Linear Context Transform Block

Dongsheng Ruan
Zhejiang University

Jun Wen
Zhejiang University

Nenggan Zheng
Zhejiang University

Abstract

Squeeze-and-Excitation (SE) block presents a channel attention mechanism for modeling the global context via explicitly capturing dependencies between channels. However, we still poorly understand for SE block. In this work, we first revisit the SE block and present a detailed empirical study of the relationship between global context and attention distribution, based on which we further propose a simple yet effective module. We call this module Linear Context Transform (LCT) block, which implicitly captures dependencies between channels and linearly transforms the global context of each channel. LCT block is extremely lightweight with negligible parameters and computations. Extensive experiments show that LCT block outperforms SE block in image classification on ImageNet and object detection/segmentation on COCO across many models. Moreover, we also demonstrate that LCT block can yield consistent performance gains for existing state-of-the-art detection architectures. For examples, LCT block brings 1.5∼1.7% AP^{bbox} and 1.0%∼1.2% AP^{mask} gains independently of the detector strength on COCO benchmark. We hope our work will provide a new insight into the channel attention mechanism.

1. Introduction

Attention mechanism has achieved remarkable success in a variety of recognition tasks such as semantic segmentation [40, 6, 20], image classification [9, 33, 17] and object detection [34, 41]. It is typically plugged into existing deep networks [13, 37, 31, 39, 15, 42, 14] to improve the representational power. One of the most prominent work is Squeeze-and-Excitation network (SENet) [17], which is a channel attention mechanism that aims to selectively emphasize informative channels and suppress less useful ones by explicitly modeling interdependencies between channels. SENet achieves significant performance gains across various models and has been applied in a variety of vision tasks[30, 25, 38, 14].

To dive into this attention mechanism, we are curious

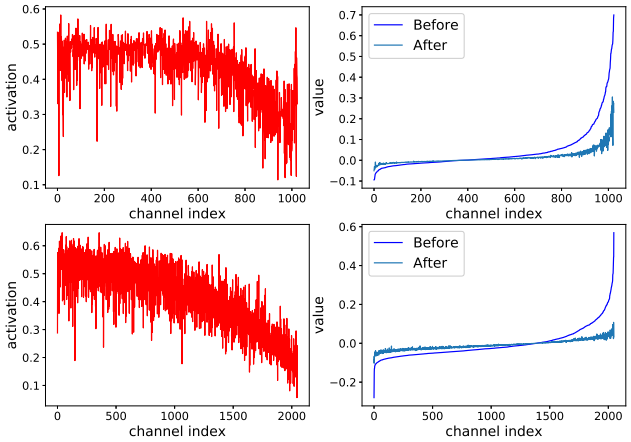

Figure 1. Visualization of average global context features before and after rescaling and average attention activations of the first block at different stages on ImageNet validation set. First column: average attention activations. Second column: average global context features before and after rescaling.

of the following two questions: 1) What is the relationship between global context and attention distribution? 2) Which channels are less useful? To answer these questions, we visualize the average global context features before and after rescaling and the corresponding attention activations on ImageNet validation set. For ease of observation, the average global context features before rescaling are sorted in ascending order, as shown in Fig. 1. Interestingly, we observe that SE block presents a negative correlation that global context features with larger absolute values tend to be attached with smaller attention activations, indicating that channels with these context features are generally less useful. By learning such a correlation, SE block effectively suppresses these channels and reduces the contextual differences between channels, which allows subsequent filters to extract useful semantic features, thereby improving the representational ability of the network. Given this observation, a question naturally arises: Can we lean such a correlation in a better way?

SENet has illustrated that explicitly modeling interdependencies between channels is effective. However, a po-

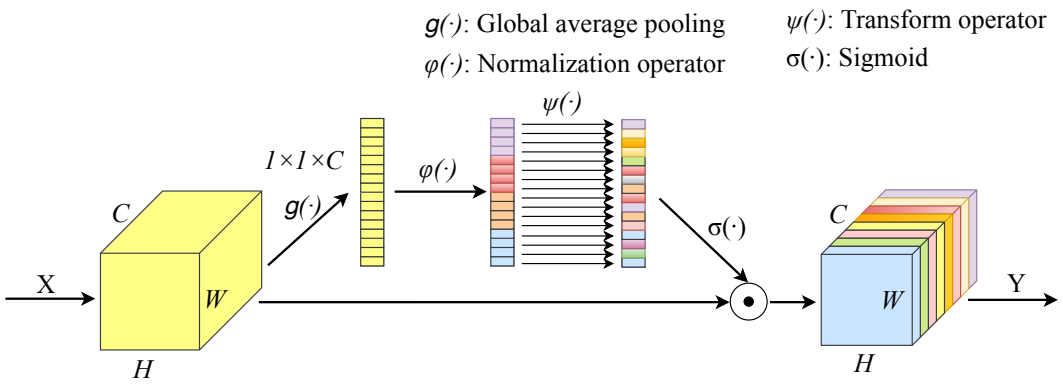

Figure 2. Architecture of linear context transform block. The input feature maps are defined as $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels and H, W are the spatial dimensions. $\mathbf{Y} \in \mathbb{R}^{C\times H\times W}$ denotes the output of LCT block. $\odot$ denotes broadcast element-wise multiplication. $\rightarrow$ denotes a linear transform.

tential problem is that when the dimension of the feature channel is high, modeling for entire channel dimension makes it difficult to learn such correlation accurately in that a lot of irrelevant information from other channels can be introduced. A feasible way is to boost the capacity of context feature transform module, as shown in GENet [16], but it will significantly increase the complexity of the model.

In this paper, we aim to explore a more efficient algorithm to learn the above negative correlation. To achieve this, we propose a simple yet effective module, called Linear Context Transform (LCT) block, which is extremely lightweight, adding almost negligible parameters and computations. Specifically, LCT block formulates the context feature transform as the composition of two cheap operators: a *normalization* operator, which normalizes the context features within each group, and a *transform* operator, which models the global context for each channel independently. Using this decomposition, LCT block can effectively exploit the global context. We empirically demonstrate that our approach outperforms SE block across different architectures and visual tasks. More importantly, we find that the roles of the fully connected (FC) layers in SE block are similar to those of the two operators in LCT block. We also investigate the differences between SE block and LCT block in terms of attention distribution and global contextual features. It is found that LCT block better learns the negative correlation and the learned attention distribution produces smaller fluctuations than that of SE block (Fig. 3). In summary, our main contributions can be summarized as follows:

- We present an empirical study of the relationship between global context and attention distribution of SENet. We find that there is a negative correlation between them, which helps researchers better understand channel-wise attention and facilitates further progress in this context.
- We propose a simple yet effective attention block (LCT) for global context modeling. To our knowledge, this is the first work to model global context for each channel independently.
- Comprehensive experiments on three visual recognition tasks (image classification on ImageNet and object detection/segmentation on COCO) demonstrate the superiority and generalization of our approach.

2. Related Work

Normalization Batch normalization (BN) [19] is a milestone technique that normalizes the statistics for each training mini-batch to stabilize the distributions of layer inputs. It enables deep networks to train faster and more stably. However, the property that depends on the mini-batch size leads to a rapid decline in network performance when the batch size becomes smaller. A series of normalization methods [1, 32, 36, 29] have been proposed to address this issue caused by inaccurate batch statistics estimation. Layer normalization (LN) [1] computes the statistics along the channel dimension and is well suited for recurrent neural network. Instance normalization [32] proposes to perform the normalization across spatial locations. Group normalization (GN) [36] divides the channel dimension into groups and normalizes the features in each group. Since GN does not exploit the batch dimension, it is still able to achieve high accuracy even in small batch size.

The design of LCT block is inspired by GN. Instead of stabilizing the distribution of layer inputs, LCT block is essentially an attention mechanism that aims to model the global context effectively.

Attention modules Recently, several attention modules [5, 34, 7, 8, 18] have been proposed to exploit the feature

context to enhance the representational power of the networks. In particular, SENet [17] develops a lightweight attention block to recalibrate feature channels by exciting the aggregated contexts from original features. Further, GENet [16] proposes a gather-excite framework for better context exploitation and yields further performance gains at the expense of increasing parameters. GCNet [3] combines simplified non-local block [34] and SE block [17] to effectively model the global context via addition fusion. In addition to channel attention, CBAM [35] and BAM [26] exploit both spatial and channel-wise information to yield further performance gains. SKNet [22] proposes a dynamic selection mechanism that enables the network to adaptively adjust receptive field. More recently, Li *et al.* [21] introduce a spatial group-wise enhance module to spatially enhance the semantic expression in each group, showing excellent performance in image classification and object detection.

Our work builds on the idea developed in SE block. However, different from SE block, LCT block implicitly captures channel-wise dependencies and linearly models the global context of each channel, which is more lightweight and effective.

3. Method

In this section, we first review the design of SE block, and then introduce our proposed linear context transform (LCT) block.

3.1. Revisiting the SE block

SE block aims at emphasizing informative features and suppressing less useful ones by modeling the channel relationship. To exploit contextual information, SE block proposes to squeeze global spatial information. Specifically, it aggregates global context information across spatial dimension by global average pooling operation. Further, to fully capture channel-wise dependencies, SE block excites the aggregated contexts using two FC layers. Here we define $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ as the input feature maps of SE block, where C is the number of channels and H, W are the spatial dimensions. The SE block can be written as

$$\mathbf{Y} = \mathbf{X}\cdot\sigma(f(g(\mathbf{X}))) = \mathbf{X}\cdot\sigma(\mathbf{W}_2 ReLU(\mathbf{W}_1 g(\mathbf{X}))), \quad (1)$$

where $\cdot$ denotes channel-wise multiplication and $g(\cdot)$ denotes global average pooling to generate channel-wise statistics. $\mathbf{W}_1$ and $\mathbf{W}_2$ denote the weights of FC layers. $\sigma(\cdot)$ denotes sigmoid function.

As shown in Fig. 1, SE block performs a non-linear transform to learn a negative correlation between global context features and attention activations by explicitly capturing interdependencies between channels. However, introducing irrelevant information from other channels makes global context modeling difficult, resulting in incorrect mapping. To tackle this problem, we propose LCT block.

3.2. Linear context transform block

Here we formally introduce LCT block and describe its operation as illustrated in Fig. 2.

As summarized in GCNet [3], global context modeling framework can be abstracted as three modules: (a) context aggregation; (b) context feature transform; (c) feature fusion. LCT block also follows this framework.

Context aggregation Context aggregation aims to help the network capture long-range dependency by exploiting information beyond the local receptive fields of each filter. A number of aggregation strategies can be chosen to aggregate contextual information, such as second-order attention pooling [7], global attention pooling [16, 3], and global average pooling [17]. Complex aggregation operators can be used to improve performance of LCT block, but are not the focus of our work. Hence we also simply employ global average pooling to aggregate the global context features across spatial dimensions generating a channel descriptor as $\mathbf{z} = \{z_k = \frac{1}{H\times W}\sum_{i=1}^{W}\sum_{j=1}^{H}\mathbf{X}_k(i,j) : k \in \{1,...,C\}\}$.

Context feature transform To effectively and efficiently model the context feature, LCT block introduces a pair of lightweight operators: a normalization operator, which normalizes the global context features in each group, and a transform operator, which takes in the normalized global contexts to produce the importance scores. Specifically, we first divide the descriptor $\mathbf{z}$ into groups and then normalize it within each group along channel dimension. More formally, we define $\mathbf{v}^i = \{z_{mi+1}, ..., z_{m(i+1)}\}$ as a local contextual representation, where $i \in \{0, ..., G-1\}$ and G are the index and the number of groups respectively. $m = C/G$ is the number of channels per group. The normalization operator φ can be expressed as:

$$\hat{\mathbf{v}}^i = \varphi(\mathbf{v}^i) = \frac{1}{\sigma^i}(\mathbf{v}^i - \mu^i), \quad (2)$$

where μ^i and σ^i are the mean and standard deviation of the i-th group computed by:

$$\mu^i = \frac{1}{m}\sum_{n\in\mathcal{S}_i} z_n, \sigma^i = \sqrt{\frac{1}{m}\sum_{n\in\mathcal{S}_i}(z_n - \mu^i)^2 + \epsilon}. \quad (3)$$

Here ϵ is a small constant. $\mathcal{S}_i$ is the set of the i-th group of channel indexes.

The normalization operator plays two crucial roles in context feature transform. First, it enables each channel to adjust its own context feature by perceiving context information within each group, which implicitly captures dependencies between channels. Second, it can effectively eliminate the inconsistency of context feature distribution caused by different samples, which stabilizes the distribution of global context features.

Next, we define a transform operator to be a function ψ: $\mathbb{R}^C \rightarrow \mathbb{R}^C$ that maps the gathered context features $\hat{\mathbf{z}}$ to the importance scores $\mathbf{a}$, formulated as:

$$\mathbf{a} = \psi(\hat{\mathbf{z}}) = \mathbf{w} \cdot \hat{\mathbf{z}} + \mathbf{b}, \tag{4}$$

where $\hat{\mathbf{z}} = [\hat{\mathbf{v}}^1, \hat{\mathbf{v}}^2, ..., \hat{\mathbf{v}}^G]$. $\mathbf{w}$ and $\mathbf{b}$ are trainable gain and bias parameters of the same dimension as $\hat{\mathbf{z}}$. Note that the transform operator ψ is a per-channel linear transform, which means that information from other channels is not considered in the context transform process. In addition, it only introduces the parameters of $\mathbf{w}$ and $\mathbf{b}$, which are almost negligible compared to the entire network. Interestingly, the composition of two operators can be regarded as a special case of GN where the spatial height H and width W are 1. In the case of $G = 1$, it is equivalent to LN. But it is worth noting that the transform operator in LCT block is designed to transform the global context features, not to compensate for the potential lost of representational ability caused by normalization.

Feature fusion Finally, the feature fusion module modulates the input features by conditioning on the transformed contexts. Specifically, the output $\mathbf{Y} \in \mathbb{R}^{C \times H \times W}$ of LCT block is obtained by rescaling the original response $\mathbf{X}$ according to the attention activations $\sigma(\mathbf{a})$ and can be expressed as:

$$\mathbf{Y} = \mathbf{X} \cdot \sigma(\mathbf{a}). \tag{5}$$

Relationship to SE block LCT block shares the same context aggregation module and feature fusion module with SE block. The main difference between them is the context transform module, which reflects different perspectives of two blocks for global context modeling. First, SE block makes use of global information from other channels to help model the global context, which actually increases the complexity of context transform. Conversely, our LCT block simplifies global context modeling by independently transforming the global context of each channel. Such additional benefit is that LCT block is more lightweight than SE block. The number of parameters in SE block is $2C^2/r$, while the number of parameters in LCT block is C, where r is the reduction radio. Apparently, LCT block has significantly fewer parameters. Second, SE block explicitly captures channel-wise dependencies using two FC layers, while our approach implicitly captures dependencies with each group by normalization operator. The results in Table. 2 show that normalization operator can effectively capture dependencies with each group.

4. Experiments

In this section, we first evaluate the proposed LCT block on the task of image classification on ImageNet-1K [28]. Then, we conduct extensive ablation experiments on ImageNet-1K. Finally, we experiment in the COCO 2017 dataset [23] to demonstrate the general applicability of LCT block.

Backbone	Params	FLOPs	Top-1 (%)	Top-5 (%)
ResNet50	25.56M	4.122G	76.15	92.87
+SE	28.09M	4.130G	77.31	93.68
+LCT	25.59M	4.127G	**77.45**	**93.71**
ResNet101	44.55M	7.849G	77.37	93.56
+SE	49.33M	7.863G	78.49	94.19
+LCT	44.61M	7.858G	**78.55**	**94.26**

Table 1. Classification accuracies on the ImageNet validation set. Params denotes the number of parameters. FLOPs denotes the number of multiply-adds.

G	1	4	8	16	32	64	128
Top-1	77.37	77.36	77.44	77.34	77.32	**77.45**	-
Top-5	93.66	93.57	93.56	93.54	93.52	**93.71**	-

Table 2. Classification accuracies (%) of LCT-ResNet50 with different group numbers G on the ImageNet validation set. - denotes that the network can not converge.

Normalization	w/	w/o
Top-1 (%)	77.45	76.89
Top-5 (%)	93.71	93.33
Transform	w/	w/o
Top-1 (%)	77.45	76.82
Top-5 (%)	93.71	93.32

Table 3. Classification accuracies of LCT-ResNet50 with and without normalization/transform operator on the ImageNet validation set.

	LCT	SE	SE+
Top-1 (%)	**77.45**	77.31	77.37
Top-5 (%)	93.71	93.68	**93.73**

Table 4. Effect of inserting a normalization operator before the two FC layers of SE block. The backbone is ResNet50.

4.1. Image Classification on ImageNet

The ImageNet 2012 dataset contains 1.28 million training images and 50K validation images with 1000 classes.

Implementation details We train all models from scratch on 4 GPUs for 100 epochs, using synchronous SGD optimizer with a weight decay of 0.0001 and momentum 0.9. The initial learning rate is set to 0.1, and decreases by a factor of 0.1 every 30 epochs. The weight initialization is adopted in [12]. For ResNet50 backbone, the total batch size is set as 256. For ResNet101 backbone, we reduce the batch size to 220 due to the limited GPU memory. The stand data augmentation is performed 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. Input images are normalized using the channel means and standard deviations.

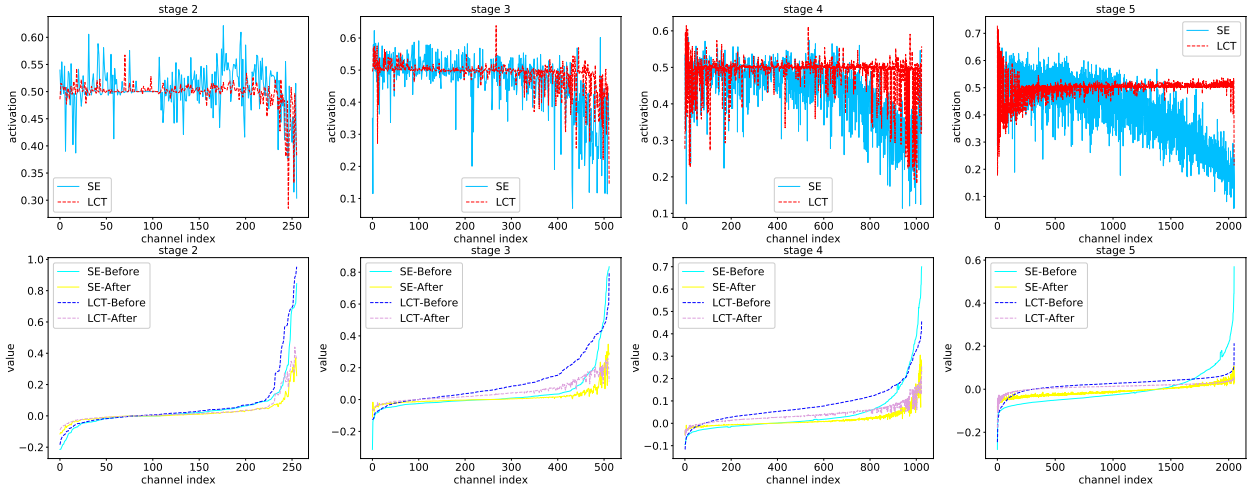

Figure 3. Visualizations of average global context features before and after rescaling and average attention activations of the first block at different stages on ImageNet validation set. The backbone is ResNet50. Top row: average attention activations. Bottom row: average global context features before and after rescaling.

Initialization			
w	**b**	Top-1 (%)	Top-5 (%)
0	0	77.36	93.60
0	1	**77.45**	**93.71**
1	0	77.24	93.54

Table 5. Ablation results of initialization with LCT-ResNet50 on the ImageNet validation set.

As is widely practiced in [17, 35], our LCT blocks are inserted into each residual block of ResNet. We use 0 and 1 to initialize all $\mathbf{w}$ and $\mathbf{b}$ parameters respectively. G is set as 64 by default. To make a fair comparison, the baseline models are reproduced in the same training settings. We report the top-1 and top-5 classification accuracies on the single 224×224 center crop in the validation set.

Classification results Table 1 presents the main results of our experiments. We observe that LCT block performs better than SE block with fewer parameters and less computation regardless of the depth of the backbone. Compared to ResNet, our LCT block adds few parameters and computations, but achieves significant performance gains ($> 1.0\% \uparrow$ on Top-1 accuracy) even in deeper ResNet101. Remarkably, LCT-ResNet50 is able to outperform ResNet101, which indicates that the improvements brought by LCT block exceed the benefits of increased network depth (51 layers). These results demonstrate that LCT block is effective for image classification.

Analysis and discussion To gain some insight into the channel attention mechanism, we investigate the relationship between global context features and attention distribution. Specifically, we first compute the average global context features before and after rescaling and the corresponding attention activations across 1000 classes on ImageNet validation set. Then we sort the average global context features in ascending order for better observation. Fig. 3 shows the results of the first block at different stages.

We observe that both SE block and LCT block learn a negative correlation that global context features with larger absolute values tend to be assigned smaller activations, which suggests that channels with these context features are generally less useful. This is reasonable to some extent, since a large amount of noise is more likely to exist in these channels. When the magnitude of the features of some channels is dramatically larger than that of other channels, subsequent filters will par more attention on these less useful channels, leading to incorrect semantic representation learning. By performing feature recalibration, both blocks effectively suppress these channels and reduce the contextual differences between channels, which enables subsequent filters to capture important semantics of each channel. In a sense, global contexts like an indicator that which channels need to be suppressed.

While both blocks learn a similar attention distribution, there are several differences. First, the attention distribution learned by LCT block is more stable because no other channel information is introduced in the transform operator. Second, LCT block does not over-suppress the original feature responses, thus retaining important semantic information. These findings provide explanations for the effectiveness of LCT block.

Detector	Backbone	ΔParams	ΔFLOPs	$AP^{bbox}_{0.5:0.95}$	$AP^{bbox}_{0.5}$	$AP^{bbox}_{0.75}$	AP^{bbox}_{small}	AP^{bbox}_{media}	AP^{bbox}_{large}
Faster R-CNN	baseline	-	-	38.5	60.5	41.8	22.3	43.2	49.8
	+SE	+4.78M	+0.191G	39.8(+1.3)	61.9	43.1	23.9	43.8	**51.5**
	+LCT	+0.06M	+0.187G	**40.0**(+**1.5**)	**62.8**	**43.4**	**24.8**	**44.4**	50.9
Mask R-CNN	baseline	-	-	39.4	61.0	43.3	23.1	43.7	51.3
	+SE	+4.78M	+0.191G	40.7(+1.3)	62.7	44.3	24.5	44.8	52.7
	+LCT	+0.06M	+0.187G	**40.9**(+**1.5**)	**63.1**	**44.6**	**25.0**	**45.1**	**52.9**
Cascade R-CNN	baseline	-	-	42.0	60.3	45.9	23.2	46.0	56.3
	+SE	+4.78M	+0.191G	43.4(+1.4)	62.2	47.4	24.7	47.4	57.0
	+LCT	+0.06M	+0.187G	**43.6**(+**1.6**)	**62.4**	**47.6**	**25.4**	**47.6**	57.3
Cascade Mask R-CNN	baseline	-	-	42.6	60.7	46.7	23.8	46.4	56.9
	+SE	+4.78M	+0.191G	43.7(+1.1)	61.8	47.5	24.3	47.5	58.6
	+LCT	+0.06M	+0.187G	**44.1**(+**1.5**)	**62.4**	**48.3**	**25.0**	**47.7**	**59.3**

Table 6. Comparisons based on ResNet101 backbone on the task of **object detection**. ΔParams denotes the change amount of parameters. ΔFLOPs denotes the change amount of computations. The numbers in brackets denote the improvements over the baseline backbone.

Detector	Backbone	$AP^{mask}_{0.5:0.95}$	$AP^{mask}_{0.5}$	$AP^{mask}_{0.75}$	AP^{mask}_{small}	AP^{mask}_{media}	AP^{mask}_{large}
Mask R-CNN	baseline	35.9	57.7	38.4	19.2	39.7	49.7
	+SE	36.9(+1.0)	59.4	39.2	20.0	**40.8**	50.3
	+LCT	**37.0**(+**1.1**)	**59.6**	**39.3**	**20.5**	**40.8**	**50.5**
Cascade Mask R-CNN	baseline	37.0	58.0	39.9	19.1	40.5	51.4
	+SE	37.7(+0.7)	59.0	40.5	19.4	41.1	52.4
	+LCT	**38.1**(+**1.1**)	**59.5**	**41.3**	**19.9**	**41.3**	**53.2**

Table 7. Comparisons based on ResNet101 backbone on the task of **instance segmentation**. The results show that LCT block outperforms SE block.

4.2. Ablation study

Number of groups In this experiment, we assess the effect of group number on the performance of LCT block. As shown in Table. 2, LCT block is not sensitive to the group number, which is reasonable because the mean and variance do not change significantly with the number of channels per group. We observe that when $G = 128$, the network has failed to converge since too many groups lead to incorrect statistics estimation. In the extreme case of $G = C$, the statistics can not even be calculated. When $G = 64$, the performance is slightly higher than that of other settings, indicating that introducing too much information from other channels is not helpful. By default, we set $G = 64$ for LCT block. Moreover, LCT block consistently outperforms SE block for all values G, which suggests that the normalization operator can capture interdependencies between channels well, even in the extreme case of $G = 1$.

Normalization operator To investigate the influence of normalization in LCT block, we conduct experiments by removing the normalization operator from LCT block. Table. 3 shows the results. It is clear that LCT block without normalization operator suffers considerable performance degradation. This comparison shows that global context can not be effectively transformed using transform operator alone. It also demonstrates that normalization operator can effectively eliminate the inconsistency of context feature distribution and captures dependencies between channels well.

We have seen that normalization operator can improve the performance of LCT block and would like to explore whether normalization operator can also help SE block yield further performance gains. For this purpose, we insert a normalization operator before the FC layers of SE block. We refer to this block as SE+. G is set to 64. The results are shown in Table. 4. We find that normalization operator does not bring significant gain to SE block. The top-1 accuracy of SE+ block is slightly inferior to ours. Based on these results, we can draw the following conclusions: 1) The two FC layers in SE block not only can transform the global context features, but also effectively prevent the inconsistency of feature distribution caused by different samples, which is surprisingly similar to two operators in LCT block. The difference is that LCT block decomposes the roles of two FC layers into two independent operators, each of which performs its own functions. 2) After normalization, a per-channel linear transform is sufficient to transform the global contexts. Introducing information from other channels complicates context feature transform. These findings provide an explanation for the effectiveness of LCT block.

Transform operator Next we study the effect of transform operator. To this end, we retain the normalization operator and remove the transform operator from LCT block. The results are shown in Table. 3. We observe that perfor-

mance is noticeably reduced and is slightly worse than that without normalization operator, suggesting that transform operator is vitally important for global context transform. The reason is that normalization operator can not learn the negative correlation between global context features and attention distribution. We also find that LCT block with two operators achieves the best performance, which indicates that two operators are complementary and indispensable for global context modeling.

Initialization Table 5 shows the ablation results of initialization. Different from the initialization in GN, IN and LN, we find that it is more appropriate to initialize $\mathbf{w}$ and $\mathbf{b}$ to 0 and 1 respectively, which is consistent with the finding in SGE [21]. Initializing $\mathbf{w}$ and $\mathbf{b}$ to 0 gets suboptimal results. As shown in Fig. 3, we observe that most of the attention values fluctuate around 0.5 for both SE block and LCT block. Hence a possible explanation is that initializing $\mathbf{w}$ to 0 makes $\sigma(0 \sim 1)$ around 0.5, which is conducive to the learning of attention distribution. When $\mathbf{w} = 1$ and $\mathbf{b} = 0$, LCT block achieves the worst results, because the transform operator is designed to transform the context features rather than compensate for the lost of representational ability caused by normalization.

4.3. Object Detection and Segmentation on COCO

Next we evaluate our block on object detection and instance segmentation on COCO 2017 [23]. We train using 118k train images and evaluate on 5k val images. The COCO-style average precisions at different boxes and the mask IoUs are reported.

Implementation details All experiments are implemented with $mmdetection$ framework [4]. The input images are resized such that the long edge and short edge are 1333 and 800 pixels respectively. We train on 4 GPUs with 1 images per each for 12 epochs. All models are trained using synchronized SGD with a weight decay of $1e$-4 and momentum of 0.9. According to the linear scaling rule [10], the initial learning rate is set to 0.005, which is decreased by 10 at the 9th and 12th epochs. The backbones of all models are pretrained on ImageNet. We finetune all layers except for c1 and c2 with FPN [24], detection and segmentation heads. During finetuning the BathNorm layers are frozen. Other hyper-parameters follow the default settings of the $mmdetection$ framework. The backbone is ResNet101 in all experiments.

Object detection We assess the ability of LCT block generalize to the task of object detection. To this end, we insert LCT blocks into four state-of-the-art detection frameworks, including Faster RCNN [27], Mask RCNN [11], Cascade RCNN [2] and Cascade Mask RCNN [4]. The results on val set are given in Table 6. We observe that our approach is better than SE block with fewer parameters and less computations regardless of the detectors, which indicates that modeling global context for each channel independently is also effective on the task of object detection. In addition, compared to baseline, LCT block consistently yields $1.5 \sim 1.6\%$ $\text{AP}^{bbox}_{0.5:0.95}$ points with almost no extra parameters and computations, suggesting that our approach is widely applicable across various detector architectures. We also find that LCT block greatly improves the detection performance of Faster RCNN, Mask RCNN and Cascade RCNN for small objects with the gain exceeding 1.9% AP^{mask}_{small}. For Cascade Mask RCNN, the detection performance of large objects is significantly boosted ($2.4\% \uparrow$ AP^{mask}_{large}).

Instance segmentation Finally, we would like to evaluate whether LCT block can be generalized to instance segmentation. We select two popular frameworks, Mask RCNN and Cascade Mask RCNN. As can been seen in Table 7, LCT block also outperforms SE block, which is consistent with the results in image classification and object detection. When adopting stronger detector Cascade Mask RCNN, the improvements achieved by LCT block are still significant, suggesting that our approach is complementary to the capacity of current model. Compared to baseline, LCT block can boost performance by 1.1 % $\text{AP}^{mask}_{0.5:0.95}$ regardless of the strength of the detectors. These results suggest the generalization and effectiveness of our approach.

5. Conclusion

In this paper, we presented an in-depth study of the relationship between global context and attention distribution. Then we considered the question of how to effectively learn the correlation between them. To this end, we introduced a simple yet effective channel attention architecture, LCT block, to explore this question and provided experimental evidence that demonstrates the effectiveness and generalization of our approach across multiple visual tasks. In further work, we plan to develop more efficient algorithms to exploit feature context, which may provide new insights into channel attention mechanism.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*, 2019.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[6] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ2-nets: Double attention networks. *Advances in Neural Information Processing Systems*, pages 352–361, 2018.

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[9] Jianlong Fu, Heliang Zheng, and Mei Tao. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 2017.

[10] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*, pages 1026–1034, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *computer vision and pattern recognition*, pages 770–778, 2016.

[14] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.

[15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[16] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 9401–9411, 2018.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *computer vision and pattern recognition*, pages 7132–7141, 2018.

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*, 2018.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *international conference on machine learning*, pages 448–456, 2015.

[20] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. *arXiv preprint arXiv:1805.10180*, 2018.

[21] Xiang Li, Xiaolin Hu, and Jian Yang. Spatial group-wise enhance: Enhancing semantic feature learning in convolutional networks. 2019.

[22] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 510–519, 2019.

[23] Tsungyi Lin, Michael Maire, Serge J Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *european conference on computer vision*, pages 740–755, 2014.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[25] Ningning Ma, Xiangyu Zhang, Haitao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *european conference on computer vision*, pages 122–138, 2018.

[26] Jongchan Park, Sanghyun Woo, Joonyoung Lee, and In So Kweon. Bam: Bottleneck attention module. *british machine vision conference*, page 147, 2018.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[29] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liangchieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *computer vision and pattern recognition*, pages 4510–4520, 2018.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *computer vision and pattern recognition*, pages 1–9, 2015.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[33] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. *computer vision and pattern recognition*, pages 6450–6458, 2017.

[34] Xiaolong Wang, Ross B Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *computer vision and pattern recognition*, pages 7794–7803, 2018.

[35] Sanghyun Woo, Jongchan Park, Joonyoung Lee, and In So Kweon. Cbam: Convolutional block attention module. *european conference on computer vision*, pages 3–19, 2018.

[36] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[37] Saining Xie, Ross B Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *computer vision and pattern recognition*, pages 5987–5995, 2017.

[38] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. *computer vision and pattern recognition*, pages 1857–1866, 2018.

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *british machine vision conference*, 2016.

[40] Hang Zhang, Kristin J Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Kumar Agrawal. Context encoding for semantic segmentation. *computer vision and pattern recognition*, pages 7151–7160, 2018.

[41] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 714–722, 2018.

[42] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.