Gaussian Context Transformer

Dongsheng Ruan[1,3,5], Daiyin Wang[2], Yuan Zheng [4], Nenggan Zheng[1, 3]*, Min Zheng [5]*
[1] Qiushi Academy for Advanced Studies, Zhejiang University
[2] College of Optical Science and Engineering, Zhejiang University
[3] College of Computer Science and Technology, Zhejiang University
[4] School of Aeronautics and Astronautics, Zhejiang University
[5] State Key Laboratory for Diagnosis and Treatment of Infectious Diseases, The First Affiliated Hospital,
College of Medicine, Zhejiang University
{dongshengruan, minzheng}@zju.edu.cn, wangdaiyin@126.com, ranchozy@gmail.com, zng@cs.zju.edu.cn

Abstract

Recently, a large number of channel attention blocks are proposed to boost the representational power of deep convolutional neural networks (CNNs). These approaches commonly learn the relationship between global contexts and attention activations by fully-connected layers or linear transformations. However, we empirically find that though many parameters are introduced, these attention blocks may not learn the relationship well. In this paper, we hypothesize that the relationship is predetermined. Based on this hypothesis, we propose a simple yet extremely efficient channel attention block, called Gaussian Context Transformer (GCT), which achieves contextual feature excitation using a Gaussian function that satisfies the presupposed relationship. According to whether the standard deviation of the Gaussian function is learnable, we develop two versions of GCT: GCT-B0 and GCT-B1. GCT-B0 is a parameter-free channel attention block by fixing the standard deviation. It directly maps global contexts to attention activations without learning. In contrast, GCT-B1 is a parameterized version, which adaptively learns the standard deviation to enhance the mapping ability. Extensive experiments on ImageNet and MS COCO benchmarks demonstrate that our GCTs lead to consistent improvements across various deep CNNs and detectors. Compared with a bank of state-of-the-art channel attention blocks, such as SE [17] and ECA [42], our GCTs are superior in effectiveness and efficiency.

1. Introduction

Deep convolutional neural networks (CNNs) have achieved significant progresses in many computer vision tasks, such as image classification [21, 13], segmentation [28, 30], and object detection [9, 3]. However, the local context characteristics of convolutional kernel prevent CNNs from effectively capturing global context information in an image, which is often essential for semantically understanding. To tackle this problem, attention mechanisms are commonly adopted. Their core is to arm CNNs with additional lightweight modules which can capture global long-range dependencies [43, 8, 43]. As one of them, channel attention mechanism has become increasingly popular, owing to its simplicity and effectiveness. The most pioneering work in this scenario is squeeze-and-excitation networks (SENet) [17], which aims to adaptively emphasize important channels and suppress trivial ones by capturing channel-wise dependencies, bringing enormous benefits for various CNNs.

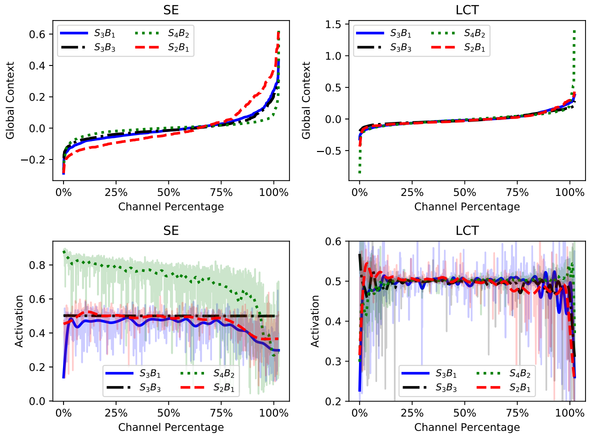

Figure 1. Visualization of the sorted global contexts and the according attention activations of different channel attention blocks at different stages across 1000 classes on ImageNet validation set. "S_iB_j" denotes the j-th attention block of stage i. In the second row, the semitransparent lines and the opaque lines represent the attention activations before and after a low-pass filter, respectively.

*Corresponding author

Methods	Pram-Free	#Param	Top-1	Top-5
ResNet50	-	-	76.15	92.87
+SE	×	$2C^2/r$	77.18	93.67
+LCT	×	$2C$	77.45	93.71
+ECA	×	$\|(log_2C+1)/2\|_{odd}$	77.48	93.68
+GCT-B0	✓	**0**	77.51	**93.86**
+GCT-B1	×	1	**77.55**	93.71

Table 1. Comparison of existing state-of-the-art channel attention blocks on ImageNet validation set. Param-Free denotes whether the attention block is free of parameters. #Param denotes the number of parameters introduced in one channel attention block. C denotes the number of channels. r denotes the reduction ratio of SE. $|\cdot|_{odd}$ indicates the nearest odd number of $\cdot$. Note that our GCTs outperform other channel attention blocks with fewer parameters introduced.

Several channel attention blocks have been thereafter proposed to improve the SE block with different perspectives including simplifying feature transform module [7, 42], changing fusion mode [4], and integrating with spatial attention mechanism [44, 31]. Despite the performance improvements, these approaches generally introduce large amounts of parameters to learn the relationship between global contexts and attention activations. However, the learnt relationship may not be good enough.

As observed in linear context transform (LCT) block [7], SE tends to learn a negative correlation that the more global contexts deviate from their mean, the smaller attention activations are attached, as shown in Fig. 1. To learn this correlation more accurately, LCT uses a per-channel transformation to replace two fully-connected layers of SE. However, we empirically find that this negative correlation may not be well-learnt by LCT. As shown in Fig. 1, the attention activations learnt by LCT fluctuate greatly.

To alleviate the above problem, we hypothesize a negative correlation between global contexts and attention activations. Based on this hypothesis, we propose a new channel attention block, called Gaussian Context Transformer (GCT), which directly maps global contexts to attention activations with a Gaussian function that represents the presupposed negative correlation. The basic structure of GCT is illustrated in Fig. 2. Specifically, after global average pooling, GCT performs normalization to stabilize the global contextual distribution. Then a Gaussian function is used to excite (transform and activate) the normalized global contexts to obtain the attention activations. When the Gaussian function is fixed, we refer to this model as GCT-B0. Note that GCT-B0 is a **parameter-free** attention block that model global contexts without contextual feature transform learning. As shown in Table 1, GCT-B0 yields significant performance gains over baselines and outperforms other state-of-the-art channel attention blocks without introducing any parameters, indicating that the contextual feature transform learning is not essential. Further, we develop a learnable GCT, called GCT-B1, which adaptively learns the standard deviation of Gaussian function. We empirically show that GCT-B1 generally performs better than GCT-B0 on ImageNet. On object detection/segmentation tasks, GCT-B0 and GCT-B1 achieve similar performance.

In summary, our main contributions can be summarized as follows:

- Our work provides a new insight into the channel attention mechanism: parameterized contextual feature transform learning is not essential. This will inform further research progress in designing more efficient channel attention blocks.
- We propose a simple yet extremely efficient channel attention block (GCT), which hypothesizes the relationship between global contexts and attention activations and excites global contexts only using a Gaussian function. Our GCTs can significantly boost various deep CNNs and detectors.
- Comprehensive experiments on ImageNet and MS COCO consistently demonstrate the superiority and generalization ability of our proposed GCTs. In particular, GCT-B0 generally outperforms other state-of-the-art channel attention blocks without introducing any parameters.

2. Related Work

Deep Networks. A wide range of work has shown that excellent network design can substantially improve network performance. AlexNet [21] and VGGNet [35] pioneer the use of convolutional networks, significantly outperforming non-convolutional learning approaches. Inception family [37, 20, 38, 36] proposes to model features in multi-scale formulation by combining convolution kernels of different sizes. Residual learning is first introduced in [13], to mitigate the exploding/vanishing gradient problem when the network is deep. Since then, a new network design paradigm has been opened up, and most of subsequent networks are built upon it [18, 45, 10, 47, 48, 46].

To meet requirements under the common mobile and embedded settings, studies on designing lightweight convolutional networks achieve great attention. SqueezeNet [19], MobileNets [15, 34, 14], and ShuffleNets [49, 29] are consequently proposed, showing good trade-offs between accuracy and number of operations, as well as actual latency and the number of parameters. Recently, neural architecture search (NAS) became a mainstream trend in designing efficient mobile-size networks and shows better efficiency than hand-crafted convolutional networks [39, 2, 40].

Different from these networks, our work is an add-on that aims to improve the representational power of deep convolutional networks by recalibrating the channel-wise features.

Attention blocks. A large number of attention blocks have been proposed to improve the performance of deep convolutional networks, which can be basically divided into spatial attention [41, 23, 43, 1, 6], channel attention [17, 24, 4, 7, 42], and a combination of both [44, 31]. Since GCT is a channel attention block, we briefly review the channel attention blocks proposed in recent years.

SE [17] and GE [16] recalibrate feature maps by capturing channel-wise dependencies, significantly boosting network performance. Further, CBAM [44] and BAM [31] integrate spatial and channel attentions to refine feature maps via rescaling. GC [4] combines NL [43] and SE to model global contexts, and achieves better performance with lightweight property. More recently, LCT [7] observes a negative correlation between global contexts and attention activations and proposes to use a per-channel linear transformation to transform global contexts, achieving comparable performance compared to SE. ECA [42] targets enhancing efficiency via constraining cross-channel interaction within a local range.

Instead of learning the relationship between global contexts and attention activation, we predetermine the relationship and use a Gaussian function to achieve contextual feature excitation. Experimental results show that our method can achieve better performance than most channel attention blocks across various tasks with fewer parameters.

3. Method

Inspired by the observation in LCT [7], we hypothesize that the channel attention mechanism learns a negative correlation that the more global contexts deviate from their mean, the smaller attention activations are attached. Based on this hypothesis, we propose the gaussian context transformer (GCT) to model global contexts. In this section, we present the architecture of GCT in detail. A diagram of GCT is illustrated in Fig. 2.

3.1. Gaussian Context Transformer

GCT consists of three operations: global context aggregation (GCA), normalization, and gaussian context excitation (GCE). The GCA operation aims to obtain channel-wise statistics via spatially aggregating global context information in feature maps, so as to help the network capture long-range dependencies. Following SE, we use the simplest aggregation form, global average pooling. Concretely, given a feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, the global contexts can be formulated as $\mathbf{z} = avg(\mathbf{X}) = \{z_k = \frac{1}{H\times W}\sum_{i=1}^{W}\sum_{j=1}^{H}\mathbf{X}_k(i,j) : k \in \{1,...,C\}\}$, where C is the number of channels and H, W are the spatial dimensions.

Previous works excite the obtained global contexts by sequentially performing two operations: transform and activation. First, the transform operation transforms global contexts using fully-connected layers or linear transformations. Then a $sigmoid$ function is used to activate the transformed global contexts to the attention activations. Different from these works, we propose a new context excitation way, which performs the transform and activation operations by a simple function $f(\cdot)$ that certainly represents the hypothetical negative relationship.

Specifically, we define the mean shift as $\mathbf{z} - \mu$, where $\mu = \frac{1}{C}\sum_{k=1}^{C} z_k$ stands for the mean of the global contexts $\mathbf{z}$. The mean shift measures the deviation between $\mathbf{z}$ and μ. However, directly setting the mean shift as input will make $f(\cdot)$ unstable because of the inconsistent mean shift distributions caused by different input samples. To alleviate this problem, we introduce an instance-specific factor σ to stabilize it in distribution with a mean of 0 and a variance of 1, which can be expressed as:

$$\boxed{\hat{\mathbf{z}} = \frac{1}{\sigma}(\mathbf{z} - \mu),} \tag{1}$$

where σ denotes the standard deviation of global contexts computed by $\sqrt{\frac{1}{C}\sum_{k=1}^{C}(z_k - \mu)^2 + \epsilon}$, with ϵ as a small constant. We observe that this formulation of $\hat{\mathbf{z}}$ turns to be same as the normalization of the global contexts $\mathbf{z}$. For simplification, we denote this formulation as the normalization operation on $\mathbf{z}$: $\hat{\mathbf{z}} = norm(\mathbf{z})$.

To satisfy the hypothesis, we seek a continuous function $f(\hat{\mathbf{z}})$ that meets the following conditions: 1. $f(\hat{\mathbf{z}})$ is in the semi-closed interval $(0, 1]$, i.e., $f(\hat{\mathbf{z}}) \in (0, 1]$; 2. $f(\hat{\mathbf{z}})$ reaches unique maxima 1 when $\hat{\mathbf{z}}$ equals to zero; 3. $f(\hat{\mathbf{z}})$ monotonically increases when $\hat{\mathbf{z}}$ is smaller than zero and monotonically decreases when $\hat{\mathbf{z}}$ is larger than zero; 4. $f(\hat{\mathbf{z}})$ asymptotically approaches zero when $\hat{\mathbf{z}}$ approximates infinity, i.e., $\lim_{\hat{\mathbf{z}}\to\pm\infty} f(\hat{\mathbf{z}}) = 0$. Among the well-known functions, Gaussian function fits these properties well, thus selected in our paper. Formally, we define the GCA operation to be a Gaussian function G:

$$G(\hat{\mathbf{z}}) = ae^{-\frac{(\hat{\mathbf{z}}-b)^2}{2c^2}}, \tag{2}$$

where a denotes the amplitude of the Gaussian function and is set to 1 to satisfy the first condition. b denotes the mean of the Gaussian function and is set as 0 to meet the second condition. c denotes the standard deviation of the Gaussian function that controls the difference in channel attention activations: the larger the standard deviation is, the smaller the difference in attention activations between channels is. Hence G can be simplified as:

$$\mathbf{g} = G(\hat{\mathbf{z}}) = e^{-\frac{\hat{\mathbf{z}}^2}{2c^2}}, \tag{3}$$

where c can be a constant or a learnable parameter. $\mathbf{g}$ represents the attention activations.

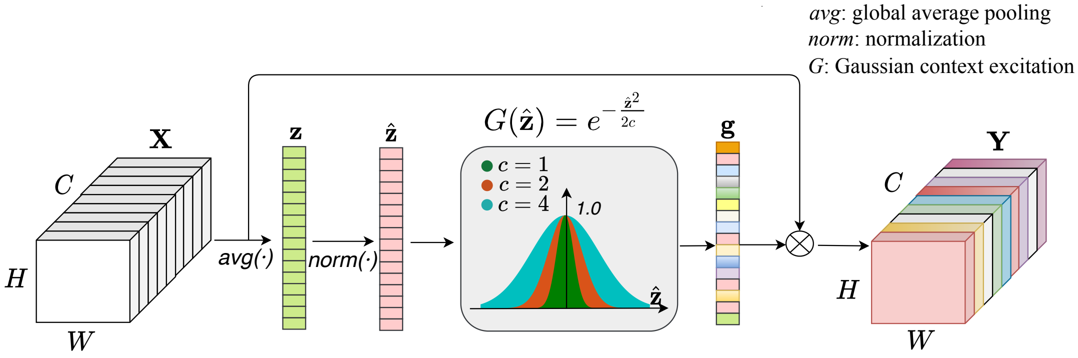
avg: global average pooling
norm: normalization
G: Gaussian context excitation
X
C
H
W
avg(·)
z
norm(·)
ẑ
$G(\hat{\mathbf{z}}) = e^{-\frac{\hat{\mathbf{z}}^2}{2c}}$
$c = 1$
$c = 2$
$c = 4$
1.0
ẑ
g
Y
C
H
W

Figure 2. Diagram of our gaussian context transformer (GCT). c denotes the standard deviation of Gaussian function, which is either a constant or a learnable parameter. When c is a constant, GCT is a parameter-free channel attention block. $\otimes$ denotes broadcast element-wise product.

We combine the above operations to form our proposed gaussian context transformer (GCT), which can be formulated as:

$$\boxed{\mathbf{Y} = e^{-\frac{norm(avg(\mathbf{X}))^2}{2c^2}}\mathbf{X}.} \tag{4}$$

3.2. Parameter-free GCT

When the standard deviation c is a constant, GCT is a **parameter-free** channel attention block. We refer to this module as GCT-B0. In Sec. 4.1.1, we vary the standard deviation of GCT-B0 on ImageNet. We empirically find that the setting of $c = 2$ achieves the best Top-1 accuracy. Except where noted otherwise, we set $c = 2$ in our experiments. Furthermore, we also find that GCT-B0 generally achieves better performance than existing channel attention blocks, such as SE, LCT, and ECA, without introducing any parameters.

3.3. Parameterized GCT

We also develop a parameterized GCT, in which c is a learnable standard deviation. We refer to this module as GCT-B1. To constrain $c \in [\beta, \alpha + \beta]$, we set its upper bound and lower bound in the following way:

$$c = \alpha \cdot sigmoid(\theta) + \beta, \tag{5}$$

where α and β are constants. θ is a learnable parameter, initialized as 0. Note that GCT-B1 has only **one** parameter θ, and adaptively learns the most appropriate Gaussian function from the dataset. As shown in Table 3, GCT-B1 with $c \in [1, 4]$ achieves better performances. By default, we set $\alpha = 3$ and $\beta = 1$ in our experiments. The experimental results in Sec. 4 demonstrate that GCT-B1 performs better than GCT-B0 on ImageNet. On MS COCO dataset GCT-B0 and GCT-B1 achieve similar performance.

3.4. Comparisons to Other Channel Attention Blocks

Our approach markedly differs from other channel attention blocks in the following ways. First, instead of learning the aforementioned negative correlation, GCT uses a Gaussian function to predetermine such a correlation without learning process. Therefore, as shown in Fig. 3, the attention activations of GCT show better stability. Second, GCT transforms and activates global contexts with a Gaussian function, breaking the traditional excitation paradigm followed by other attention blocks. In the end, our GCT is more efficient than other channel attention blocks. In particular, GCT-B0 outperforms SE, LCT, and ECA without introducing any parameters.

4. Experiments

In this section, we evaluate the proposed GCT on two basic tasks, image classification on ImageNet [22] and object detection/segmentation on COCO [25]. Experimental results show that GCT generally outperforms other state-of-the-art channel attention blocks.

4.1. Image Classification on ImageNet

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

Setup. Our experiments are implemented with PyTorch framework [32]. Unless otherwise noted, the standard deviation c in GCT-B0 is set to 2. The range of the standard deviation c in GCT-B1 is $1 \sim 4$ where c is initialized to 2.5.

Implementation details. We train all models from scratch on 8 GPUs with 32 images per GPU (total batch size of 256) 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

c	1	2	3	4	5	6
Top-1	77.15	**77.51**	77.24	77.27	77.11	77.06
Top-5	93.59	**93.86**	93.71	93.73	93.61	93.47

Table 2. Classification accuracy (%) of GCT-ResNet50-B0 with different standard deviation c on the ImageNet validation set.

c	$1 \sim 3$	$1 \sim 4$	$2 \sim 3$	$2 \sim 4$
Top-1	77.52	**77.55**	77.48	77.39
Top-5	93.77	93.71	93.72	**93.81**

Table 3. Classification accuracy (%) of GCT-ResNet50-B1 with different ranges of c on the ImageNet validation set.

in [12]. For training ShuffleNetV2, we follow the settings in [29], where networks are trained within 300 epochs using SGD with weight decay of $4e^{-5}$, momentum of 0.9, label smoothing of 0.1 and mini-batch size of 1024 (128 images per GPU). The initial learning rate is set to 0.5, and is decreased by a linear decay strategy. We perform 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. The Top-1 and Top-5 classification accuracy are reported on the single 224×224 center crop in the validation set.

4.1.1 Ablation Study

We report the ablation studies based on ResNet50 backbone. GCT is placed after the last BatchNorm [20] layer inside each bottleneck of ResNet.

Standard deviation of GCT-B0. We investigate the effect of fixed standard deviation c in GCT-B0 on classification accuracy. The results are shown in Table 2. We observe that with the increase of c, the network performance presents a trend of first increasing and then decreasing. It is reasonable because a large variance reduces the difference in attention activations between channels, preventing the deviant global contexts from being well suppressed. On contrary a small variance excessively suppresses the global deviant contexts. When $c = 2$, GCT-ResNet50-B0 achieves the best Top-1 and Top-5 accuracy. Hence we set $c = 2$ for GCT-B0 by default.

Standard deviation of GCT-B1. Compared to GCT-B0, the standard deviation c of GCT-B1 is a learnable parameter whose range is determined by α and β. We constrain c by setting different α and β. For example, when $\alpha = 3$ and $\beta = 1$, the range of c is $1 \sim 4$. As observed in Table 2, the reasonable standard deviation should be between 1 and 4, so we explore its performance in this range. Table 3 shows the experimental results. It can be seen that the Top-1 accuracy of range $1 \sim 4$ is similar to that of range $1 \sim 3$. To make GCT more exploratory, we set the range of c as $1 \sim 4$ by default.

4.1.2 Comparisons with state-of-the-art Channel Attention Blocks.

To evaluate our approach, we compare GCT with a series of state-of-the-art channel attention blocks including SE [17], LCT [7], and ECA [42]. Here we select four popular networks as the backbones, including BN-Inception [20], ResNet [13], ResNext [45] and ShuffleNetV2 [29]. Table 4 presents the main results of our experiments.

BN-Inception. We first evaluate GCT on the non-residual network BN-Inception [20]. All attention blocks are placed after the Inception module. We can see that GCT-B1 outperforms other channel attention blocks with fewer parameters. GCT-B0 is superior to ECA, but slightly inferior to SE and LCT, probably because the setting of $c = 2$ is not suitable for BN-Inception. Furthermore, our GCT-B0 improves the original BN-Inception by 0.5% in terms of Top-1 accuracy without parameter increase. The results show that our GCT can be used to improve the performance of the non-residual network.

ResNet/ResNeXt. We further verify the effectiveness of our approach on two popular residual networks ResNet [13] and ResNeXt [45]. All attention blocks are placed after the last BatchNorm [20] layer inside each bottleneck of ResNet/ResNext. We make the following four observations. First, GCT-B0 and GCT-B1 consistently bring significant performance improvements across different depths. In particular, GCT-ResNet-B0 achieves a $\sim 1.3\%$ gain over ResNet without introducing any parameters. Second, except for ECA-ResNet101, the results of GCT-B0 are better than those of other approaches. Third, GCT-B1 outperforms SE, LCT, and ECA across different depths and backbones with fewer parameters and similar computational costs. Last but not the least, GCT-B1 performs better than GCT-B0 at the cost of one parameter per introduced block, which suggests that adaptive learning of Gaussian function is effective.

ShuffleNetV2. Finally, we investigate the performance of our approach on a lightweight model. To this end, we employ ShuffleNetV2 as backbone and integrate all attention blocks before channel shuffle. The results in Table 4 show that both GCT-B0 and GCT-B1 are better than other channel attention blocks. In particular, GCT-B0 outperforms SE by 1.0% Top-1 accuracy. Compared to vanilla ShuffleNetV2, our GCTs achieve a 1.4% gain in Top-1 accuracy. These results indicate that GCT can be successfully applied to lightweight model.

All the above results fully demonstrate the effectiveness and generality of our proposed GCT.

Methods	#Param	GFLOP	Top-1	Top-5
BN-Inception	**11.30M**	**2.050**	74.48	91.96
+SE	+0.64M	2.052	75.11	92.24
+LCT	+13.25K	2.051	75.12	92.32
+ECA	+0.05K	2.051	74.76	92.06
+GCT-B0	**+0K**	2.051	74.98	92.20
+GCT-B1	+0.01K	2.051	**75.35**	**92.40**
ResNet18	**11.69M**	**1.822**	69.76	89.08
+SE	+0.09M	1.823	70.59	89.78
+LCT	+3.84K	1.823	70.85	89.75
+ECA	+0.03K	1.823	70.50	89.56
+GCT-B0	**+0K**	1.823	70.90	90.03
+GCT-B1	+0.01K	1.823	**71.21**	**90.04**
ResNet50	**25.56M**	**4.122**	76.15	92.87
+SE	+2.54M	4.130	77.18	93.67
+LCT*	+30.21K	4.127	77.45	93.71
+ECA*	+0.09K	4.127	77.48	93.68
+GCT-B0	**+0K**	4.127	77.51	**93.86**
+GCT-B1	+0.01K	4.127	**77.55**	93.71
ResNet101	**44.55M**	**7.849**	77.37	93.56
+SE	+4.78M	7.863	78.47	94.10
+LCT*	+65.02K	7.858	78.55	94.26
+ECA*	+0.17K	7.858	78.65	94.34
+GCT-B0	**+0K**	7.858	78.60	94.23
+GCT-B1	+0.03K	7.858	**78.85**	**94.41**
ResNeXt50	**25.03M**	**4.273**	77.62	93.70
+SE	+2.54M	4.281	78.12	93.90
+LCT	+30.21K	4.279	78.11	93.93
+ECA	+0.09K	4.279	77.71	93.88
+GCT-B0	**+0K**	4.279	78.47	**94.24**
+GCT-B1	+0.01K	4.279	**78.66**	94.17
ShuffleNetV2	**2.28M**	**0.150**	69.36	88.32
+SE	+0.17M	0.151	69.79	89.05
+LCT	+8.36K	0.151	70.59	89.51
+ECA	+0.08K	0.151	70.36	89.37
+GCT-B0	**+0K**	0.151	**70.79**	**89.80**
+GCT-B1	+0.01K	0.151	70.74	89.66

Table 4. Comparisons with the state-of-the-art channel attention blocks on ImageNet validation set. #Param denotes the number of parameters of the channel attention block. GFLOPs denote the computations. The best results are marked as bold. $*$ indicates that the results are from the original paper.

4.2. Object Detection/Segmentation on COCO

We evaluate our GCT on object detection and instance segmentation on COCO 2017, which has 80 object categories. Its training set is with 115k images, and validation set with 5k images. 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, both box AP (AP^{bbox}) and mask AP (AP^{mask}) are evaluated.

Setup. To validate the effectiveness and generality of our approach, we carry out experiments with different combinations of popular backbone ResNet, and state-of-the-art detection architectures including Faster RCNN [33], Mask RCNN [11], and RetinaNet [27]. By default, GCT is integrated into stages c3-c5 of ResNet. Similar to the setting in Sec. 4.1, the standard deviation c in GCT-B0 is 2. The range of the standard deviation c in GCT-B1 is $1 \sim 4$ where θ is initialized to 0.

Implementation details. All experiments are implemented with mmdetection v2.6 [5]. The short edge of the input image is resized to 800, and the long edge is limited to 1333. All models are trained on 8 GPUs with two images per each (mini-batch size of 16). The training is optimized by synchronized SGD with a weight decay of $1e^{-4}$ and momentum of 0.9. And the total training epoch is 12. The initial learning rate is set as 0.02, decreased by a factor of 10 at the 9th and 12th epochs. The backbone networks of all models are pre-trained on ImageNet. **All attention blocks are trained from scratch in the same training setting.** Following the conventional finetuning setting [11], we use frozen BatchNorm instead of synchronized BatchNorm. All layers except for c1 and c2 are jointly finetuned with FPN [26], detection and segmentation heads. Other hyperparameters follow the default settings in the mmdetection framework.

4.2.1 Object Detection

Table 5 shows the performances of four channel attention blocks (SE, LCT, ECA, and GCT) based on three detectors with ResNet as backbone on object detection task. Compared to vanilla ResNet50/ResNet101, our GCTs can yield a significant gain of $1.0 \sim 1.6\%$ AP^{bbox} for various detectors at the cost of no more than 0.03K parameters. Note that GCT can also be successfully used in one-stage detector RetinaNet, suggesting our approach's excellent generality. What's more, it can be seen that our GCTs consistently outperform SE, ECA, and LCT across different backbones and detectors with fewer parameters and similar computational burdens. In particular, GCT-B0 achieves state-of-the-art performance without introducing any parameters, suggesting our module's high efficiency in modeling global contexts. We also observe no significant performance difference between GCT-B0 and GCT-B1 on object detection, which is inconsistent with the results on ImageNet. This may be because the object detection dataset is relatively small.

4.2.2 Instance Segmentation

We further evaluate GCT by comparing it with SE, LCT, and ECA on instance segmentation task. We select Mask

Detector	Methods	#Param	GFLOPs	AP$^{bbox}_{0.5:0.95}$	AP$^{bbox}_{0.5}$	AP$^{bbox}_{0.75}$	AP$_S$	AP$_M$	AP$_L$
Faster R-CNN	ResNet50	**41.53M**	**207.07**	37.4	58.1	40.4	21.2	41.0	48.1
	+SE	+2.54M	207.19	38.1	59.5	41.1	22.4	41.8	49.1
	+LCT	+30.21K	207.19	38.3	59.5	41.8	22.7	42.2	49.0
	+ECA	+0.09K	207.19	38.7	60.3	41.9	22.4	42.7	49.2
	+GCT-B0	**+0K**	207.19	38.8	60.4	42.0	23.3	42.8	49.7
	+GCT-B1	+0.01K	207.19	**38.9**	60.4	42.3	22.8	43.1	49.7
	ResNet101	**60.52M**	**283.14**	39.4	60.1	43.1	22.4	43.7	51.1
	+SE	+4.78M	283.33	39.7	60.7	43.2	23.1	43.7	52.0
	+LCT	+65.02K	283.32	40.3	61.6	43.8	23.8	44.4	52.2
	+ECA	+0.17K	283.32	40.5	62.0	44.1	23.7	44.7	52.8
	+GCT-B0	**+0K**	283.32	**40.7**	62.1	44.6	23.7	45.1	52.6
	+GCT-B1	+0.03K	283.32	**40.7**	61.9	44.6	23.3	45.3	52.8
Mask R-CNN	ResNet50	**44.18M**	**275.58**	38.2	58.8	41.4	21.9	40.9	49.5
	+SE	+2.54M	275.69	38.8	60.0	42.3	22.7	42.6	50.0
	+LCT	+30.21K	275.69	39.0	60.0	42.7	23.1	42.8	50.5
	+ECA	+0.09K	275.69	39.2	60.6	42.4	23.1	43.1	50.6
	+GCT-B0	**+0K**	275.69	39.3	60.7	42.8	23.4	43.1	50.7
	+GCT-B1	+0.01K	275.69	**39.4**	60.8	42.9	23.6	43.3	50.7
	ResNet101	**63.17M**	**351.65**	40.4	60.5	44.0	22.6	44.0	52.6
	+SE	+4.78M	351.84	40.5	61.2	44.1	23.6	44.5	52.7
	+LCT	+65.02K	351.83	41.1	62.0	45.1	24.5	45.3	53.8
	+ECA	+0.17K	351.83	41.2	62.4	44.8	23.9	45.4	54.2
	+GCT-B0	**+0K**	351.83	41.4	62.5	45.1	23.7	45.6	53.9
	+GCT-B1	+0.03K	351.83	**41.5**	62.6	45.3	24.1	45.6	53.9
RetinaNet	ResNet50	**37.74M**	**239.32**	36.5	55.4	39.1	20.4	40.3	48.1
	+SE	+2.54M	239.43	36.9	56.2	39.5	21.6	40.6	48.3
	+LCT	+30.21K	239.43	36.7	55.8	39.2	20.7	40.5	47.6
	+ECA	+0.09K	239.43	37.5	56.9	40.1	21.0	41.2	49.1
	+GCT-B0	**+0K**	239.43	**38.1**	57.6	40.5	22.2	41.6	50.1
	+GCT-B1	+0.01K	239.43	37.8	57.1	40.4	21.8	41.7	49.7
	ResNet101	**56.74M**	**315.39**	38.4	57.6	41.0	21.7	42.8	50.4
	+SE	+4.78M	315.58	38.7	57.8	41.5	22.0	42.7	51.4
	+LCT	+65.02K	315.57	39.1	58.3	42.2	22.3	43.3	51.5
	+ECA	+0.17K	315.57	39.2	58.5	42.0	21.9	43.0	52.0
	+GCT-B0	**+0K**	315.57	**39.6**	59.1	42.6	22.5	43.8	51.8
	+GCT-B1	+0.03K	315.57	**39.6**	59.0	42.4	22.6	43.9	52.0

Table 5. Comparisons with other state-of-the-art channel attention blocks based on Faster R-CNN, Mask R-CNN, and RetinaNet with ResNet backbone on the task of object detection. #Param denotes the number of parameters. GFLOPs denote the computations. The best results are marked as bold.

R-CNN to demonstrate the superiority of our approach. The results are shown in Table 6. We can clearly see that both GCT-B0 and GCT-B1 also improve the segmentation performance of two backbone networks. Note that the performance improvement of GCT-B0 comes with zero parameter overhead, indicating that our approach is extremely efficient. Furthermore, our GCTs also outperform most existing state-of-the-art channel attention blocks with less parameters and computational overhead. These results verify the effectiveness and good generalization ability of our GCTs for various tasks.

4.3. Visualization Analysis

To better understand the channel attention mechanism, we visualize the global contexts and the attention distribution of existing channel attention blocks, including SE, LCT, ECA, and GCT. Specifically, we first compute the average global contexts before the attention blocks, i.e., $\bar{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} avg(\mathbf{X})$, across 1000 classes on ImageNet, where n is the number of images in the validation set. For better visualization, we define the absolute mean shift η of the average global contexts $\bar{\mathbf{z}}$ as follows:

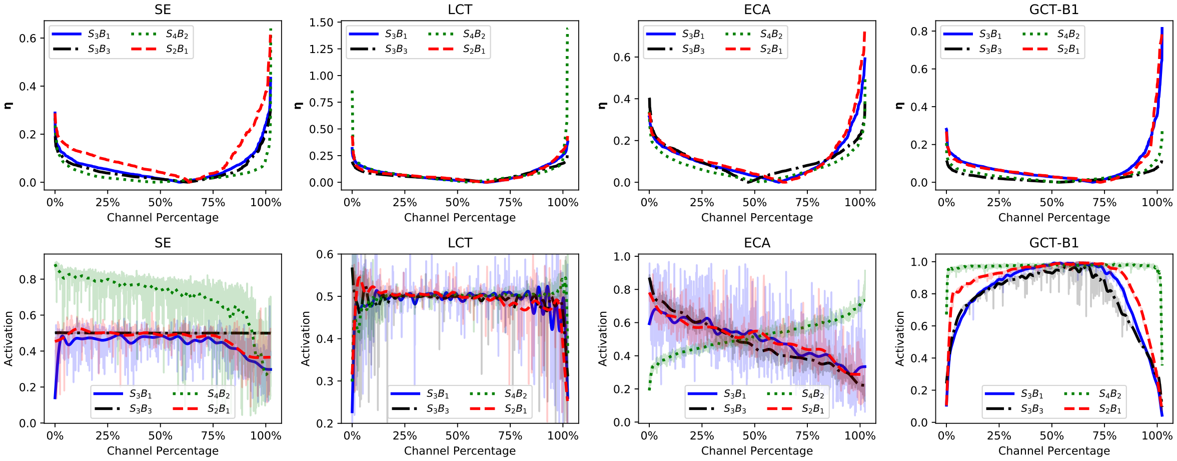

Figure 3. Visualization of the absolute mean shift $\boldsymbol{\eta}$ and the attention activations of different channel attention blocks at different stages across 1000 classes on ImageNet validation set. "S_iB_j" denotes the j-th attention block of stage i. In the second row, the semitransparent lines and the opaque lines represent the attention activations before and after a low-pass filter, respectively. *Best viewed in color.*

Methods	$\text{AP}^{mask}_{0.5:0.95}$	$\text{AP}^{mask}_{0.5}$	$\text{AP}^{mask}_{0.75}$	AP_S	AP_M	AP_L
ResNet50	34.7	55.7	37.2	18.3	37.4	47.2
+SE	35.2	56.8	37.4	18.9	38.5	47.9
+LCT	35.3	56.8	37.6	18.6	39.0	47.5
+ECA	35.7	57.5	38.1	19.5	39.3	48.5
+GCT-B0	**35.8**	57.5	38.1	19.8	39.4	48.2
+GCT-B1	35.7	57.6	38.0	19.7	39.5	48.2
ResNet101	36.1	57.5	38.6	18.8	39.7	49.5
+SE	36.5	58.1	39.0	19.5	40.0	49.5
+LCT	36.9	58.7	39.7	20.4	40.8	50.3
+ECA	37.1	59.2	39.7	19.9	41.1	50.8
+GCT-B0	37.2	59.5	39.6	19.7	41.0	50.8
+GCT-B1	**37.3**	59.5	39.7	19.9	41.1	50.9

Table 6. Comparisons with other state-of-the-art channel attention blocks based on Mask R-CNN with ResNet backbone on the task of instance segmentation. The best results are marked as bold.

$\boldsymbol{\eta} = |sort(\bar{\mathbf{z}} - mean(\bar{\mathbf{z}}))|$, where $mean(\cdot)$ denotes calculating the mean. $sort(\cdot)$ denotes sort in ascending order. Finally, we plot $\boldsymbol{\eta}$ and the corresponding attention activations. To better demonstrate the correlation, we also plot the attention activations after a low-pass filter. Fig. 3 shows the visualization results of different attention blocks at different stages. Since the number of channels of the blocks at different stages is different, the x-axis is represented by the channel percentage.

From Fig. 3, we make the following observations. First, for SE, some blocks, such as S_3B_1 and S_2B_1, learn a possible negative correlation that the more the global contexts deviate from their mean, the smaller the attention activations are attached, while others learn only partial correlation. For example, $SE\_S_4B_2$ tends to decrease monotonically, i.e., the larger the global context, the smaller the attention activation. Second, although LCT learns the negative relationship, the attention activations fluctuate greatly. The reason may be the limited learning ability of linear transformation. Third, we observe ECA only learns the trend of monotonic increase or decrease, perhaps because ECA uses a shared linear transformation to transform the global contexts across all channels. Finally, since GCT assumes the negative relationship between the absolute mean shift and the attention activations, GCT doesn't learn this relationship in the training process. It makes GCT transform global contexts more stably than other channel attention blocks, which provides a possible explanation for the efficiency of GCT.

5. Conclusion

In this paper, we propose a new channel attention block, Gaussian Context Transformer (GCT), which only uses a Gaussian function to model global contexts. Our method improves network performance with almost no extra parameters and calculations. In particular, our GCT-B0, a parameter-free attention block, shows that parameterized contextual feature transform learning is not necessary. Comprehensive experiments on ImageNet and MS COCO benchmarks are conducted to evaluate our approach. The experimental results show that our GCTs can provide a solid improvement over baselines across various backbones and detectors, outperforming other channel attention blocks. In further work, we plan to explore more efficient functions to perform context feature transform.

6. Acknowledgement

This work is supported by Zhejiang Provincial Natural Science Foundation (LR19F020005), National Natural Science Foundation of China (61572433, 61972347), and 13-5 State S&T Projects of China (2018ZX1030206).

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019. 3

[2] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 2

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1

[4] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV*, 2019. 2, 3

[5] 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:1906.07155*, 2019. 6

[6] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A$\hat{2}$-nets: Double attention networks. In *NeurIPS*, 2018. 3

[7] Ruan Dongsheng, Wen Jun, and Zheng Nenggan. Linear context transform block. In *AAAI*, 2020. 2, 3, 5

[8] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *CVPR*, 2019. 1

[9] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 1

[10] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *CVPR*, 2017. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017. 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 2015. 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 5

[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. In *CVPR*, 2019. 2

[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:1704.04861*, 2017. 2

[16] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*, 2018. 3

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CVPR*, pages 7132–7141, 2018. 1, 3, 5

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[19] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡ 0.5 mb model size. *arXiv:1602.07360*, 2016. 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2, 5

[21] Alex Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1, 2

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 4

[23] Xiang Li, Xiaolin Hu, and Jian Yang. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv:1905.09646*, 2019. 3

[24] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, 2019. 3

[25] 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. In *ECCV*, 2014. 4

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, 2017. 6

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[29] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 2, 5

[30] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1

[31] Jongchan Park, Sanghyun Woo, Joonyoung Lee, and In So Kweon. Bam: Bottleneck attention module. In *BMCV*, 2018. 2, 3

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 6

[34] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[36] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of

residual connections on learning. *arXiv:1602.07261*, 2016. 2

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[39] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 2

[40] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICLR*, 2019. 2

[41] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, 2017. 3

[42] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *CVPR*, 2020. 1, 2, 3, 5

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 3

[44] Sanghyun Woo, Jongchan Park, Joonyoung Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2, 3

[45] Saining Xie, Ross B Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2, 5

[46] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 2

[47] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv:1605.07146*, 2016. 2

[48] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, 2017. 2

[49] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 2