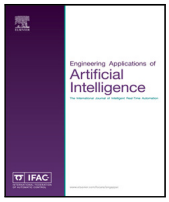
Engineering Applications of Artificial Intelligence 126 (2023) 107079

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



MCA: Multidimensional collaborative attention in deep convolutional neural networks for image recognition

Yang Yu, Yi Zhang, Zeyu Cheng, Zhe Song, Chengkai Tang *

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, China

ARTICLE INFO

Keywords:
Multidimensional collaborative attention
Attention mechanisms
Cross-dimension feature responses
Local feature interactions

ABSTRACT

A broad range of prior research has demonstrated that attention mechanisms offer great potential in advancing the performance of deep convolutional neural networks (CNNs). However, most existing approaches either ignore modeling attention in both channel and spatial dimensions or introduce higher model complexity and heavier computational burden. To alleviate this dilemma, in this paper, we propose a lightweight and efficient multidimensional collaborative attention, MCA, a novel method for simultaneously inferring attention in channel, height, and width dimensions with almost free additional overhead by using a three-branch architecture. For the essential components of MCA, we not only develop an adaptive combination mechanism for merging dual cross-dimension feature responses in *squeeze transformation*, enhancing the informativeness and discriminability of feature descriptors but also design a gating mechanism in *excitation transformation* that adaptively determines the coverage of interaction to capture local feature interactions, overcoming the paradox of performance and computational overhead trade-off. Our MCA is simple yet general and can be easily plugged into various classic CNNs as a plug-and-play module and trained along with the vanilla networks in an end-to-end manner. Extensive experimental results for image recognition on CIFAR and ImageNet-1K datasets demonstrate the superiority of our method over other state-of-the-art (SOTA) counterparts. In addition, we also provide insight into the practical benefits of MCA by visually inspecting the GradCAM++ visualization results. The code is available at https://github.com/ndsclark/MCANet.

1. Introduction

Convolutional Neural Networks (CNNs) have been universally used in the computer vision community based on their mighty representation power, and their continuous evolution has constantly pushed the boundaries of various complex visual cognition tasks, including image recognition (Zhang et al., 2023), object detection (Wang et al., 2023), text detection (Ranjbarzadeh et al., 2022), medical image segmentation (Ranjbarzadeh et al., 2023; Tataei Sarshar et al., 2021; Ranjbarzadeh et al., 2021; Anari et al., 2022), and semantic segmentation (Shojaiee and Baleghi, 2023). In recent years, to continuously enhance the performance of CNNs, researchers have been concentrating on three fundamental factors of networks: depth, width, and cardinality, thus developing many sophisticated models (He et al., 2016a; Zagoruyko and Komodakis, 2016; Gao et al., 2019a). However, these models come with some limitations while improving performance: (1) designing these models requires excellent expertise and enormous effort; (2) the substantial increase in model complexity may bring difficulties in training and optimization; (3) introducing heavy storage and computational overhead can severely restrict their deployment on resource-constrained devices. To this end, rather than designing the complex architecture, recent works mainly focus on investigating another essential factor, the attention mechanism, which helps divert the network's attention to the most meaningful regions in the image while ignoring unnecessary parts. Attention mechanisms originate from the human visual system, allowing us to effectively and efficiently analyze and understand complex real-world scenes. Motivated by this intuition, researchers introduce attention mechanisms into visual recognition systems to boost their performance by mimicking this characteristic of the human visual system. In visual recognition, such attention mechanisms can be viewed as a dynamic weight adjustment process based on the importance of input image features and are used to tell networks to pay attention to "what" and "where" (Guo et al., 2022; de Santana Correia and Colombini, 2022). Therefore, how to design a simple but powerful attention mechanism is the key to further improving the performance of CNNs.

* Corresponding author.
E-mail addresses: yuyang_lark@mail.nwpu.edu.cn (Y. Yu), zhangyi@nwpu.edu.cn (Y. Zhang), czy1102@mail.nwpu.edu.cn (Z. Cheng), hamlet@mail.nwpu.edu.cn (Z. Song), cktang@nwpu.edu.cn (C. Tang).

https://doi.org/10.1016/j.engappai.2023.107079
Received 8 May 2023; Received in revised form 28 July 2023; Accepted 28 August 2023
Available online 7 September 2023

Attention mechanisms used in visual recognition tasks can be roughly divided into three categories: channel attention, spatial attention, and mixed-domain attention. Different attention mechanisms encode feature information at different locations to strengthen the original features. However, they differ in utilizing aggregation strategies, transformations, and integration functions (Yu et al., 2022). By abstracting these functions, a three-step general framework for modeling attention mechanisms is presented: first, feature aggregation for contextual information embedding; second, feature transformation to capture inter-channel relationships or spatial information inter-dependencies; third, feature integration responsible for recalibrating the original information. In the deep learning era, numerous efforts (Yang et al., 2021; Liu et al., 2020; Hu et al., 2020; Wang et al., 2020; Lee et al., 2019; Woo et al., 2018) have revealed that properly incorporating attention mechanisms into convolution blocks is beneficial to improve the performance of networks by a large margin. One of the most impressive methods is still the Squeeze-and-Excitation (SE) module (Hu et al., 2020), which is independent of network architecture and can learn channel attention for each convolution block, bringing dramatic performance gains to a wide range of CNNs at a considerably low computational cost. Based on the pipeline in SE, Squeeze (feature aggregation) and Excitation (feature transformation), later works either attempt to reduce the complexity of the module by simplifying the Excitation phase, or boost the performance of the module by improving all phases simultaneously, thus proposing Efficient Channel Attention (ECA) (Wang et al., 2020) and Style-based Recalibration Module (SRM) (Lee et al., 2019). Their diagrams are depicted in Fig. 1(a) and Fig. 1(b), respectively. Nevertheless, these attention modules only encode inter-channel relationships while ignoring the interactions between features in the spatial dimension, which has been empirically confirmed to be crucial for more accurately locating and recognizing the objects of interest. To mitigate this problem, Convolutional Block Attention Module (CBAM) (Woo et al., 2018) integrates channel and spatial attention into one module and achieves remarkable performance improvements over its SE counterparts with a small computational overhead. However, as shown in Fig. 1(c), CBAM constructs channel and spatial attention separately, making it impossible to exploit any relation between them, inevitably resulting in a significant loss of information. Based on the above elucidations, we observe that there are still several challenging problems to be solved in attention learning. On the one hand, structurally, how to learn attention weights in both channel and spatial dimensions in a cheap but efficient way to provide richer feature representations? On the other hand, only utilizing global average pooling (GAP) to aggregate contextual information in attention learning inevitably results in the loss of detailed feature representations. Given this, how to develop an efficient mechanism for aggregating cross-dimension feature responses to strengthen the representation of feature descriptors? Further, for feature transformation, it is inefficient and redundant to introduce dimensionality reduction (such as SE and CBAM) while capturing feature interactions, making the correspondence between the dimension and its weight indirect. Consequently, how to design a feature transformation that guarantees efficiency and effectiveness to capture feature interactions better? In brief, how to investigate and develop a lightweight, efficient, and scalable attention module, which can not only model complementary attention in both channel and spatial dimensions but also effectively break down the existing barriers in feature aggregation and feature transformation in attention learning, is an urgent problem to be solved.

Based on the above discussion, in this paper, we aim to emphasize learning complementary attention in channel (what to pay attention to), height (where to pay attention in the H dimension), and width (where to pay attention in the W dimension) dimensions simultaneously, which is beneficial to inform a CNN model what to pay attention to and where to look and enables it to take full advantage of the correlation between channel and spatial attention, thus achieving better performance with fewer layers. Following this advanced philosophy, we propose a plug-and-play, lightweight yet efficient multidimensional collaborative attention module, MCA, with good generalization ability for various CNNs and datasets. As illustrated in Fig. 1(d), the proposed MCA module consists of three branches, where each branch from right to left is responsible for modeling attention in channel, width, and height dimensions, respectively. To construct core components in MCA, we propose the *squeeze transformation* and *excitation transformation* by improving the SE pipeline. Specifically, in the *squeeze transformation*, we not only utilize global average and standard deviation pooling to aggregate cross-dimension feature responses but also develop a combination mechanism for adaptively merging average-pooled and standard-deviation-pooled features to enhance the representation of feature descriptors. In the *excitation transformation*, instead of utilizing the inefficient dimensionality reduction strategy involved in SE, we adaptively capture local feature interactions in a highly lightweight way to better overcome the paradox of performance and computational overhead trade-off. More details about the MCA module can be found in Section 3.1.

To the best of our knowledge, the proposed MCA is a pioneer work capable of simultaneously inferring attention in multiple dimensions. In short, the critical contributions of this paper are summarized as follows:

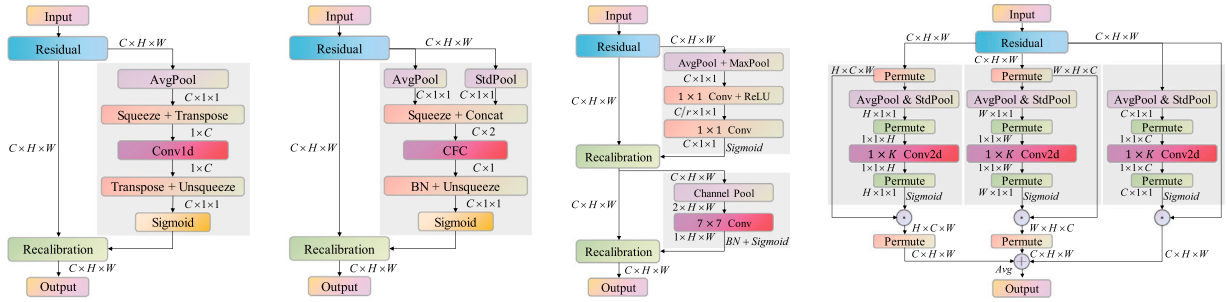
- We introduce a lightweight, efficient, generalizable attention module with a three-branch structure, MCA, which accounts for the importance of modeling multidimensional collaborative attention in an efficient manner while bringing outstanding performance improvement. Meanwhile, in the MCA module, we design the *squeeze transformation* for adaptively aggregating dual cross-dimension feature responses and develop the *excitation transformation* to capture local feature interactions adaptively.
- We conduct comprehensive analysis and ablation experiments to verify the correctness and effectiveness of each component in our design scheme.
- As a plug-and-play attention module, we demonstrate that MCA can seamlessly integrate into various well-established CNNs and significantly outperform other state-of-the-art (SOTA) attention methods on multiple benchmarks (CIFAR Krizhevsky et al., 2009 and ImageNet-1K Russakovsky et al., 2015) for image recognition with much cheaper overhead.
- We also visualize the output of sample images utilizing the Grad-CAM++ (Chattopadhay et al., 2018) technique to provide more intuitive insights into the superiority of our method.

2. Related work

In this section, we briefly review the literature for this paper, including some representative works on network architecture design and plug-and-play attention modules.

2.1. Network architectures

Since AlexNet (Kirzhevsky et al., 2012) was successfully released for image recognition, deep CNNs have come into people's view. Since then, various approaches have been proposed to enhance the representation learning ability of CNNs. Some researchers have spared no effort to design deeper network architectures, among which VGGNet (Simonyan and Zisserman, 2014) with 19 layers showed that stacking convolution blocks with the identical shape achieved better results, and the InceptionNet series (Szegedy et al., 2015, 2016) aggregated more informative and multifarious features by introducing a multi-branch architecture where each branch was carefully configured with customized kernel filters. Nevertheless, due to the difficulty of gradient backpropagation, it was found that the performance of networks can reach saturation or even degradation rapidly by simply extending the depth. Things took a turn for the better when ResNet (He et al., 2016a) was implemented. This architecture adds skip-connection from



(a) ECA (b) SRM (c) CBAM (d) MCA

Fig. 1. Schematic comparison of the proposed multidimensional collaborative attention module (d) to the ECA module (a), SRM (b), and CBAM (c). Here, *AvgPool*, *StdPool*, and *MaxPool* denote the global average pooling, global standard deviation pooling, and global max pooling, respectively. *CFC* and *BN* represent channel-wise fully connected layer and batch normalization, respectively. Besides, the feature maps are shown as feature dimensions, e.g., $C \times H \times W$ refers to a feature map with channel number C, height H, and width W. $\odot$ refers to broadcast element-wise multiplication, and $\oplus$ refers to broadcast element-wise summation.

input to output within each convolution block to ease the optimization issues, enabling networks to scale up to hundreds of layers. Following the same spirit, PreActResNet (He et al., 2016b) was proposed utilizing the identity skip-connection and after-addition activation. This architecture facilitates training and enhances generalization, pushing CNNs to thousands of layers. However, some works pointed out that blindly increasing the depth cannot continuously improve the performance of networks but will bring higher model complexity and heavier computational burden. Besides depth, a simple and intuitive way is to increase the width of networks. Following this philosophy, WideResNet (Zagoruyko and Komodakis, 2016), with a more significant number of filters and fewer layers, is developed based on the ResNet architecture. PyramidNet (Han et al., 2017) is a rigorous generalization of WideResNet, in which the network gradually becomes wider with increasing depth. Empirical studies show these networks achieve significant performance gains over their thin and deep counterparts. Instead of depth and width, ResNeXt (Xie et al., 2017) and Xception (Chollet, 2017) increase the cardinality of networks by exploiting group convolutions. They empirically demonstrate that cardinality reduces parameter overhead and performs well compared to depth and width. In addition, to improve the usability of CNNs on resource-constrained devices, researchers have put much effort into making networks more lightweight. Later works, such as the MobileNet family (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019), ShuffleNetV2 series (Zhang et al., 2018; Ma et al., 2018), and MobileNeXt (Zhou et al., 2020), also show excellent performance. Unlike these mentioned works, which either focus on depth, width, and cardinality factors or aim at designing mobile networks, our goal is to develop a plug-and-play attention module.

2.2. Attention mechanisms

Human attention can be regarded as a tool for selecting available processing resources, capable of prioritizing task-relevant information in an input signal while attenuating irrelevant ones. Such attention mechanisms have been generalized to CNNs in the way of refining feature activations and have shown great potential in image recognition. For the first time, the most representative work, SE (Hu et al., 2020), presents an effective method for learning channel attention while achieving notable performance. It first aggregates the spatial information with the help of 2D global average pooling and then utilizes two fully-connected layers with dimensionality reduction to capture inter-channel interactions. Inheriting the settings of Squeeze and Excitation in SE, later methods either put some effort into boosting the Squeeze phase (e.g., GSoP-Net Gao et al., 2019b and FcaNet Qin et al., 2021) or reduce the complexity of the Excitation phase by adopting a 1D convolution filter (e.g., ECA Wang et al., 2020), or attempt to improve all phases at the same time (e.g., SRM (Lee et al., 2019) and GCT (Yang et al., 2020)). Besides channel attention, spatial attention, regarded as an adaptive spatial region selection mechanism, plays another vital role in inferring fine attention. Some works, GE (Hu et al., 2018) and SGE (Li et al., 2019), emphasize important regions that play an active role in the network's decision-making process by predicting a soft mask in the spatial domain while suppressing the rest. Other works, NL-Net (Wang et al., 2018), GC-Net (Cao et al., 2019), SASA (Ramachandran et al., 2019), and ViT (Dosovitskiy et al., 2020), capture long-range spatial contextual information by modeling self-attention. CBAM (Woo et al., 2018) and BAM (Park et al., 2018) provide robust representative attention by effectively exploiting the advantages of channel and spatial attention. Inspired by their design scheme, SA (Zhang and Yang, 2021) adopts Shuffle Units to combine complementary channel and spatial attention efficiently. HAM (Li et al., 2022) further advances the idea of CBAM by redesigning the channel and spatial attention modules. More recently, TA (Misra et al., 2021) mainly emphasizes the importance of capturing cross-dimension interaction, and both CA (Hou et al., 2021) and LMA (Yu et al., 2022) aim at modeling long-range feature dependencies, all of which are beneficial for capturing rich discriminative feature representations.

Different from these previous methods, which either develop attention only in the channel or spatial dimension or compute channel and spatial attention independently, which inevitably results in effective information loss, our MCA mainly focuses on simultaneously modeling complementary attention in the channel, height, and width dimensions to enhance the expressive power of the learned features and precisely locate the objects of interest. It is worth emphasizing that our method performs favorably against other counterparts while having very competitive parameters and computational costs.

3. Methodology

In this section, we first introduce our MCA module in detail, including the information propagation process with a three-branch structure responsible for capturing correlations between features in different dimensions and our developed *squeeze transformation*, *excitation transformation*, and *integration strategy*. Then, we showcase how to apply MCA to the basic and bottleneck residual blocks of ResNets, which are excellent in large-scale visual recognition tasks, and further describe the architecture details of MCA-integrated ResNets for CIFAR-10/100 and ImageNet-1K classification, respectively, to provide more intuitive explanations for other researchers. Finally, we theoretically analyze and empirically verify the parameter complexity of our method and further compare it with other SOTA attention mechanisms, thus demonstrating the lightweight of our method.

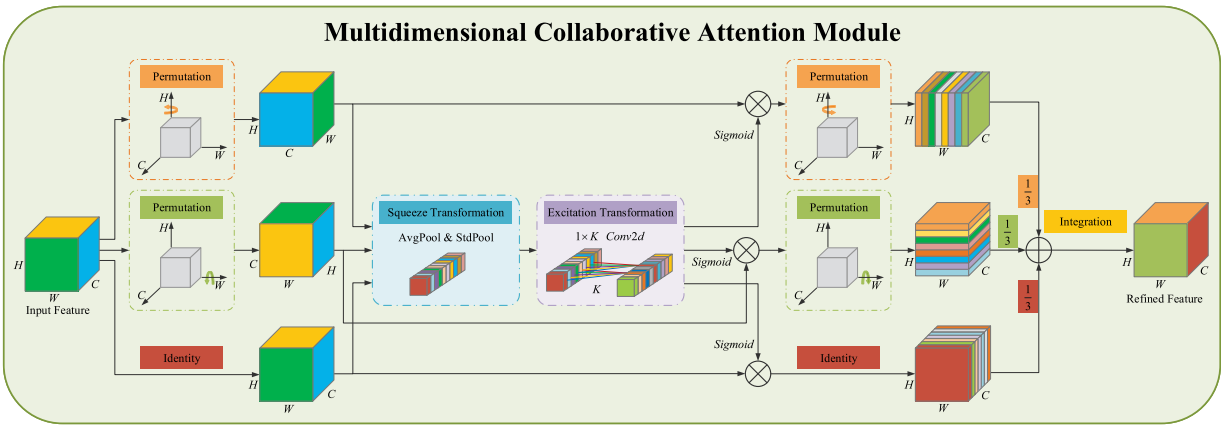


Fig. 2. The overall architecture of the proposed MCA module with three branches. The top branch is used to capture the interaction between features in the spatial dimension W. Similarly, the middle branch is used to capture the interaction between features in the spatial dimension H. The bottom branch is responsible for capturing the interaction between channels. In the first two branches, we employ the permute operation to capture long-range dependencies between the channel dimension and either one of the spatial dimensions. Finally, the outputs from all three branches are aggregated by simple averaging in the *integration* phase. Moreover, $\otimes$ denotes broadcast element-wise multiplication, and $\oplus$ denotes broadcast element-wise summation.

3.1. Multidimensional collaborative attention

As discussed in Section 1, this paper aims to investigate a better way to develop a lightweight, effective, and generalizable attention module for efficient network design. Inspired by the philosophy of cross-dimension interaction presented in TA, we conduct research from the perspective of how to capture the interaction between features in the channel, height, and weight dimensions simultaneously and successfully propose an almost parameter-free multidimensional collaborative attention module, termed MCA. A schematic of the MCA module is depicted in Fig. 2. As we can see, the MCA module is composed of three parallel branches, where the top two branches are responsible for capturing feature inter-dependencies in spatial dimensions W and H, respectively, while the last branch is mainly used to capture inter-channel interactions.

Clearly, our MCA can be regarded as a computational unit that expresses a specific transformation from an input tensor to a refined output tensor of the same shape. Specifically, let $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ be the output of a convolutional layer and the subsequent input feature map to the MCA module, where C, H, and W denote the number of channels (i.e., the number of filters), height, and width of the spatial feature map, respectively. We first feed $\mathbf{F}$ into each branch of the MCA module for further refinement. In the top branch, $\mathbf{F}$ is first rotated 90° anti-clockwise along the H axis, and the resulting rotated feature map is denoted as $\widehat{\mathbf{F}}_W \in \mathbb{R}^{W\times H\times C}$. To model the long-range dependencies between the channel dimension C and the spatial dimension H, $\widehat{\mathbf{F}}_W$ is then fed into our developed *squeeze transformation*, and the resulting aggregated feature map is described as $\hat{\mathbf{F}}_W \in \mathbb{R}^{W\times 1\times 1}$. $\hat{\mathbf{F}}_W$ is then passed into our designed *excitation transformation* to capture the interaction between features in the spatial dimension W, and the resulting width-wise feature weights are expressed as $\tilde{\mathbf{F}}_W \in \mathbb{R}^{W\times 1\times 1}$. Next, pass $\tilde{\mathbf{F}}_W$ through the sigmoid activation function to generate input-specific attention weights $\mathcal{A}_W \in \mathbb{R}^{W\times 1\times 1}$ in the W dimension. $\mathcal{A}_W$ is then applied to $\widehat{\mathbf{F}}_W$ through an element-wise multiplication operation, resulting in an enhanced feature map $\mathbf{F'}_W \in \mathbb{R}^{W\times H\times C}$. Finally, $\mathbf{F'}_W$ is rotated 90° clockwise along the H axis to obtain a feature map $\mathbf{F''}_W \in \mathbb{R}^{C\times H\times W}$ with the same shape as the original input. Mathematically, this process can be summarized as the following equations:

$$\widehat{\mathbf{F}}_W = PM_H(\mathbf{F}) \tag{1}$$

$$\hat{\mathbf{F}}_W = T_{sq}\left(\widehat{\mathbf{F}}_W\right), \tilde{\mathbf{F}}_W = T_{ex}\left(\hat{\mathbf{F}}_W\right) \tag{2}$$

$$\mathcal{A}_W = \sigma\left(\tilde{\mathbf{F}}_W\right), \mathbf{F'}_W = \mathcal{A}_W \otimes \widehat{\mathbf{F}}_W, \mathbf{F''}_W = PM_H^{-1}\left(\mathbf{F'}_W\right) \tag{3}$$

where $PM_H(\cdot)$ denotes rotation through 90° anti-clockwise along the H axis, while $PM_H^{-1}(\cdot)$ denotes the inverse, both of which can be easily implemented by the permute operation in the PyTorch toolbox (Paszke et al., 2019). $\sigma(\cdot)$ stands for the sigmoid activation function. $T_{sq}(\cdot)$ and $T_{ex}(\cdot)$ refer to the *squeeze transformation* and *excitation transformation*, respectively. Please refer to Sections 3.1.1 and 3.1.2 for a more detailed description.

Similarly, $\mathbf{F}$ is first rotated 90° anti-clockwise along the W axis for the middle branch, obtaining the rotated feature map $\widehat{\mathbf{F}}_H \in \mathbb{R}^{H\times C\times W}$. To model the inter-dependencies between the channel dimension C and the spatial dimension H and further capture inter-height interactions, $\widehat{\mathbf{F}}_H$ is then sequentially fed into the *squeeze transformation* and *excitation transformation*, from which the aggregated feature map $\hat{\mathbf{F}}_H \in \mathbb{R}^{H\times 1\times 1}$ and height-wise feature weights $\tilde{\mathbf{F}}_H \in \mathbb{R}^{H\times 1\times 1}$ can be deduced successively. Subsequently, passing $\tilde{\mathbf{F}}_H$ through the sigmoid activation function can generate input-specific attention weights $\mathcal{A}_H \in \mathbb{R}^{H\times 1\times 1}$ in the H dimension. $\widehat{\mathbf{F}}_H$ is then recalibrated by $\mathcal{A}_H$ to produce an enhanced feature map $\mathbf{F'}_H \in \mathbb{R}^{H\times C\times W}$. Eventually, $\mathbf{F'}_H$ is rotated 90° clockwise along the W axis to obtain a feature map $\mathbf{F''}_H \in \mathbb{R}^{C\times H\times W}$ with the same shape as the original input. Formally, this process is generalized as follows:

$$\widehat{\mathbf{F}}_H = PM_W(\mathbf{F}) \tag{4}$$

$$\hat{\mathbf{F}}_H = T_{sq}\left(\widehat{\mathbf{F}}_H\right), \tilde{\mathbf{F}}_H = T_{ex}\left(\hat{\mathbf{F}}_H\right) \tag{5}$$

$$\mathcal{A}_H = \sigma\left(\tilde{\mathbf{F}}_H\right), \mathbf{F'}_H = \mathcal{A}_H \otimes \widehat{\mathbf{F}}_H, \mathbf{F''}_H = PM_W^{-1}\left(\mathbf{F'}_H\right) \tag{6}$$

Here, $PM_W(\cdot)$ represents a 90° anti-clockwise rotation along the W axis, while $PM_W^{-1}(\cdot)$ represents the inverse.

The design concept of the remaining bottom branch is similar to that of some representative channel attention mechanisms (e.g., SE, ECA, and SRM), which is mainly responsible for modeling spatial (H and W) inter-dependencies and capturing the interaction between channels. In short, $\mathbf{F}$ is first passed through an identity mapping to generate its identical feature map $\widehat{\mathbf{F}}_C \in \mathbb{R}^{C\times H\times W}$ (i.e., $\widehat{\mathbf{F}}_C = \mathbf{F}$). $\widehat{\mathbf{F}}_C$ is then sequentially input into the *squeeze transformation* and *excitation transformation*, which can successively infer the aggregated feature map $\hat{\mathbf{F}}_C \in \mathbb{R}^{C\times 1\times 1}$ and channel-wise feature weights $\tilde{\mathbf{F}}_C \in {}^{C\times 1\times 1}$. After that, $\tilde{\mathbf{F}}_C$ is activated by the sigmoid function to deduce the input-specific channel attention weights $\mathcal{A}_C \in \mathbb{R}^{C\times 1\times 1}$. Subsequently, $\widehat{\mathbf{F}}_C$ is rescaled via $\mathcal{A}_C$ to generate an enhanced feature map $\mathbf{F'}_C \in \mathbb{R}^{C\times H\times W}$. In the end, $\mathbf{F'}_C$ is again mapped to the feature map $\mathbf{F''}_C \in \mathbb{R}^{C\times H\times W}$ by the identity mapping function. Similarly, the process is summed up as follows:

$$\widehat{\mathbf{F}}_C = IM(\mathbf{F}) \tag{7}$$

$$\hat{\mathbf{F}}_C = T_{sq}\left(\widehat{\mathbf{F}}_C\right), \tilde{\mathbf{F}}_C = T_{ex}\left(\hat{\mathbf{F}}_C\right) \tag{8}$$

$$\mathcal{A}_C = \sigma\left(\tilde{\mathbf{F}}_C\right), \mathbf{F}'_C = \mathcal{A}_C \otimes \widehat{\mathbf{F}}_C, \mathbf{F}''_C = IM\left(\mathbf{F}'_C\right) \tag{9}$$

where $IM(\cdot)$ refers to the identity mapping function.

Finally, in the *integration* phase, the final refined feature map can be deduced by simple average aggregation of all the outputs of the three branches recalibrated by the attention weights generated in different dimensions.

3.1.1. Squeeze: Adaptive aggregation of dual interactive information

To illustrate how to model better the cross-dimension interdependencies between the spatial dimension H or W and the channel dimension C, we first take the bottom branch as an example to introduce how to aggregate the interactive features between the spatial dimensions H and W. To aggregate feature responses across spatial dimensions, Zhou et al. showed that exploiting global average pooling is a simple and effective way (Zhou et al., 2016). In subsequent works (Yang et al., 2021; Hu et al., 2020; Wang et al., 2020), it has also been widely used to extract channel-wise feature descriptors. Although this pooling operation can capture long-range dependencies specific to spatial information, it inevitably brings about a significant loss of detailed feature representations to generate suboptimal features. Recent works, CBAM and SRM, respectively, have demonstrated that global max pooling and global standard deviation pooling operations also play unexpectedly significant roles in computing spatial information statistics and empirically confirmed that each of these two poolings combined with average pooling can further strengthen the representation of feature descriptors. On account of this, to trade off performance versus computational overhead, we use two different poolings simultaneously to aggregate feature responses rather than combining multiple poolings (three or more). Motivated by this intuition, we investigate the practical benefits of different pooling methods (see Section 4.2.2), and the resulting results prompt us to use both average and standard deviation pooling to better aggregate cross-dimension feature responses. Beyond the previous works, we argue that it is a suboptimal combination that average-pooled and standard-deviation-pooled features are permanently assigned the same weight since their credits are not the same at different stages of image feature extraction. Therefore, we design an adaptive mechanism called *squeeze transformation* to effectively combine average-pooled and standard-deviation-pooled features to improve the representation of feature descriptors significantly. See Fig. 3 for a diagram of *squeeze transformation*. Next, we elaborate on it.

As illustrated in Fig. 3, following the above definitions, given $\widehat{\mathbf{F}}_C = [\widehat{\mathbf{f}}_1, \widehat{\mathbf{f}}_2, \ldots, \widehat{\mathbf{f}}_C] \in \mathbb{R}^{C\times H\times W}$ as the input feature map, we first employ both global average and standard deviation pooling to aggregate spatial information, generating two different channel-wise feature statistics, $\widehat{\mathbf{F}}_C^{avg} = [\widehat{f}_1^{avg}, \widehat{f}_2^{avg}, \ldots, \widehat{f}_C^{avg}] \in \mathbb{R}^{C\times 1\times 1}$ and $\widehat{\mathbf{F}}_C^{std} = [\widehat{f}_1^{std}, \widehat{f}_2^{std}, \ldots, \widehat{f}_C^{std}] \in \mathbb{R}^{C\times 1\times 1}$, which denote average-pooled and standard-deviation-pooled feature descriptors, respectively. Specifically, the two pooling operations for the m-th channel can be separately formulated as follows:

$$\widehat{f}_m^{avg} = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\widehat{\mathbf{f}}_m(i,j) \tag{10}$$

$$\widehat{f}_m^{std} = \sqrt{\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\widehat{\mathbf{f}}_m(i,j) - \widehat{f}_m^{avg}\right)^2} \tag{11}$$

where $\widehat{\mathbf{f}}_m \in \mathbb{R}^{1\times H\times W}$ represents a feature map of the m-th channel of input $\widehat{\mathbf{F}}_C$. $\widehat{f}_m^{avg}$ and $\widehat{f}_m^{std}$ refer to different output feature descriptors associated with the m-th channel, respectively. Then, $\widehat{\mathbf{F}}_C^{avg}$ and $\widehat{\mathbf{F}}_C^{std}$ are fed into our designed adaptive combination mechanism, from which the channel-wise feature descriptors $\hat{\mathbf{F}}_C$ can be generated. Mathematically, this process can be expressed as follows:

$$\hat{\mathbf{F}}_C = T_{sq}\left(\widehat{\mathbf{F}}_C\right) = \frac{1}{2}\otimes\left(\widehat{\mathbf{F}}_C^{avg} \oplus \widehat{\mathbf{F}}_C^{std}\right) \oplus \alpha\otimes\widehat{\mathbf{F}}_C^{avg} \oplus \beta\otimes\widehat{\mathbf{F}}_C^{std} \tag{12}$$

where α and β are two trainable floating parameters greater than zero and less than one, which can be optimized by stochastic gradient descent (SGD). In this regard, our adaptive mechanism inherently introduces dynamics conditioned on the input, which can assign different weights to average-pooled and standard-deviation-pooled features at different stages of image feature extraction, boosting the discriminability of output feature descriptors.

Similarly, based on the above discussion, we can easily deduce the width-wise feature descriptors $\hat{\mathbf{F}}_W$ and the height-wise feature descriptors $\hat{\mathbf{F}}_H$ in the first two branches. Subsequently, we empirically confirm the effectiveness of the proposed method for adaptively aggregating dual information. See Section 4.2.2 for more details.

3.1.2. Excitation: Adaptive capture of local feature interactions

To take full advantage of the dimension-dependent feature descriptors produced in the *squeeze transformation*, they should be further transformed. Next, following the above statement, we still take the bottom branch as an example to discuss how to capture inter-channel interactions efficiently. In previous work, the SE module initially proposed to use a multi-layer perceptron (MLP) with one hidden layer to capture inter-channel correlations and confirmed the effectiveness of this operation. Soon, following this philosophy, CBAM and CA also independently used it in their channel attention modules. However, this operation involving dimensionality reduction not only makes the correspondence between the channel and its weight indirect, resulting in the loss of inter-channel relationships, but also the nonlinear global cross-channel dependencies it captures are inefficient and redundant for channel attention. This issue has been frequently pointed out in recent works (Wang et al., 2020; Misra et al., 2021; Hou et al., 2021; Yu et al., 2022). Given this, to overcome this dilemma, we need to explore a novel transformation that ensures both efficiency and effectiveness to capture the interaction between channels better. To achieve this goal, we claim that the transformation function should meet the following criteria: first, it must both capture cross-channel interaction and form the direct correspondence between the channel and its weight and second, it must learn a non-mutually-exclusive relationship that helps to emphasize multiple channels simultaneously and third, it must be lightweight and efficient. To develop this transformation that satisfies these criteria, inspired by ECA, we design a simple mechanism called *excitation transformation* that adaptively determines the coverage of interaction to capture local feature interactions between channels. Fig. 4 depicts a diagram of *excitation transformation*. Next, we describe this process in detail.

As shown in Fig. 4, with the above definitions, given the channel-wise feature descriptors $\hat{\mathbf{F}}_C = [\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_C] \in \mathbb{R}^{C\times 1\times 1}$ as an input, the corresponding channel feature weights $\tilde{\mathbf{F}}_C = T_{ex}\left(\hat{\mathbf{F}}_C\right) = [\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_C] \in \mathbb{R}^{C\times 1\times 1}$ can be inferred. In this process, for the m-th channel, by only considering the interaction between each channel and its K_C neighbors, the channel feature weight $\tilde{f}_m$ can be computed as:

$$\tilde{f}_m = \sum_{\xi=1}^{K_C} w^{\xi}\hat{f}_m^{\xi}, \hat{f}_m^{\xi} \in \boldsymbol{\Theta}_m^{K_C} \tag{13}$$

where $\boldsymbol{\Theta}_m^{K_C}$ denotes the set of feature descriptors of K_C adjacent channels associated with the m-th channel, and w^{ξ} indicates the shared learnable parameters that are not specific to the channel. Note that this transformation can be easily implemented by a 2D convolution operation with kernel size $(1, K_C)$. Given that different levels of features can be learned at different stages of image feature extraction, it is reasonable to assume that the coverage of interaction K_C is not always the same in different stages. Straightforwardly, the optimal coverage of interaction for convolutional blocks with different channel

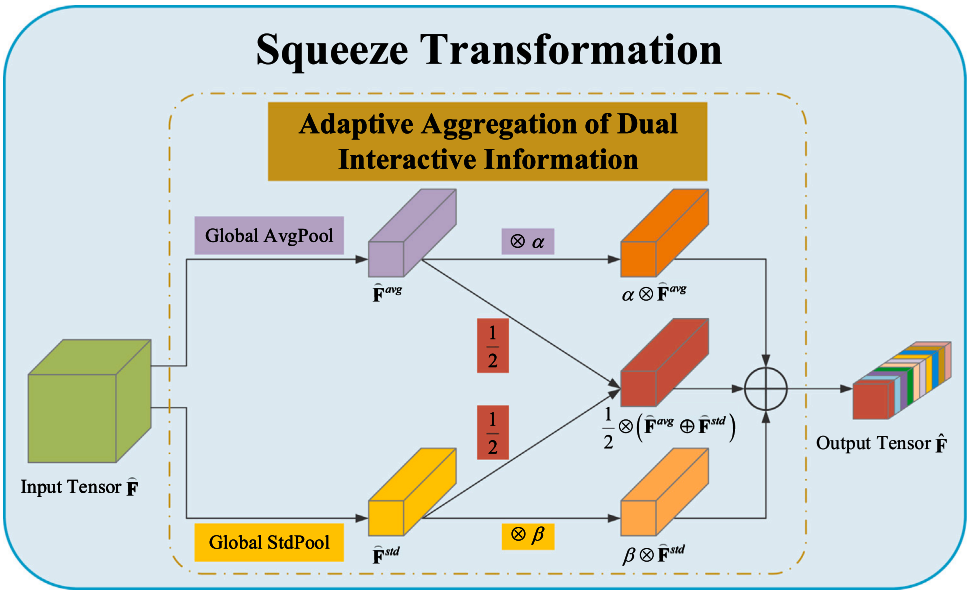


Fig. 3. Illustration of the proposed *squeeze transformation* in the MCA module. In this process, we develop an adaptive mechanism for aggregating global average and standard deviation information.

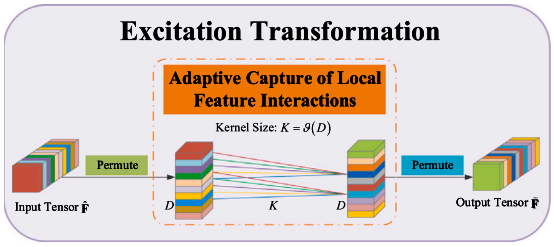


Fig. 4. Diagram of the proposed *excitation transformation* in the MCA module. Given the aggregated features obtained by the *squeeze transformation*, we design an adaptive mechanism that captures local feature interactions to generate attention weights.

numbers can be manually tuned through cross-validation. However, this approach will bring about a significant consumption of computing resources. For that reason, inspired by the philosophy contained in group convolutions (Xie et al., 2017; Huang et al., 2018; Su et al., 2020), we propose a simple gating mechanism to express the mapping between the coverage of interaction K_C and the channel dimension C by considering the following factors: first, K_C and C should be positively correlated, i.e., high-dimensional channels involve long-range interactions, and vice versa, and second, to avoid limitations, this correlation is nonlinear and third, C is usually set to a power of 2. Mathematically, this mapping can be written as:

$$C = \varphi\left(K_C\right) = 2^{(\lambda \times K_C + \gamma)} \tag{14}$$

where λ and γ are two hyperparameters. Then, considering that the kernel size K_C is always set to an odd number, given C, K_C can be approximately expressed as the following equation.

$$K_C = \vartheta(C) = \left| \frac{log_2(C) - \gamma}{\lambda} \right|_{odd} \tag{15}$$

Here, $|\ell|$ refers to the nearest odd number less than or equal to ℓ. We empirically set λ and γ to be 1.5 and 1, respectively, throughout all the experiments.

Similarly, following the above formulation, we can easily compute width-wise feature weights $\tilde{\mathbf{F}}_W$ and height-wise feature weights $\tilde{\mathbf{F}}_H$ in the first two branches. After that, we conduct empirical comparisons to analyze the effect of coverage of interaction on performance, confirming the efficiency of our method. See Section 4.2.3 for more details.

3.1.3. Integration: Collaboration with triple attention

As illustrated in Fig. 2, following the notations and operations defined above, the final refined feature map $\mathbf{F}''$ can be deduced from the augmented feature maps $\mathbf{F}''_W$, $\mathbf{F}''_H$, and $\mathbf{F}''_C$ independently generated by the three branches through a simple average summation in the *integration* phase. Formally, the process is generalized as follows.

$$\mathbf{F}'' = \frac{1}{3} \otimes \left(\mathbf{F}''_W \oplus \mathbf{F}''_H \oplus \mathbf{F}''_C\right) \tag{16}$$

Here, $\mathbf{F}''_W$, $\mathbf{F}''_H$, and $\mathbf{F}''_C$ can be respectively expressed by the following equations:

$$\mathbf{F}''_W = PM_H^{-1}\left(\sigma\left(T_{ex}\left(T_{sq}\left(PM_H\left(\mathbf{F}\right)\right)\right)\right) \otimes PM_H\left(\mathbf{F}\right)\right) \tag{17}$$

$$\mathbf{F}''_H = PM_W^{-1}\left(\sigma\left(T_{ex}\left(T_{sq}\left(PM_W\left(\mathbf{F}\right)\right)\right)\right) \otimes PM_W\left(\mathbf{F}\right)\right) \tag{18}$$

$$\mathbf{F}''_C = IM\left(\sigma\left(T_{ex}\left(T_{sq}\left(IM\left(\mathbf{F}\right)\right)\right)\right) \otimes IM\left(\mathbf{F}\right)\right) \tag{19}$$

To sum up, based on the above discussion, it is clear that our MCA module can both be responsible for determining what to pay attention to in the bottom branch and where to pay attention in the first two branches. In this regard, MCA explicitly introduces a dimension-specific triple attention collaboration mechanism, which helps locate objects of interest more precisely and boost feature discriminability. Next, we conduct comprehensive experiments to verify the superiority of our method. See Section 4 for detailed experimental results and analysis.

3.2. Instantiations

Here, we mainly take the basic building units (i.e., the basic residual block and the bottleneck residual block) in ResNets as examples to illustrate how to embed the proposed MCA module into them to comprehensively demonstrate the advantages of our method over other SOTA attention methods. For a fair comparison, by maintaining the same configuration as some representative attention networks (Hu et al., 2020; Wang et al., 2020; Lee et al., 2019; Misra et al., 2021) and just replacing their attention modules with our MCA at corresponding positions in the network, the resulting networks are named MCA-Nets. Fig. 5 clearly shows the exact location of our MCA when it is seamlessly plugged into the basic building units in ResNets. Beyond that, based on similar schemes, our MCA can also be embedded into various well-established CNNs, e.g., PreActResNet (He et al., 2016b),

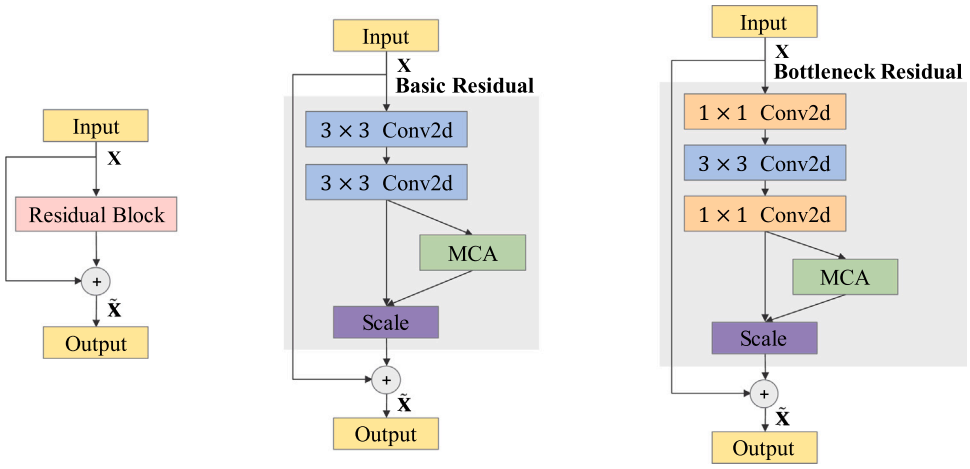


Fig. 5. Schema of the original residual block (left), MCA integrated into a basic residual block (middle), and MCA integrated into a bottleneck residual block (right).

Table 1
Architecture details of the constructed MCA-ResNet-$6n+2$ ($n = \{3, 5, 7, 9, 18\}$) for CIFAR-10/100. Each row denotes a sequence of 1 or more identical building blocks, repeated n times, and shapes and operations with specific parameter settings of a building block are shown inside the brackets. Each convolutional layer is followed by a batch normalization layer and ReLU activation function. Down-sampling is performed by the first building block of each stage with a stride of 2, except for the first stage. The k–d fc operation in the last row refers to a fully-connected layer with k classes.

Stage	Input	MCA-ResNet-$6n+2$ $n = \{3, 5, 7, 9, 18\}$	Output
Starting	$32 \times 32 \times 3$	3×3, 16, stride 1	$32 \times 32 \times 16$
1	$32 \times 32 \times 16$	$\begin{bmatrix} \text{conv}, 3 \times 3, 16 \\ \text{conv}, 3 \times 3, 16 \\ \text{MCA} \end{bmatrix} \times n$	$32 \times 32 \times 16$
2	$32 \times 32 \times 16$	$\begin{bmatrix} \text{conv}, 3 \times 3, 32 \\ \text{conv}, 3 \times 3, 32 \\ \text{MCA} \end{bmatrix} \times n$	$16 \times 16 \times 32$
3	$16 \times 16 \times 32$	$\begin{bmatrix} \text{conv}, 3 \times 3, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{MCA} \end{bmatrix} \times n$	$8 \times 8 \times 64$
Ending	$8 \times 8 \times 64$	Global average pool, k–d fc, softmax	$1 \times 1 \times k$

MobileNet-v2 (Sandler et al., 2018), WideResNet (Zagoruyko and Komodakis, 2016), and ResNeXt (Xie et al., 2017), to construct more variants. Furthermore, for concrete examples of MCA-Nets, the detailed descriptions of MCA-ResNet-$6n$+2 ($n = \{3, 5, 7, 9, 18\}$) for both CIFAR-10 and CIFAR-100 are presented in Table 1, and the detailed configurations of MCA-ResNet-18, MCA-ResNet-34, and MCA-ResNet-50 for CIFAR-100 and ImageNet-1K datasets are given in Table 2.

3.3. Parameter complexity analysis

For the proposed MCA module to be feasible in practice, it is designed to be lightweight in both terms of parameters and computational complexity. To confirm this claim, we first theoretically analyze the additional parameters introduced by MCA. All its parameters come from the *squeeze* and *excitation transformation* phases, accounting only for a small fraction of the total network parameters. More precisely, for the bottom branch, the number of parameters for each phase is $\sum_{s=1}^{S} N_s \times 2$ and $\sum_{s=1}^{S} N_s \times K_C^s$, respectively, where S represents the number of stages (where each stage refers to the set of convolutions operating on feature maps of the identical spatial dimension), N_s denotes the number of repeated building blocks at the s-th stage, and K_C^s refers to the coverage of cross-channel interaction mapped by the channel dimension C for s-th stage, which can be expressed as $K_C^s = \vartheta(C_s) = \left|\frac{log_2(C_s)-\gamma}{\lambda}\right|_{odd}$ according to Eq. (15). Thus, the total parameters introduced by this branch are calculated as $\sum_{s=1}^{S} N_s \times (K_C^s + 2)$.

Similarly, the parameters brought by the first two branches can be denoted as $\sum_{s=1}^{S} N_s \times (K_W^s + 2)$ and $\sum_{s=1}^{S} N_s \times (K_H^s + 2)$, respectively, where K_W^s (K_H^s) refers to the coverage of cross-width (cross-height) interaction mapped by the dimension W (H) for stage s, which are computed as $K_W^s = \vartheta(W_s) = \left|\frac{log_2(W_s)-\gamma}{\lambda}\right|_{odd}$ and $K_H^s = \vartheta(H_s) = \left|\frac{log_2(H_s)-\gamma}{\lambda}\right|_{odd}$, respectively. In summary, the additional parameters introduced by MCA can be derived by simply summing the parameters contained in each branch, which can be accurately expressed as:

$$\sum_{s=1}^{S} N_s \times (K_C^s + K_H^s + K_W^s + 6) \tag{20}$$

Obviously, it can be concluded that the additional parameters brought by the proposed MCA are usually negligible compared to other counterparts presented in Table 3, e.g., SE's $\frac{2}{r}\sum_{s=1}^{S} N_s \times C_s^2$ and SRM's $6\sum_{s=1}^{S} N_s \times C_s$, where r represents the reduction ratio used in the bottleneck of the MLP, and C_s denotes the dimension of the output channel in the s-th stage, here $C_s \gg K_C^s \geq K_H^s = K_W^s$.

Next, utilizing ResNet-50 as the baseline and embedding the attention layer into each of its basic building blocks, we empirically verify the parameter efficiency of MCA compared to other counterparts. The evaluation results are shown in Table 3, where C denotes the output channel dimension, G refers to the number of groups, f is the kernel size for 2D convolution, and K is the kernel size for 1D convolution. For a fair and consistent comparison, we adopt standard hyperparameter settings, where r, f, and G are set to 16, 7, and 64, respectively, while K for ECA and K_C, K_H, and K_W for MCA are all adaptively determined by the dimension of the feature map. From Table 3, it can be clearly observed that the parameter overhead introduced by MCA is almost free compared to SE and CBAM, e.g., about 0.12‰ of SE, and is 1 to 2 orders of magnitude less than TA, SGE, SRM, e.g., about 5‰ of SRM,

Table 2
Specifications for the constructed MCA-ResNet-18 (left), MCA-ResNet-34 (middle), and MCA-ResNet-50 (right) for ImageNet-1K and CIFAR-100. The configuration presented in the first row of the starting stage is for ImageNet-1K, and the second row is for CIFAR-100. Shapes and operations with specific parameter settings of a building block are listed inside the brackets, and the number of stacked building blocks in a stage is described outside. Each convolutional layer is followed by a batch normalization layer and ReLU activation function. Down-sampling is performed by the first building block of each stage with a stride of 2, except for the first stage. The $k{-}d\ fc$ operation in the ending stage denotes a fully-connected layer with k classes.

Stage	MCA-ResNet-18	MCA-ResNet-34	MCA-ResNet-50	Output	
				ImageNet-1K	CIFAR-100
Starting	(7 × 7, 64, stride 2), (3 × 3, max pool, stride 2)			56 × 56	–
	3 × 3, 64, stride 1			–	32 × 32
1	$\begin{bmatrix} \text{conv}, 3\times3, 64 \\ \text{conv}, 3\times3, 64 \\ \text{MCA} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{conv}, 3\times3, 64 \\ \text{conv}, 3\times3, 64 \\ \text{MCA} \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 64 \\ \text{conv}, 3\times3, 64 \\ \text{conv}, 1\times1, 256 \\ \text{MCA} \end{bmatrix} \times 3$	56 × 56	32 × 32
2	$\begin{bmatrix} \text{conv}, 3\times3, 128 \\ \text{conv}, 3\times3, 128 \\ \text{MCA} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{conv}, 3\times3, 128 \\ \text{conv}, 3\times3, 128 \\ \text{MCA} \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1\times1, 128 \\ \text{conv}, 3\times3, 128 \\ \text{conv}, 1\times1, 512 \\ \text{MCA} \end{bmatrix} \times 4$	28 × 28	16 × 16
3	$\begin{bmatrix} \text{conv}, 3\times3, 256 \\ \text{conv}, 3\times3, 256 \\ \text{MCA} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{conv}, 3\times3, 256 \\ \text{conv}, 3\times3, 256 \\ \text{MCA} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1\times1, 256 \\ \text{conv}, 3\times3, 256 \\ \text{conv}, 1\times1, 1024 \\ \text{MCA} \end{bmatrix} \times 6$	14 × 14	8 × 8
4	$\begin{bmatrix} \text{conv}, 3\times3, 512 \\ \text{conv}, 3\times3, 512 \\ \text{MCA} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{conv}, 3\times3, 512 \\ \text{conv}, 3\times3, 512 \\ \text{MCA} \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 512 \\ \text{conv}, 3\times3, 512 \\ \text{conv}, 1\times1, 2048 \\ \text{MCA} \end{bmatrix} \times 3$	7 × 7	4 × 4
Ending	Global average pool, $k{-}d\ fc$, softmax			1 × 1	1 × 1

Table 3
Comparisons of various existing attention modules in terms of their parameter complexity and overhead using ResNet-50 as the baseline. *+Parameters-to-Baseline* refers to the number of parameters increased compared to the baseline.

Attention Modules	Parameters	Overhead (+Parameters-to-Baseline)
Baseline (No Attention)	N/A	0
SE	$2C^2 / r$	2.5310 M
CBAM	$2C^2 / r + 2f^2$	2.5326 M
ECA	K	0.0001 M
TA	$6f^2$	0.0048 M
SRM	$6C$	0.0604 M
SGE	$2G$	0.0020 M
SA	$3C / G$	0.0007 M
MCA (Ours)	$K_C + K_H + K_W + 6$	0.0003 M

while also being quite competitive compared to ECA and SA, showing that our MCA is a really lightweight module.

4. Experiments and results

In this section, we perform extensive experiments on standard benchmarks for the challenging vision task of image recognition, comprehensively investigating the performance and practical benefits of the proposed MCA module across a range of datasets and model architectures. We first describe the setup of all experiments, followed by conducting a series of ablation experiments to confirm the contribution of each component in the MCA module to the performance and the validity of our design scheme. Second, we seamlessly incorporate MCA into various representative network architectures and consistently achieve leading performance against other attention modules on different benchmarks, demonstrating MCA's lightweight, effectiveness, and generalization ability. Finally, we provide Grad-CAM++ (Chattopadhay et al., 2018) visualizations for sample images from the ImageNet-1K validation set to intuitively showcase our method's ability to capture salient informative features.

4.1. Implementation details

To thoroughly evaluate the performance of the MCA module, we perform all experiments on the CIFAR (Krizhevsky et al., 2009) and ImageNet-1K (Russakovsky et al., 2015) benchmarks that are widely used in evaluating the representation of networks, using various well-known baseline architectures. The CIFAR dataset comprises two variants, CIFAR-10 with 10 classes and CIFAR-100 with 100 classes. Both variants consist of 60K 32 × 32 color images, of which 50K images are used for training, and 10K images are applied for testing. Concretely, the CIFAR-10/100 dataset contains 6000/600 images per class, including 5000/500 training images and 1000/100 test images. The CIFAR-10/100 dataset is divided into 5 training batches and 1 test batch, each with 10K images. The test batch contains exactly 1000/100 randomly-selected images from each class, while the training batches contain the remaining images in random order. Without bells and whistles, all experiments on CIFAR-10/100 follow the same standard training configuration as He et al. (2016a) and Yang et al. (2021) for data augmentation and optimization to facilitate useful comparative analysis between models. Specifically, for data augmentation, each

image is zero-padded with 4 pixels on each side, and a 32 × 32 patch is randomly cropped from the padded image or its horizontal flip. The input images are then normalized using RGB mean values and standard deviations. For optimization, we use a synchronous SGD optimizer with weight decay of 5e-4, momentum of 0.9, and mini-batch size of 128, and set the initial learning rate to 0.1 and divide it by 5 at the 60th, 120th, and 160th epochs, to train all models from scratch for 200 epochs on two RTX A6000 GPUs. Differently, ResNet-6n + 2 (n = $\{3, 5, 7, 9, 18\}$) are optimized using a learning scheme with an initial learning rate of 0.1, divided by 10 at the 100th and 150th epochs, while MobileNet-v2 is optimized with a cosine learning schedule with an initial learning rate of 0.1. During testing, all models only accept the original images. Simultaneously, the ImageNet-1K dataset consists of 1.28 million training images and 50K validation images from 1000 classes. The number of images for each class ranges from 732 to 1300 in the training set, while there are precisely 50 images per class in the validation set. For all experiments conducted on ImageNet-1K, we follow the same data augmentation and hyper-parameter settings in He et al. (2016a), Wang et al. (2020) and Zhang and Yang (2021) for fair and consistent comparison with other methods. Precisely, the input images are randomly cropped to 224 × 224 patches, and a random horizontal flipping with a probability of 0.5 is applied. Then, the input images are normalized by practical mean channel subtraction. During training, all networks are trained from scratch using synchronous SGD with weight decay of 1e-4, momentum of 0.9, and mini-batch size of 256 on four RTX A6000 GPUs within 100 epochs, starting from the initial learning rate of 0.1 and decreasing it by a factor of 10 per 30 epochs. For testing on the validation set, the shorter side of the input image is first resized to 256, and a single center crop of 224 × 224 patches is performed for evaluation. It is worth noting that we train all networks on the training set utilizing naive softmax cross-entropy without label-smoothing regularization and report the average classification accuracy over 5 runs for each network on the test set to allow better comparison of significant differences between different models. Moreover, we re-implement all competitors under identical settings using the PyTorch toolbox (Paszke et al., 2019) and report our reproduced results throughout the experiments to achieve apple-to-apple comparisons.

4.2. Ablation studies

In this section, we conduct extensive ablation experiments to understand better the effect of employing different configurations on each component of MCA and gain insight into the superior performance of MCA. Our ablation experiments are divided into three aspects. First, we verify the importance of each branch in MCA and demonstrate the validity of our design choice. Second, we explore the influence of utilizing different pooling operators and combination mechanisms on the representation power of MCA in the *squeeze transformation*. Finally, we study how the behavior of MCA varies with different coverage of local feature interactions in the *excitation transformation*. The experimental results and analysis are presented below.

4.2.1. Importance of multidimensional attention

To delve into the importance of simultaneously modeling attention in multiple dimensions, utilizing ResNet-56 as the base model, we perform a series of ablation experiments on CIFAR-10/100 to observe the impact of each branch in MCA on model performance. The corresponding experimental results are shown in Table 4. For the convenience of labeling, involving only channel attention, i.e., the first two branches of MCA are switched off, is denoted as MCA (Channel). Similarly, MCA (Height) and MCA (Weight) refer to modeling only height or width attention, respectively, and MCA (Spatial) represents the combination of height and width, i.e., the bottom branch of MCA is switched off, while MCA (Ours) denotes multidimensional collaborative attention where all branches are turned on. As shown in Table 4, MCA (Channel) consistently achieves better Top-1 accuracy on CIFAR-10/100 against the competing channel attention modules ECA and SRM, which objectively demonstrates that our *squeeze transformation* and *excitation transformation* are beneficial for general performance improvement. Meanwhile, MCA (Height) and MCA (Weight) with similar results under consistent settings have better or comparable performance to ECA, SRM, and CBAM, while MCA (Spatial) can further promote performance improvement. These results reflect that these two kinds of attention can enhance feature representations from different perspectives, and their effects are complementary. Intuitively, we judge that the reason for the above phenomenon is that MCA (Channel), MCA (Height) and MCA (Weight) can learn attention in the channel, height and width dimensions respectively, which is beneficial to inform the model what to pay attention to or where to pay attention. In addition, it can be observed that our MCA with three branches can always yield the best results with minimal additional overhead and can attain 0.7% and 0.78% improvement in terms of Top-1 Acc on CIFAR-10 and CIFAR-100, respectively, compared to ECA with excellent performance. These results support our design idea that attention based on different dimensions can play different roles in improving attention inference, and they are mutually orthogonal, thus further verifying the effectiveness of our design scheme.

4.2.2. Impact of different pooling mechanisms

To strictly examine the impact of different pooling mechanisms configured in *squeeze transformation* on the performance of MCA, we perform ablation experiments on standard CIFAR-10/100 under the ResNet-56 model, and the resulting experimental results are listed in Table 5. Firstly, we compare the behavior of three pooling methods: AvgPool, MaxPool, and StdPool. As can be seen from the results reported in Table 5, while each of them is effective compared to the baseline, both AvgPool and StdPool with competitive performance can yield better results than MaxPool, suggesting that choosing them as pooling operators is more helpful for attention inference. Then, we adopt the mechanism illustrated in Lee et al. (2019) and Woo et al. (2018) for merging pooled features, i.e., element-wise summation, to explore the performance change brought about by combining any two of these three poolings. We notice that combining AvgPool or StdPool with MaxPool gives rise to worse results than exploiting AvgPool or StdPool independently, while taking both AvgPool and StdPool into account attains better results unexpectedly. Intuitively, this phenomenon is because the average-pooled and standard-deviation-pooled features have a complementary relationship, while they are incompatible style features with max-pooled features. These results reflect that utilizing both AvgPool and StdPool to aggregate feature responses in our *squeeze transformation* is reasonable. Moreover, we further validate the practical benefits of the adaptive combination mechanism depicted in Fig. 3 against element-wise summation. The corresponding experimental results indicate that exploiting our adaptive mechanism to merge pooled features can always bring better performance than employing element-wise summation. More importantly, our method (AvgPool & StdPool) can yield the highest accuracies. These results both support our claim that the contributions of different pooled features are not always equal at different stages of image feature extraction and demonstrate the correctness and superiority of our design choice.

4.2.3. Effect of coverage of local feature interactions

As discussed in Section 3.3, the coverage of local feature interactions K involved in *excitation transformation*, i.e., the kernel size in Eq. (15), is a crucial factor controlling the number of parameters of the MCA module. Here, for the sake of description, K refers to the general term for kernel size across all dimensions, namely $K = K_C = K_H = K_W$. In this part, we assess the effect of different K on MCA and validate the efficiency and effectiveness of our method for adaptively determining K. We perform ablation experiments on ImageNet-1K by setting K from 3 to 7 with a wider ResNet-34 as the baseline network.

Table 4
Effect of different branches in the MCA module on CIFAR-10/100 classification performance when taking ResNet-56 as the baseline. Param. and FLOPs indicate the number of parameters and floating-point operations, respectively, and Top-1 Acc stands for top-1 classification accuracy. All results are the average of 5 runs.

Description	CIFAR-10			CIFAR-100		
	Param. (M)	FLOPs (G)	Top-1 Acc (%)	Param. (M)	FLOPs (G)	Top-1 Acc (%)
ResNet-56 (Baseline)	0.856	0.127	93.49	0.862	0.127	71.51
ResNet-56 + CBAM	0.866	0.129	93.67	0.871	0.129	72.12
ResNet-56 + SRM	0.860	0.127	94.05	0.866	0.127	72.25
ResNet-56 + ECA	0.856	0.128	94.08	0.862	0.128	72.33
ResNet-56 + MCA (Spatial)	0.856	0.128	94.27	0.862	0.129	72.68
ResNet-56 + MCA (Height)	0.856	0.128	94.13	0.862	0.128	72.21
ResNet-56 + MCA (Weight)	0.856	0.128	94.09	0.862	0.128	72.35
ResNet-56 + MCA (Channel)	0.856	0.128	94.19	0.862	0.128	72.52
ResNet-56 + MCA (Ours)	0.856	0.128	94.78	0.862	0.128	73.11

Table 5
Effect of different pooling methods configured in the MCA module on CIFAR-10/100 classification performance when taking ResNet-56 as the baseline. $\oplus$ refers to broadcast element-wise summation, and & denotes our developed adaptive combination mechanism. All results are the average of 5 runs.

Description	CIFAR-10 Top-1 Acc (%)	CIFAR-100 Top-1 Acc (%)
ResNet-56 (Baseline)	93.49	71.51
ResNet-56 + AvgPool	94.32	72.49
ResNet-56 + MaxPool	93.87	72.12
ResNet-56 + StdPool	94.18	72.38
ResNet-56 + AvgPool $\oplus$ MaxPool	93.99	72.29
ResNet-56 + MaxPool $\oplus$ StdPool	93.98	72.18
ResNet-56 + AvgPool $\oplus$ StdPool	94.51	72.68
ResNet-56 + AvgPool & MaxPool	94.30	72.51
ResNet-56 + MaxPool & StdPool	94.31	72.42
ResNet-56 + MCA (Ours)	94.78	73.11

The evaluation results are summarized in Table 6, from which we can derive the following insights. First, it is easy to see that either choosing a fixed K in all stages of the network or adaptively computing K utilizing our method can outdo the baseline in classification accuracy with trivial additional overhead, demonstrating the correctness of our design concept of capturing local feature interactions. Second, when K is fixed, it can be noticed that as K varies from 3 to 7, the additional overhead introduced increases continuously, but its performance is not monotonically improved but reaches the optimal at $K = 5$. These results reflect that the network performance fluctuates to some extent with the change of K. We speculate that the reason may be that the coverage of feature interactions in different stages of image feature extraction is not always the same but should be positively correlated with the dimension size of the feature map. Besides, when adaptively determining K with our proposed method, we can consistently attain better performance than utilizing a fixed K while achieving a good trade-off between performance and overhead. For example, compared with the optimal result (when $K = 5$), our method can achieve gains of 0.22% and 0.05% in Top-1 and Top-5 Acc, respectively, while reducing 26.79% and 0.36% in the number of additional parameters and FLOPs introduced. Meanwhile, our approach can significantly improve the baseline at almost free computational overhead. As a brief conclusion, these empirical results demonstrate the effectiveness of our approach in attaining better and more stable performance while validating its lightweight.

4.3. Image classification on CIFAR-10/100

4.3.1. Comparisons using thinner ResNet with different depths

With thinner ResNet of various depths as backbones, i.e., ResNet with 20, 32, 44, 56, and 110 layers, we thoroughly evaluate the superior performance of the MCA module by comparing with several SOTA attention methods on CIFAR-10/100, including SE, CBAM, SRM, ECA, TA, and CA. Throughout the experiments, we comply with the procedure specified in Section 4.1, using evaluation metrics including efficiency (i.e., Parameters and FLOPs) and effectiveness (i.e., Top-1 Acc). The experimental results are listed in Table 7, from which we can see that no matter which backbone is considered, no matter which dataset is selected, the MCA-integrated model can significantly outperform all corresponding baselines with almost the same overhead, demonstrating that our MCA is indeed lightweight and can enhance the representational learning capability of models. Meanwhile, from Table 7, we observe that compared with other competing counterparts, our MCA can always reach the best accuracies across different baselines and datasets while only introducing the fewest or almost equivalent additional parameters and FLOPs, which reveals that our approach is more efficient in attention inference. For example, regarding Top-1 Acc, MCA provides 0.55%/0.39%, 0.68%/0.37%, and 0.64%/0.56% gains over the most competitive TA on CIFAR-10/100 under the settings of 20, 44 and 56 layers, respectively, while benefiting much lower additional overhead. These comparisons imply that our approach is more powerful, indicating the efficacy of the *squeeze transformation* capable of aggregating dual interactive features, the *excitation transformation* that adaptively captures local feature interactions, and a three-branch architecture for modeling multidimensional collaborative attention. More importantly, the ResNet-32 augmented with MCA obtains a Top-1 accuracy of 93.88%/71.23% on CIFAR-10/100, exceeding the deeper vanilla ResNet-44 (93.33%/70.81%) by 0.55%/0.42% with only about two-thirds of the model parameters (0.467 M/0.473 M vs. 0.661 M/0.667 M) and FLOPs (0.070 G vs. 0.099 G). Surprisingly, this phenomenon is repeated for deeper models, suggesting that while the MCA module itself adds depth, it does so in a highly efficient manner. In addition, we depict the training and validation curves of ResNets with MCA and other counterparts in Fig. A.7. See Appendix A for further details.

In brief, all these results not only demonstrate that our MCA is lightweight yet efficient and is robust and generalizable to different depth networks and various fine-grained datasets in accuracy but also further reflect that the improvements induced by MCA can be used in combination with increasing the depth of the base model.

4.3.2. Comparisons using wider ResNet with different depths

To in-depth study the advantages of combining the MCA module with wider ResNets, we rigorously assess MCA on CIFAR-100 using ResNet with 18, 34, and 50 layers as baselines in comparison with several advanced counterparts. Throughout the experiments, we follow the exact training rules (see Section 4.1) and network specifications (see Table 2) to allow a fair and consistent comparison with other competitors. All experimental results are recorded in Table 8. These experimental results of comparisons show similar performance trends to previous experiments, i.e., our method can match or far outstrip the corresponding baselines and other competitors in all evaluation

Table 6
Comparisons of various MCA modules equipped with different kernel sizes K on ImageNet-1K validation when using ResNet-34 as the baseline. *+Parameters-to-Baseline* and *+FLOPs-to-Baseline* represent the number of parameters and FLOPs increased compared to the baseline, respectively, and Top-5 Acc stands for top-5 classification accuracy. All results are the average of 5 runs.

Description	+Parameters-to-Baseline (K)	+FLOPs-to-Baseline (M)	Top-1 Acc (%)	Top-5 Acc (%)
ResNet-34 (Baseline)	0	0	73.31	91.42
ResNet-34 + MCA ($K = 3$)	0.240	1.653	74.36	92.10
ResNet-34 + MCA ($K = 5$)	0.336	1.662	74.63	92.17
ResNet-34 + MCA ($K = 7$)	0.432	1.671	74.44	92.13
ResNet-34 + MCA (Ours)	0.246	1.656	74.85	92.22

Table 7
Comparisons of various attention methods on the CIFAR-10/100 test sets using ResNets as backbones in terms of network parameters (Parameters), floating-point operations (FLOPs), and Top-1 accuracy (%). All results are the average of 5 runs. Best records are marked in bold.

Methods	Backbones	CIFAR-10			CIFAR-100		
		Parameters (M)	FLOPs (G)	Top-1 Acc (%)	Parameters (M)	FLOPs (G)	Top-1 Acc (%)
ResNet-20	ResNet-20	**0.272**	**0.041**	91.88	**0.278**	**0.041**	67.96
+ SE		0.275	0.041	92.33	0.281	0.042	68.68
+ CBAM		0.276	0.042	92.54	0.282	0.042	68.75
+ SRM		0.274	0.041	92.27	0.280	0.041	68.20
+ ECA		0.272	0.041	92.60	0.278	0.042	68.84
+ TA		0.275	0.043	92.71	0.281	0.043	69.06
+ CA		0.282	0.042	92.57	0.287	0.042	68.62
+ MCA (Ours)		**0.272**	**0.041**	**93.26**	**0.278**	**0.041**	**69.45**
ResNet-32	ResNet-32	**0.467**	**0.070**	92.75	**0.473**	**0.070**	69.62
+ SE		0.471	0.070	92.99	0.477	0.070	70.32
+ CBAM		0.472	0.071	92.95	0.478	0.071	70.19
+ SRM		0.469	0.070	93.26	0.475	0.070	70.36
+ ECA		0.467	0.070	93.21	0.473	0.070	70.44
+ TA		0.471	0.072	93.61	0.477	0.072	70.64
+ CA		0.482	0.071	93.30	0.488	0.071	70.41
+ MCA (Ours)		**0.467**	**0.070**	**93.88**	**0.473**	**0.070**	**71.23**
ResNet-44	ResNet-44	**0.661**	**0.099**	93.33	**0.667**	**0.099**	70.81
+ SE		0.667	0.099	93.39	0.673	0.099	71.26
+ CBAM		0.669	0.100	93.44	0.675	0.100	71.18
+ SRM		0.664	0.099	93.62	0.670	0.099	71.41
+ ECA		0.661	0.099	93.65	0.667	0.099	71.48
+ TA		0.668	0.102	93.87	0.673	0.102	71.55
+ CA		0.682	0.100	93.53	0.688	0.100	71.40
+ MCA (Ours)		**0.661**	**0.099**	**94.55**	**0.667**	**0.099**	**71.92**
ResNet-56	ResNet-56	**0.856**	**0.127**	93.49	**0.862**	**0.127**	71.51
+ SE		0.863	0.128	93.54	0.869	0.128	72.15
+ CBAM		0.866	0.129	93.67	0.871	0.129	72.12
+ SRM		0.860	0.127	94.05	0.866	0.127	72.25
+ ECA		0.856	0.128	94.08	0.862	0.128	72.33
+ TA		0.864	0.131	94.14	0.870	0.131	72.55
+ CA		0.883	0.128	93.89	0.888	0.128	72.25
+ MCA (Ours)		**0.856**	0.128	**94.78**	**0.862**	0.128	**73.11**
ResNet-110	ResNet-110	**1.731**	**0.256**	93.85	**1.737**	**0.256**	72.79
+ SE		1.745	0.257	94.19	1.751	0.257	73.70
+ CBAM		1.750	0.259	94.17	1.756	0.259	73.01
+ SRM		1.739	0.256	94.35	1.745	0.256	73.84
+ ECA		1.731	0.257	94.43	1.737	0.257	73.97
+ TA		1.747	0.264	94.51	1.753	0.264	74.06
+ CA		1.784	0.258	94.01	1.790	0.258	73.78
+ MCA (Ours)		**1.731**	0.257	**95.14**	**1.737**	0.257	**74.36**

Table 8
Comparisons of efficiency (i.e., Parameters and FLOPs) and effectiveness (i.e., Top-1 Acc) of different attention methods on CIFAR-100 test set when taking ResNet with 18, 34, and 50 layers as backbones. All results are the average of 5 runs. Best records are marked in bold.

Methods	Backbones	CIFAR-100		
		Parameters (M)	FLOPs (G)	Top-1 Acc (%)
ResNet-18	ResNet-18	**11.220**	**0.557**	76.02
+ SE		11.309	0.558	77.88
+ CBAM		11.310	0.558	77.91
+ ECA		11.220	0.558	78.12
+ TA		11.223	0.561	78.26
+ MCA (Ours)		**11.220**	0.558	**79.11**
ResNet-34	ResNet-34	**21.328**	**1.162**	76.95
+ SE		21.489	1.163	78.41
+ CBAM		21.491	1.163	78.68
+ ECA		21.328	1.163	78.83
+ TA		21.333	1.169	78.86
+ MCA (Ours)		**21.328**	1.163	**79.83**
ResNet-50	ResNet-50	**23.705**	**1.308**	77.79
+ SE		26.236	1.312	79.76
+ CBAM		26.238	1.313	80.19
+ ECA		23.705	1.310	80.42
+ TA		23.710	1.335	80.64
+ SRM		23.766	1.308	80.77
+ SGE		23.707	1.310	80.83
+ SA		23.706	1.312	81.17
+ MCA (Ours)		**23.705**	1.312	**81.58**

metrics, again proving that MCA has impressive performance. In particular, MCA-ResNet-18 has a Top-1 accuracy of 79.11%, surpassing the vanilla ResNet-34 (76.95%) by 2.16% with almost half of the number of parameters (11.220 M vs. 21.328 M) and FLOPs (0.558 G vs. 1.162 G). This pattern is repeated in the deeper ResNet-50, which validates that MCA can indeed be done in an extremely efficient way in terms of increasing network depth. Furthermore, by comparing with the experiments performed in Section 4.3.1, it can be known that MCA is robust and generalizable for increasing the width of the network, and these comparisons further reflect the improvements induced by MCA may be complementary to those achieved by increasing the width of the base network.

4.3.3. Comparisons using other baseline architectures

At the end of this part, we rigorously evaluate the practical benefits of the proposed MCA module in various network architectures, including PreActResNet, MobileNet-v2, WideResNet, and ResNeXt. For a fair and consistent comparison, we conduct all experiments using the CIFAR-10 and 100 datasets and following the same training policy specified in Section 4.1. The corresponding experimental results are given in Table 9. As can be seen, no matter which dataset is considered, the network augmented with MCA can consistently outperform all corresponding baselines by a large margin in Top-1 Acc with almost the same overhead, indicating that our MCA is indeed lightweight and efficient and has strong robustness and generalization ability for various network architectures. For example, for MobileNet-v2 widely deployed on low-end devices, the MCA-integrated variant can provide 0.84%/1.23% performance gains in terms of Top-1 Acc on CIFAR-10/100 while only introducing a justified negligible additional overhead. In the meantime, the experimental results in ResNeXt-29 ($8 \times 64d$) also show similar performance trends, where the MCA-incorporated variant achieves a Top-1 accuracy of 96.84%/82.85% on CIFAR-10/100, surpassing the vanilla network (95.75%/81.33%) by 1.09%/1.52%. In sum, these experimental results reflect that the effectiveness of our MCA is not confined to some specific networks.

4.4. Image classification on ImageNet-1K

In this section, we further explore the practical benefits of the proposed MCA module on large-scale ImageNet-1K classification. Throughout the experiments, we follow the specifications elaborated in Section 3.2 and the protocol specified in Section 4.1 to incorporate MCA into ResNet with 18, 34, and 50 layers and compare it with other competing attention methods. The evaluation metrics include both efficiency (i.e., Parameters and FLOPs) and effectiveness (i.e., Top-1/Top-5 accuracy). All experimental results are shown in Table 10. It can be found that our MCA module can consistently improve the Top-1 and Top-5 Acc by a large margin over all employed backbone networks, with negligible increases in additional overhead. More precisely, MCA-ResNets can improve 1.81%/1.31%, 1.54%/0.8%, and 1.94%/1.2% over the corresponding baselines in terms of Top-1/5 Acc, respectively, with almost the same number of parameters and FLOPs. Meanwhile, our MCA also performs favorably against other competitors under all backbones while having minimal or competitive overhead. For example, in ResNet-18, MCA achieves 0.48%/0.4% gains in Top-1/5 Acc compared to the most competitive TA while having less overhead. A similar performance trend can be seen in the larger ResNet-34, where MCA provides 0.6%/0.28% improvements in Top-1/5 Acc. For ResNet-50, MCA outperforms the strongest competitor SA by 0.35% and 0.26% in Top-1 and Top-5 Acc, respectively, while having slightly cheaper overhead. These empirical results again demonstrate the lightweight and effectiveness of our method and further show that these properties are not constrained to CIFAR-10/100. Additionally, we present the training and validation curves of the backbone networks and their respective equivalents MCA-Nets in Fig. B.8. See Appendix B for further details. It is worth noting that compared with the experimental results presented in Section 4.3.2, it can be attested that MCA is robust and generalizable to different scale datasets in accuracy.

4.5. Visualization analysis with Grad-CAM++

In this section, we provide visualization results of several random samples from the ImageNet-1K validation set by utilizing the Grad-CAM++ technique to more intuitively reveal the superiority of our method and provide more insights into its practical benefits. Grad-CAM++ is a popular visualization method that visualizes the gradients of the top-class prediction with respect to the input sample as a colored overlay and illustrates how the networks make decisions in classification in the form of a heat map. By observing the results of Grad-CAM++, it can be seen that the networks concentrate on "what" and "where". Throughout the experiments, we employ ResNet-50 as the baseline, both comparing MCA with the aforementioned attention methods (CBAM and TA) and observing the visualization results of the feature maps before and after the MCA module in the last building block. All results are illustrated in Fig. 6. Note that the hotter colored regions in the visualization results indicate higher responsiveness to the network, i.e., regions with more significant influence on the top-class prediction.

As illustrated in Fig. 6, it is evident that our approach is more conducive to refining meaningful features and more helpful in precisely locating the relevant regions of the objects of interest than other compared methods. Specifically, in columns 2, 6, and 7 in Fig. 6, other methods either fail at predicting correctly or can only generate correct labels with lower accuracy, while our approach can make correct predictions with 90.16%, 98.0%, and 75.28% accuracy, respectively, and can consistently better focus on discriminative object regions. The most interesting visualization is shown in column 1, where the baseline and the networks embedded with CBAM or TA predict the wrong "swing" label with 74.27%, 53.34%, and 24.35% probabilities, respectively, while the MCA-integrated network can predict the correct "crutch" label with 99.57% accuracy. We argue that the reason for this

Table 9
Performance comparisons for various CNNs with and without our MCA module on the CIFAR-10/100 test sets. All results are the average of 5 runs. Best records are marked in bold.

Architectures	CIFAR-10			CIFAR-100		
	Parameters (M)	FLOPs (G)	Top-1 Acc (%)	Parameters (M)	FLOPs (G)	Top-1 Acc (%)
PreActResNet-164	1.70	**0.257**	94.77	1.73	**0.257**	77.03
PreActResNet-164 + MCA	**1.70**	0.263	**95.63**	**1.73**	0.263	**78.37**
MobileNet-v2	2.24	0.093	92.44	2.35	0.093	72.08
MobileNet-v2 + MCA	**2.24**	**0.093**	**93.28**	**2.35**	**0.093**	**73.31**
WideResNet16-8	10.96	1.55	95.73	11.01	1.55	79.29
WideResNet16-8 + MCA	**10.96**	**1.55**	**96.68**	**11.01**	**1.55**	**79.87**
WideResNet28-10	36.48	5.25	96.18	36.54	5.25	80.79
WideResNet28-10 + MCA	**36.48**	**5.25**	**97.04**	**36.54**	**5.25**	**81.75**
ResNeXt-29 ($8 \times 64d$)	34.43	**5.41**	95.75	34.52	**5.41**	81.33
ResNeXt-29 ($8 \times 64d$) + MCA	**34.43**	5.42	**96.84**	**34.52**	5.42	**82.85**

Table 10
Comparisons of efficiency (i.e., Parameters and FLOPs) and effectiveness (i.e., Top-1/Top-5 Acc) of different attention methods on ImageNet-1K validation set when taking ResNet with 18, 34, and 50 layers as backbones. All results are the average of 5 runs. Best records are marked in bold.

Methods	Backbones	ImageNet-1K			
		Parameters (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
ResNet-18	ResNet-18	**11.690**	**1.822**	69.76	89.08
+ SE		11.779	1.823	70.59	89.78
+ CBAM		11.779	1.823	70.73	89.91
+ ECA		11.690	1.823	70.88	89.90
+ TA		11.692	1.829	71.09	89.99
+ MCA (Ours)		**11.690**	1.823	**71.57**	**90.39**
ResNet-34	ResNet-34	**21.798**	**3.676**	73.31	91.42
+ SE		21.959	3.677	73.87	91.65
+ CBAM		21.960	3.677	74.01	91.76
+ ECA		21.798	3.677	74.19	91.80
+ TA		21.802	3.688	74.25	91.94
+ MCA (Ours)		**21.798**	3.677	**74.85**	**92.22**
ResNet-50	ResNet-50	**25.557**	**4.122**	76.13	92.86
+ SE		28.088	4.130	76.71	93.38
+ CBAM		28.090	4.128	77.34	93.69
+ ECA		25.557	4.128	77.40	93.61
+ TA		25.562	4.169	77.48	93.68
+ SRM		25.617	4.122	77.13	93.51
+ SGE		25.559	4.127	77.58	93.66
+ SA		25.558	4.130	77.72	93.80
+ MCA (Ours)		**25.557**	4.130	**78.07**	**94.06**

performance benefits from the fact that our MCA simultaneously models attention with complementary relationships in multiple dimensions. Additionally, the comparison of the visualizations in the last two rows in Fig. 6 demonstrates that our approach can capture tighter and more relevant regions for a particular object class.

Overall, these results reflect that our MCA helps networks to focus on richer and more discriminative feature regions, prompting them to locate objects better and make more accurate predictions.

5. Conclusion

This paper proposes a lightweight yet efficient, plug-and-play multidimensional collaborative attention module, MCA, with strong generalization ability and robustness for various network architectures and datasets. The proposed MCA consists of three parallel branches, accounting for simultaneously modeling attention in multiple dimensions. The core components of MCA include *squeeze transformation* and *excitation transformation*. For the *squeeze transformation*, we develop an adaptive combination mechanism to aggregate average-pooled and standard-deviation-pooled features, introducing input-specific dynamics and improving the discriminability of feature descriptors. Meanwhile, rather than utilizing a nonlinear strategy with dimensionality reduction to capture global feature interactions, we design a simple strategy in the *excitation transformation* that adaptively determines the coverage of interaction to capture local feature interactions, thus achieving a balance between performance and complexity. To evaluate the efficacy of MCA, we first conduct comprehensive ablation experiments on CIFAR to demonstrate the correctness and effectiveness of our design scheme. Then, we compare MCA with other SOTA counterparts by embedding them into various well-established CNNs on CIFAR and ImageNet-1K datasets, from which experimental results verify the effectiveness and efficiency of MCA. For instance, on the ImageNet-1K benchmark, for ResNet-50 with 25.557M parameters and 4.122 GFLOPs, our proposed plug-in, MCA, results in an increase of parameters by about 0.3K and GFLOPs by about 8e-3 respectively while achieving 1.94%/1.2% improvement in terms of Top-1/Top-5 accuracy. In the meantime, compared with the best-performing competitor SA, our MCA can provide 0.35% and 0.26% gains in Top-1 and Top-5 accuracy, respectively, while having slightly cheaper overhead. Similar performance improvements can also be observed across various datasets and baseline networks. Finally, the visualization results based on the Grad-CAM++ technique manifest that MCA can indeed induce networks to locate and recognize the objects of interest more accurately and further support the intrinsic philosophy of our method. In addition, the MCA module also sheds light on the limitations of previous competitors in inferring attention, which may prove helpful for other applications requiring strong informative features. In the future, we plan to investigate and develop attention methods that can capture

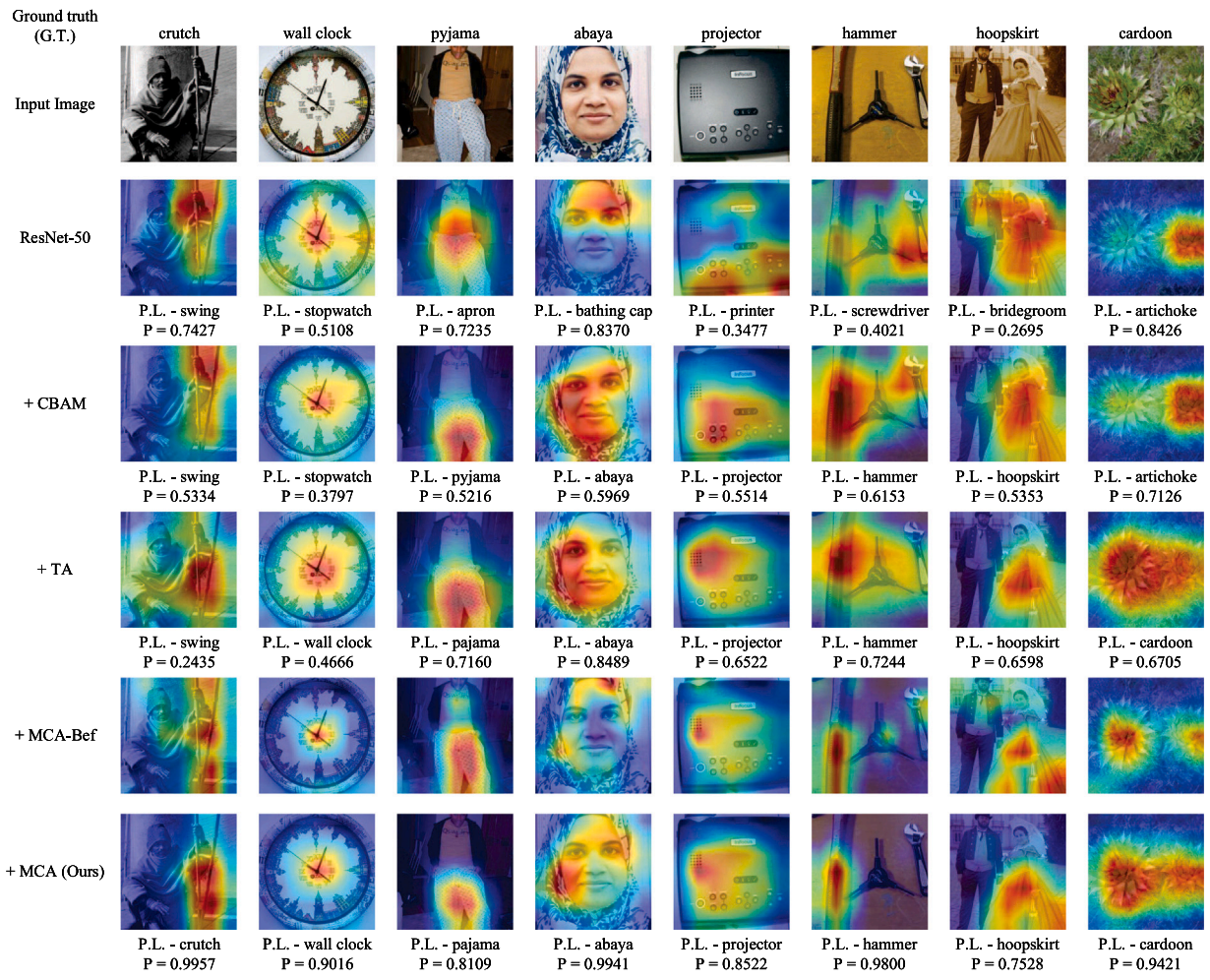


Fig. 6. Visualization of feature maps generated by networks with different attention methods in the last building block using Grad-CAM++ as a visualization tool. When taking ResNet-50 as the baseline, we both compare the MCA-integrated (+MCA) network with CBAM-integrated (+CBAM) or TA-integrated (+TA) networks and observe the visualization results of the feature maps before(+MCA-Bef) and after(+MCA) the MCA module. Ground truth (G.T.) labels for images are provided on the top of the original samples. Prediction labels (P.L.) and accuracy scores are provided below the corresponding visualizations.

multi-scale features at a more granular level, and we also expect that other novel and efficient methods of capturing multi-scale features when computing attention weights can improve upon our results while reducing complexity and computational overhead.

CRediT authorship contribution statement

Yang Yu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Yi Zhang:** Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Zeyu Cheng:** Formal analysis, Investigation, Data curation, Writing – review & editing. **Zhe Song:** Formal analysis, Investigation, Data curation, Writing – review & editing. **Chengkai Tang:** Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank editors for rigorous work and the anonymous reviewers for their comments and suggestions. This work was supported in part by National Natural Science Foundation of China under Grant 62171735, 62271397, 62173276, 62101458, 62001392, 61803310 and 61801394, in part by the Natural Science Basic Research Program of Shaanxi under Grant 2022GY-097, 2021JQ-122 and 2021JQ-693, in part by Shenzhen Science and Technology Innovation Program under Grant JCYJ20220530161615033 and in part by China Postdoctoral Science Foundation under Grant 2020M673482 and 2020M673485.

Appendix A. Comparisons of training and validation curves for networks on CIFAR-10/100

To gain insight into the impact of MCA on optimizing base models, Fig. A.7 depicts the training and validation curves of ResNets with MCA and other counterparts such as SRM, ECA, and TA on CIFAR-10 and CIFAR-100. It can be seen that MCA exhibits improved optimization properties and yields consistent gains in performance throughout the training process.

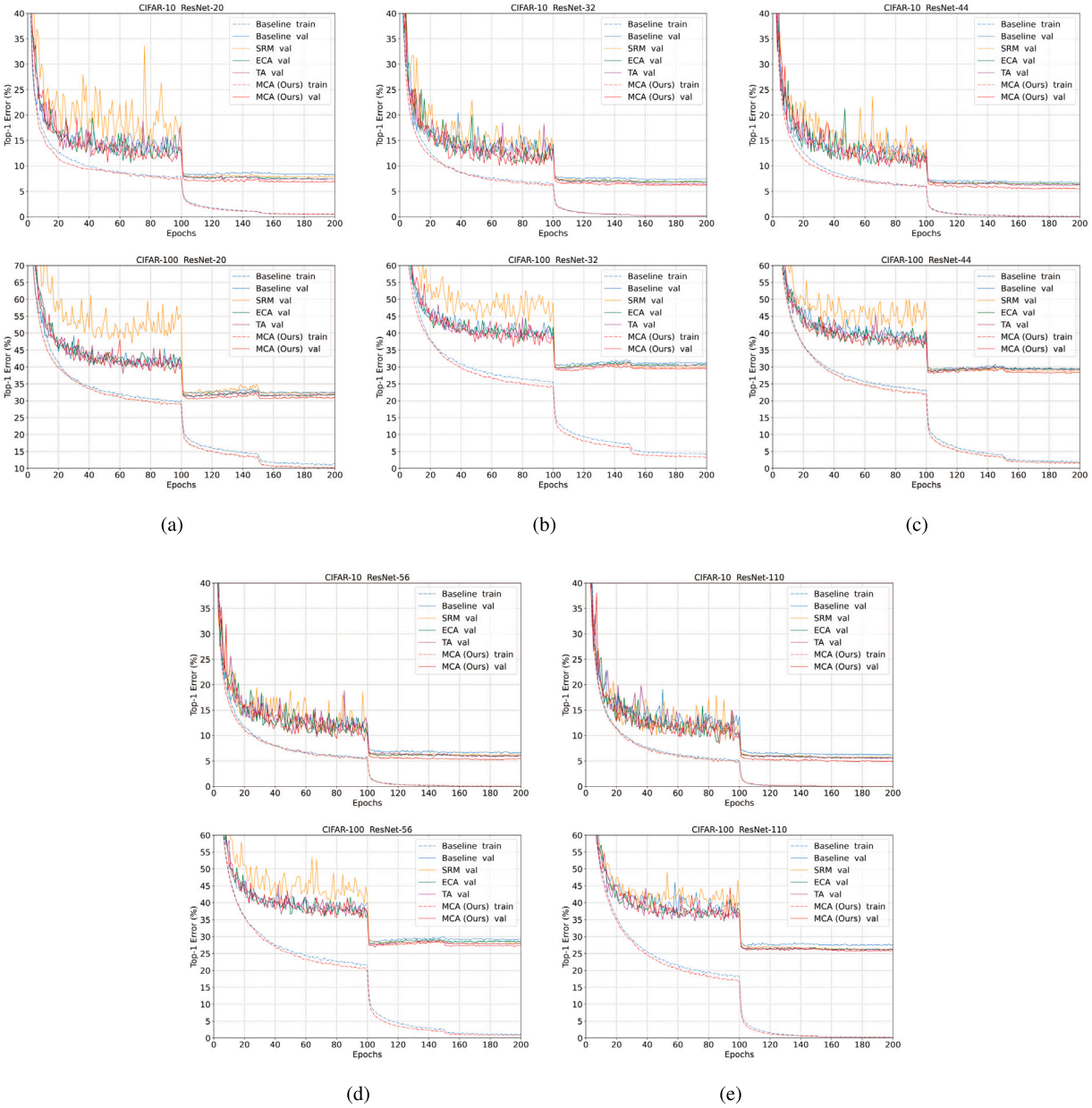
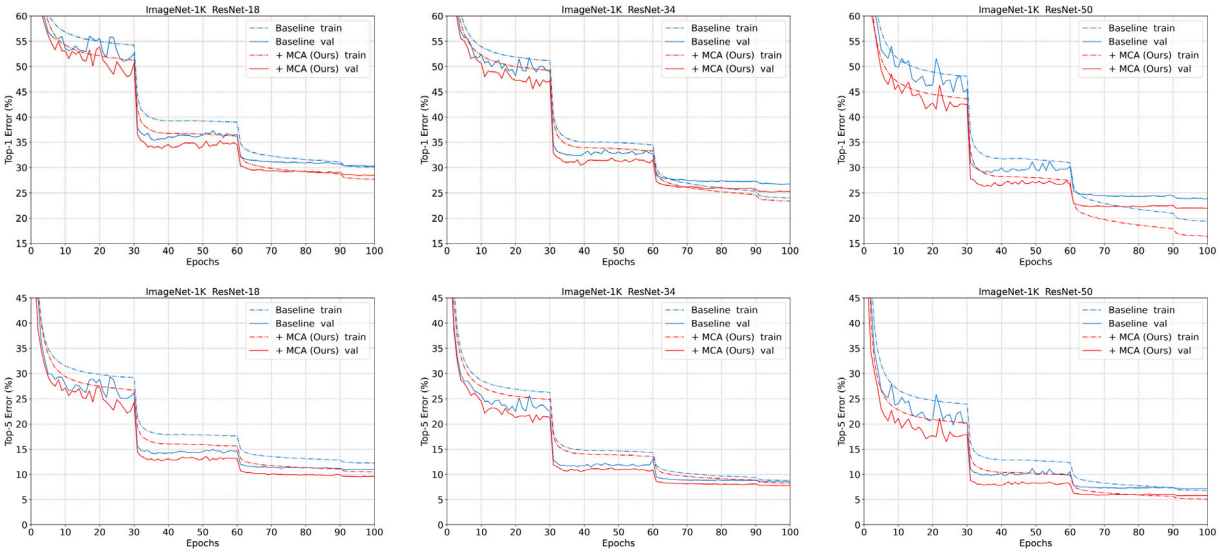


Fig. A.7. Comparisons of training and validation curves of our MCA and several attention methods (i.e., SRM, ECA, and TA) on CIFAR-10/100. Dash-dot lines denote training errors, and solid lines denote validation errors. Subfigures (a)–(e) refer to utilizing ResNet-20, ResNet-32, ResNet-44, ResNet-56, and ResNet-110 as baselines, respectively.



(a) ResNet-18 (b) ResNet-34 (c) ResNet-50

Fig. B.8. Comparisons of training and validation curves for ResNets with and without the MCA module on ImageNet-1K. Dash-dot lines denote training errors, and solid lines denote validation errors.

Appendix B. Comparisons of training and validation curves for networks on ImageNet-1K

To intuitively illustrate the advantages that MCA brings to the optimization of base networks, we present the training and validation curves of the backbone networks and their respective equivalents MCA-Nets in Fig. B.8. As can be seen, in terms of training and validation accuracies, networks equipped with MCA consistently perform better than their corresponding backbones throughout the optimization procedure, suggesting that MCA can facilitate optimization and improve performance across a range of different depth networks.

References

Anari, S., Tataei Sarshar, N., Mahjoori, N., Dorosti, S., Rezaie, A., 2022. Review of deep learning approaches for thyroid cancer diagnosis. Math. Probl. Eng. 2022, 1–8.

Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H., 2019. GCNet: Non-local networks meet squeeze-excitation networks and beyond. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop. ICCV, IEEE, pp. 1971–1980.

Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N., 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 839–847.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1251–1258.

de Santana Correia, A., Colombini, E.L., 2022. Attention, please! A survey of neural attention models in deep learning. Artif. Intell. Rev. 55 (8), 6037–6124.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., Torr, P., 2019a. Res2net: A new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell. 43 (2), 652–662.

Gao, Z., Xie, J., Wang, Q., Li, P., 2019b. Global second-order pooling convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3024–3033.

Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R.R., Cheng, M.-M., Hu, S.-M., 2022. Attention mechanisms in computer vision: A survey. Comput. Vis. Media 8 (3), 331–368.

Han, D., Kim, J., Kim, J., 2017. Deep pyramidal residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5927–5935.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer, pp. 630–645.

Hou, Q., Zhou, D., Feng, J., 2021. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13713–13722.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A., 2018. Gather-excite: exploiting feature context in convolutional neural networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 9423–9433.

Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E., 2020. Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. 42 (08), 2011–2023.

Huang, G., Liu, S., Van der Maaten, L., Weinberger, K.Q., 2018. Condensenet: An efficient densenet using learned group convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2752–2761.

Kirzhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Krizhevsky, A., Hinton, G., et al., 2009. Learning Multiple Layers of Features from Tiny Images. Toronto, ON, Canada.

Lee, H., Kim, H.-E., Nam, H., 2019. Srm: A style-based recalibration module for convolutional neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1854–1862.

Li, G., Fang, Q., Zha, L., Gao, X., Zheng, N., 2022. HAM: Hybrid attention module in deep convolutional neural networks for image classification. Pattern Recognit. 129, 108785.

Li, X., Hu, X., Yang, J., 2019. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. arXiv preprint arXiv:1905.09646.

Liu, J.-J., Hou, Q., Cheng, M.-M., Wang, C., Feng, J., 2020. Improving convolutional networks with self-calibrated convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10096–10105.

Ma, N., Zhang, X., Zheng, H.-T., Sun, J., 2018. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 116–131.

Misra, D., Nalamada, T., Arasanipalai, A.U., Hou, Q., 2021. Rotate to attend: Convolutional triplet attention module. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3139–3148.

Park, J., Woo, S., Lee, J.-Y., Kweon, I.-S., 2018. BAM: Bottleneck attention module. In: British Machine Vision Conference. BMVC, British Machine Vision Association (BMVA).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: an imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. pp. 8026–8037.

Qin, Z., Zhang, P., Wu, F., Li, X., 2021. Fcanet: Frequency channel attention networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 783–792.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J., 2019. Stand-alone self-attention in vision models. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. pp. 68–80.

Ranjbarzadeh, R., Bagherian Kasgari, A., Jafarzadeh Ghoushchi, S., Anari, S., Naseri, M., Bendechache, M., 2021. Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images. Sci. Rep. 11 (1), 10930.

Ranjbarzadeh, R., Jafarzadeh Ghoushchi, S., Anari, S., Safavi, S., Tataei Sarshar, N., Babaee Tirkolaee, E., Bendechache, M., 2022. A deep learning approach for robust, multi-oriented, and curved text detection. Cogn. Comput. 1–13.

Ranjbarzadeh, R., Jafarzadeh Ghoushchi, S., Tataei Sarshar, N., Tirkolaee, E.B., Ali, S.S., Kumar, T., Bendechache, M., 2023. ME-CCNN: Multi-encoded images and a cascade convolutional neural network for breast tumor segmentation and recognition. Artif. Intell. Rev. 1–38.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520.

Shojaiee, F., Baleghi, Y., 2023. EFASPP U-net for semantic segmentation of night traffic scenes using fusion of visible and thermal images. Eng. Appl. Artif. Intell. 117, 105627.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Su, Z., Fang, L., Kang, W., Hu, D., Pietikäinen, M., Liu, L., 2020. Dynamic group convolution for accelerating convolutional neural networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. Springer, pp. 138–155.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826.

Tataei Sarshar, N., Ranjbarzadeh, R., Jafarzadeh Ghoushchi, S., de Oliveira, G.G., Anari, S., Parhizkar, M., Bendechache, M., 2021. Glioma brain tumor segmentation in four MRI modalities using a convolutional neural network and based on a transfer learning method. In: Brazilian Technology Symposium. Springer, pp. 386–402.

Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803.

Wang, S.-y., Qu, Z., Li, C.-j., Gao, L.-y., 2023. Banet: Small and multi-object detection with a bidirectional attention network for traffic scenes. Eng. Appl. Artif. Intell. 117, 105504.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q., 2020. ECA-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11534–11542.

Woo, S., Park, J., Lee, J.-Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1492–1500.

Yang, L., Zhang, R.-Y., Li, L., Xie, X., 2021. Simam: A simple, parameter-free attention module for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 11863–11874.

Yang, Z., Zhu, L., Wu, Y., Yang, Y., 2020. Gated channel transformation for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11794–11803.

Yu, Y., Zhang, Y., Song, Z., Tang, C.-K., 2022. LMA: lightweight mixed-domain attention for efficient network design. Appl. Intell. 1–20.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. In: British Machine Vision Conference 2016. British Machine Vision Association.

Zhang, Q., Xu, Y., Zhang, J., Tao, D., 2023. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. Int. J. Comput. Vis. 1–22.

Zhang, Q.-L., Yang, Y.-B., 2021. Sa-net: Shuffle attention for deep convolutional neural networks. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 2235–2239.

Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6848–6856.

Zhou, D., Hou, Q., Chen, Y., Feng, J., Yan, S., 2020. Rethinking bottleneck structure for efficient mobile network design. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. Springer, pp. 680–697.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2921–2929.