loss of the expression of the micro-region to some extent. In the case of general metric $AP_{50:95}$, SGE outperforms the popular SE by a considerably nonnegligible margin, including 0.6% absolute improvement on Faster/Mask RCNN and 1% on Cascade RCNN.

4.3 Visualization and Interpretation

In order to verify that our approach achieves the goal of improving the semantic feature representation, we first demonstrate *several examples* with specific se-

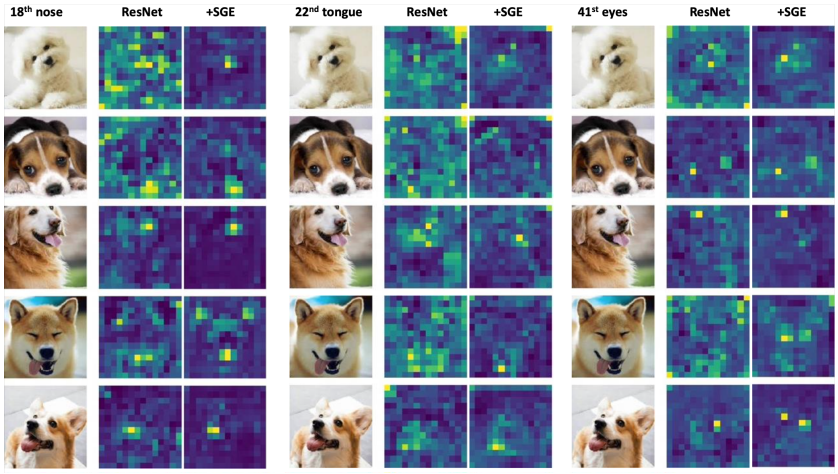

Fig. 3. We select several feature groups with representative semantics to display before and after using SGE on ResNet50. We sample images of different shapes, categories, and angles to verify the robustness of the SGE module.

mantic visual clues (in Figure 3) and show how SGE helps to improve detection accuracy especially in small objects (in Figure 4).

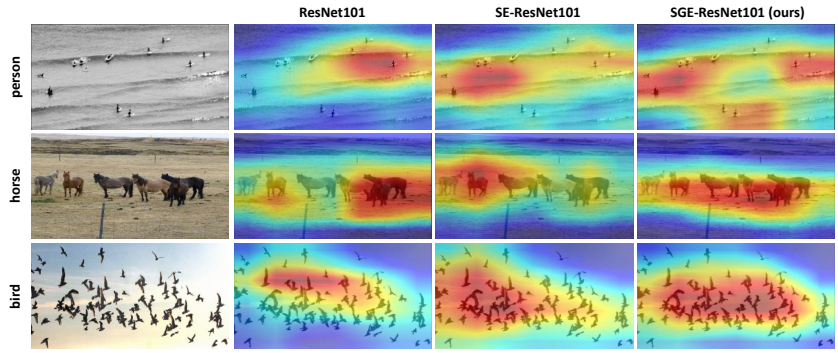

Fig. 4. Grad-CAM [15] visualization results for detection backbone. We compare the visualization results of SE-ResNet101 and SGE-ResNet101 with the ResNet101 baseline. It is clear that our SGE module shows good coverage of target semantic spatial features than other counterparts.

Visualization of Different Semantic Enhancement. We train a network based on ResNet50 on ImageNet [12] and place the SGE module after the last

BatchNorm [32] layer of each bottleneck with reference to SENet [1], by setting $G = 64$. To better reflect the semantic information while preserving the large spatial resolution as much as possible, we choose to examine the feature maps of the 4th stage with output size of 14×14. For each feature vector of each group, we use its length (i.e., $\|\mathbf{x}_i\|$) to indicate their activation value and linearly normalize it to the interval $[0, 1]$ for a better view. Figure 3 shows three representative groups with semantic responses. As listed in three large columns, they are the 18th, 22nd, and 41st group, which are empirically found to correspond to the concept of the nose, tongue, and eyes. Each large column contains three small columns, where the first small column is the original image, the second small column is the feature map response from the original ResNet50, and the third one is the feature map response enhanced by the SGE module. We select images of dogs of different angles and types to test the robustness of SGE for feature enhancement. Despite its simplicity, the SGE module is very effective in improving the feature representation of specific semantics at corresponding locations while suppressing a large amount of noise. It is worth noting that in the 4th and 7th rows, SGE can strongly emphasize the activation of the eye areas, although their eyes are almost closed. In contrast, the original ResNet fails to capture such patterns.

Activation Map for Detecting Objects. We apply Grad-CAM [15] to several backbones using the images from COCO test set. Grad-CAM can explicitly emphasize the critical regions for semantic feature representations through the gradient guidance. As the regions are considered as important clues for the network to predict correctly, we attempt to judge how the model is making good use of image features. From Figure 4, thanks to the explicit spatial enhancement mechanism, SGE module is able to cover more critical and accurate locations for semantic expressions, which clearly explains why the detection performance of small or middle objects could be boosted significantly as show in Table 8.

5 Conclusion

To explore the two missing ingredients for attention mechanism in CNN: multiple spatial semantics and local-global similarites, we propose a Spatial Group-wise Enhance (SGE) module that enables each of its feature groups to enhance the learnt semantic representation, guided by its respective local-global similarities. SGE is designed nearly without introducing additional parameters and computational complexity. We visually show that the feature groups have the ability to express different semantics, while the SGE module can significantly enhance this ability. Despite its simplicity, SGE has achieved a steady improvement in both image classification and detection tasks, which demonstrates its compelling effectiveness in practice.

References

1. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. (2018)

2. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514 (2018)
3. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. arXiv preprint arXiv:1807.06521 (2018)
4. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. arXiv preprint arXiv:1904.11492 (2019)
5. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: CVPR. (2019)
6. Lee, H., Kim, H.E., Nam, H.: Srm: A style-based recalibration module for convolutional neural networks. arXiv preprint arXiv:1903.10829 (2019)
7. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: Exploiting feature context in convolutional neural networks. In: NeurIPs. (2018)
8. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. arXiv preprint arXiv:1704.06904 (2017)
9. Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J.: An empirical study of spatial attention mechanisms in deep networks. arXiv preprint arXiv:1904.05873 (2019)
10. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR. (2018)
11. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: NeurIPs. (2017)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
13. Krizhevsky A, H.G.: Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases (2009)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
15. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: CVPR. (2017)
16. Yang Zongxin, Zhu Linchao, W.Y., Yi, Y.: Gated channel transformation for visual recognition. In: CVPR. (2020)
17. Hu, W.Q..B.W..P.Z..P.L..W.Z..Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. CVPR (2020)
18. Li, Q.Z..P.Z..F.W..X.: Fcanet: Frequency channel attention networks. ICCV (2021)
19. Hou, Qibin, D.Z., Feng, J.: Coordinate attention for efficient mobile network design. CVPR (2021)
20. Misra Diganta, Trikay Nalamada, A.U.A., Hou, Q.: Rotate to attend: Convolutional triplet attention module. WACV (2021)
21. Feng, L.J.J..Q.H..M.M.C..C.W..J.: Improving convolutional networks with self-calibrated convolutions. CVPR (2020)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPs. (2017)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPs. (2012)
24. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR. (2017)
25. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

26. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR. (2018)
27. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. arXiv preprint arXiv:1905.02244 (2019)
28. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. CVPR (2017)
29. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. arxiv 2017. arXiv preprint arXiv:1707.01083 (2017)
30. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. arXiv preprint arXiv:1807.11164 (2018)
31. Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.: Res2net: A new multi-scale backbone architecture. arXiv preprint arXiv:1904.01169 (2019)
32. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
33. Wu, Y., He, K.: Group normalization. In: ECCV. (2018)
34. Qiao, S., Wang, H., Liu, C., Shen, W., Yuille, A.: Weight standardization. arXiv preprint arXiv:1903.10520 (2019)
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015)
36. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. (2016)
37. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. (2015)
38. Paszke, A., Gross, S., Chintala, S., Chanan, G.: Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration (2017)
39. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. (2017)
40. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. (2016)
41. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. CoRR (2017)
42. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
43. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. (2017)
44. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. (2017)
45. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: mmdetection. https://github.com/open-mmlab/mmdetection (2018)
46. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. (2017)
47. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS. (2015)
48. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: CVPR. (2018)