Strengthening Layer Interaction via Dynamic Layer Attention

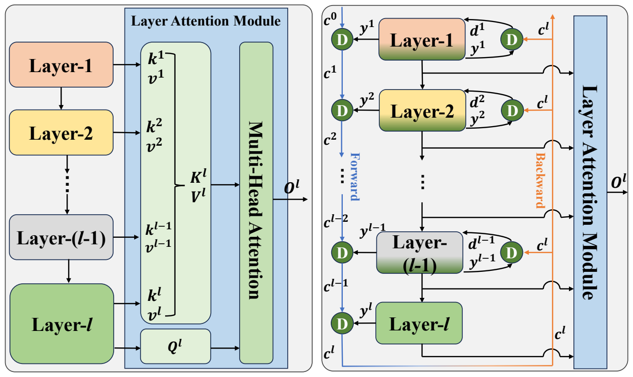
Kaishen Wang[1] , **Xun Xia** [2] , **Jian Liu**[2] , **Zhang Yi**[1] , **Tao He**[1,*]
[1]College of Computer Science, Sichuan University
[2]Clinical Medical College and The First Affiliated Hospital of Chengdu Medical College
wangks@stu.scu.edu.cn, xiaxun@cmc.edu.cn, liujiansh@126.com, {zhangyi, tao_he}@scu.edu.cn

Abstract

In recent years, employing layer attention to enhance interaction among hierarchical layers has proven to be a significant advancement in building network structures. In this paper, we delve into the distinction between layer attention and the general attention mechanism, noting that existing layer attention methods achieve layer interaction on fixed feature maps in a static manner. These static layer attention methods limit the ability for context feature extraction among layers. To restore the dynamic context representation capability of the attention mechanism, we propose a Dynamic Layer Attention (DLA) architecture. The DLA comprises dual paths, where the forward path utilizes an improved recurrent neural network block, named Dynamic Sharing Unit (DSU), for context feature extraction. The backward path updates features using these shared context representations. Finally, the attention mechanism is applied to these dynamically refreshed feature maps among layers. Experimental results demonstrate the effectiveness of the proposed DLA architecture, outperforming other state-of-the-art methods in image recognition and object detection tasks. Additionally, the DSU block has been evaluated as an efficient plugin in the proposed DLA architecture. The code is available at https://github.com/tunantu/Dynamic-Layer-Attention.

1 Introduction

Numerous studies have highlighted the significance of enhancing interaction among hierarchical layers in Deep Convolutional Neural Networks (DCNNs), which have made substantial progress across various tasks. For instance, ResNet [He *et al.*, 2016] introduced a straightforward and highly effective implementation by incorporating skip connections between two consecutive layers. DenseNet [Huang *et al.*, 2017] further improved inter-layer interaction by recycling information from all previous layers. Meanwhile, attention mechanisms are playing an increasingly crucial role in DCNNs. The



(a) Static Layer Attention (b) Dynamic Layer Attention

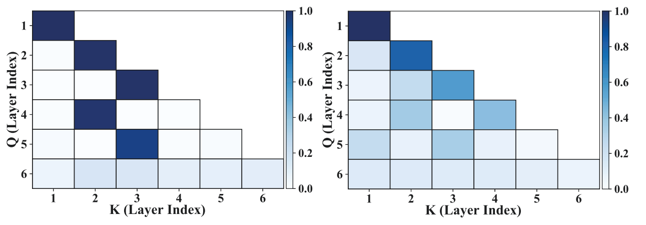
Figure 1: The comparison between **static** and **dynamic** layer attention architecture.

evolution of attention mechanisms in DCNNs has progressed through several stages, including channel attention ([Hu *et al.*, 2018], [Wang *et al.*, 2020a]), spatial attention ([Carion *et al.*, 2020], [Wang *et al.*, 2018]), branch attention ([Srivastava *et al.*, 2015], [Li *et al.*, 2019]), and temporal attention ([Xu *et al.*, 2017], [Chen *et al.*, 2018a]).

Recently, attention mechanisms have been successfully applied to another direction, (e.g., DIANet [Huang *et al.*, 2020], RLANet [Zhao *et al.*, 2021], MRLA [Fang *et al.*, 2023]), indicating the feasibility of strengthening interaction among layers through attention mechanisms. Compared to simple interaction like those in ResNet and DenseNet, the introduction of attention mechanisms makes the interaction among layers more closely effective. DIANet employed a parameter-shared LSTM module along the network's depth to facilitate interaction among layers. RLANet proposed a layer aggregation structure to reuse features of previous layers for enhancing layer interaction. MRLA first introduced the concept of layer attention, treating each feature as a token to learn useful information from others by attention mechanisms.

However, we have identified a common drawback in existing layer attention mechanisms: they are applied in a **static** manner, limiting inter-layer information interaction. In channel and spatial attention, for the input $\boldsymbol{x} \in \mathbb{R}^{C\times H\times W}$, tokens are input to the attention module, all of which are generated from $\boldsymbol{x}$ concurrently. However, in existing layer attention,

[*]Corresponding author.



(a) Static Layer Attention (b) Dynamic Layer Attention

Figure 2: Visualization of attention scores from stage 3 of the ResNet-56 between **static** and the proposed **dynamic** layer attention on CIFAR-100 dataset.

features generated from different time steps are treated as tokens and passed into the attention module, as shown in Figure 1(a). Since tokens generated earlier do not change once produced, the input tokens are relatively static, leading to a reduction in information interaction between the current layer and previous layers. Figure 2(a) visualizes the MRLA attention scores from stage 3 of ResNet-56 trained on CIFAR-100. When the first 5 layers reuse information from previous layers through static layer attention, the key values from only one specific layer are activated, with almost zero attention assigned to other layers. This observation verifies that static layer attention compromises the efficiency of information interaction among layers.

To solve the static problem of layer attention, we propose a novel **Dynamic Layer Attention** (DLA) architecture to improve information flow among layers, where the information of previous layers can be dynamically modified during the feature interaction. As shown in Figure 2(b), during the reutilization of information from preceding layers, the attention of the current feature undergo a shift from exclusively focusing on a particular layer to incorporating information from various layers. DLA facilitates a more thorough exploitation of information, enhancing inter-layer information interaction efficiency. Experimental results demonstrate the effectiveness of the proposed DLA architecture, outperforming other state-of-the-art methods in image recognition and object detection tasks. The contributions of this paper are summarized as follows:

- We propose a novel DLA architecture, which consists of dual paths, where the forward path extracts context feature among layers using a Recurrent Neural Network (RNN) and the backward path refreshes the original feature at each layer using these shared context representation.
- A novel RNN block, named Dynamic Sharing Unit (DSU), is proposed to be a suitable component for DLA. It effectively promotes the dynamic modification of information within DLA and demonstrates commendable performance in the layer-wise information integration as well.

2 Related Work

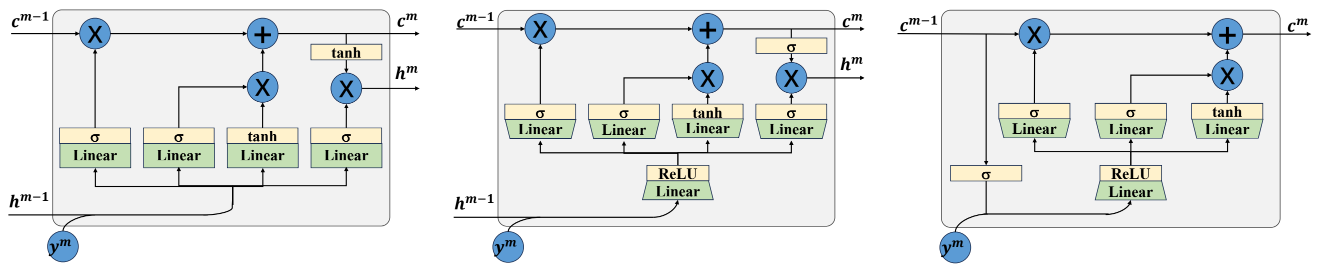
Layer Interaction. Layer interaction has consistently been an intriguing aspect of DCNNs. Initially, the implementation of layer interaction was relatively simple. ResNet [He *et al.*, 2016] introduced skip connections between consecutive layers, mitigating issues of gradient vanishing and exploding to some extent. DenseNet [Huang *et al.*, 2017] further enhanced layer interaction by reusing information generated from all preceding layers. In U-Net [Ronneberger *et al.*, 2015], a commonly utilized architecture in medical segmentation, the encoder and decoder are connected through skip connections to improve feature extraction and achieve higher accuracy.

Recent studies have explored effective methods for implementing layer interaction. DREAL [Li and Chen, 2020] optimized parameters by introducing arbitrary attention modules and employed a Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] to incorporate previous attention weights. The update of parameters for both LSTM and attention layers was achieved through deep reinforcement learning. DIANet [Huang *et al.*, 2020] incorporated a LSTM module at the layer level, constructing a DIA block shared by all layers in the entire network to facilitate inter-layer interaction. RealFormer [He *et al.*, 2020] integrated residual connections between adjacent attention modules, adding the attention scores from the previous layer to the current one. RLANet [Zhao *et al.*, 2021] introduced a lightweight recurrent layer aggregation module to describe how information from previous layers can be efficiently reused for better feature extraction in the current layer. [Zhao *et al.*, 2022] proposed a straightforward and versatile approach to strike a balance between effectively utilizing neural network information and maintaining high computational efficiency. By seamlessly integrating features from preceding layers, it foster effective interaction of information. MRLA [Fang *et al.*, 2023] treated the features of each layer as tokens, enabling interaction among different hierarchical layers through an attention mechanism, further strengthening the interaction among layers.

Dynamic Network Architecture. Dynamic networks represent a type of neural network structure whose topology or parameters can dynamically change during runtime, providing the network with considerable flexibility or other benefits. [Wang *et al.*, 2020b] proposed a universally adaptable inference framework for the majority of DCNNs, with costs that can be easily dynamically adjusted without additional training. RANet [Yang *et al.*, 2020] introduced an adaptive network that could effectively reduce the spatial redundancy involved in inferring high-resolution inputs. [Han *et al.*, 2021] have shown that dynamic neural networks can enhance the efficiency of deep networks. CondenseNetv2 [Yang *et al.*, 2021] utilized a Sparse Feature Reactivation (SFR) module to reactivate pruned features from CondenseNet [Huang *et al.*, 2018], thereby enhancing feature utilization efficiency. To the best of our knowledge, this paper is the first attempt to build a dynamic network architecture for strengthening layer interaction.

3 Dynamic Layer Attention

We will start by reconsidering the current layer attention architecture and elucidating its static nature. Subsequently, we will introduce the Dynamic Layer Attention (DLA). Finally,



(a) LSTM (b) DIA (c) DSU

Figure 3: A comparison of LSTM, DIA, and the proposed DSU blocks.

we will present an enhanced RNN plugin block named the Dynamic Sharing Unit (DSU), integrated within the DLA architecture.

3.1 Rethinking Layer Attention

Layer attention was recently defined by [Fang *et al.*, 2023] and is illustrated in Figure 1(a), where the attention mechanism enhances layer interaction. [Fang *et al.*, 2023] focused on reducing the computational cost of layer attention and proposed the Recurrent Layer Attention (RLA) architecture. In RLA, features from distinct layers are treated as tokens and undergo computations, ultimately producing attention output. Let the feature output of the l-th layer be $\boldsymbol{x}^l \in \mathbb{R}^{C\times W\times H}$. The vectors $\boldsymbol{Q}^l$, $\boldsymbol{K}^l$, and $\boldsymbol{V}^l$ can be calculated as follows:

$$\begin{cases} \boldsymbol{Q}^l = f_q^l(\boldsymbol{x}^l) \\ \boldsymbol{K}^l = \text{Concat}\left[f_k^1(\boldsymbol{x}^1), \dots, f_k^l(\boldsymbol{x}^l)\right] \\ \boldsymbol{V}^l = \text{Concat}\left[f_v^1(\boldsymbol{x}^1), \dots, f_v^l(\boldsymbol{x}^l)\right], \end{cases} \tag{1}$$

where f_q is a mapping function to extract information from the l-th layer, while f_k and f_v are corresponding mapping functions intended to extract information from the 1st to l-th layers, respectively. The attention output $\boldsymbol{o}^l$ is given as follows:

$$\begin{aligned} \boldsymbol{o}^l &= \text{Softmax}\left(\frac{\boldsymbol{Q}^l(\boldsymbol{K}^l)^{\mathrm{T}}}{\sqrt{D_k}}\right)\boldsymbol{V}^l \\ &= \sum_{i=1}^{l} \text{Softmax}\left(\frac{\boldsymbol{Q}^l\left[f_k^i(\boldsymbol{x}^i)\right]^{\mathrm{T}}}{\sqrt{D_k}}\right) f_v^i(\boldsymbol{x}^i), \end{aligned} \tag{2}$$

where D_k serves as the scaling factor. A lightweight version of RLA was proposed to recurrently update the attention output $\boldsymbol{o}^l$ as follows:

$$\boldsymbol{o}^l = \boldsymbol{\lambda}_o^l \odot \boldsymbol{o}^{l-1} + \text{Softmax}\left(\frac{\boldsymbol{Q}^l\left[f_k^l(\boldsymbol{x}^l)\right]^{\mathrm{T}}}{\sqrt{D_k}}\right) f_v^l(\boldsymbol{x}^l), \tag{3}$$

where $\boldsymbol{\lambda}_o^l$ is a learnable vector and $\odot$ indicates the element-wise multiplication. With the multi-head structure design, Multi-head RLA (MRLA) is introduced.

3.2 Motivation

MRLA successfully integrated the attention mechanism to enhance layer interaction, effectively addressing computational costs. However, when MRLA is applied in the l-th layer, a preceding feature output $\boldsymbol{x}^m$ ($m < l$) has already been generated in the m-th layer, with no subsequent changes. Consequently, the information processed by MRLA comprises fixed features from the previous layers. In contrast, widely used attention-based models, such as channel attention [Hu *et al.*, 2018; Wang *et al.*, 2020a], spatial attention [Carion *et al.*, 2020; Huang *et al.*, 2019], and Transformers [Vaswani *et al.*, 2017; Liu *et al.*, 2021; Chen *et al.*, 2023; Jiao *et al.*, 2023], pass tokens into the attention module generated simultaneously. Applying the attention module between freshly generated tokens ensures that each token consistently learns up-to-date features. Therefore, we categorize MRLA as a **static** layer attention mechanism, limiting interaction between the current layer and shallower layers.

In a general self-attention mechanism, the feature $\boldsymbol{x}^m$ serves two purposes: conveying essential information and representing context. The essential information extracted at the current layer distinguishes it from that at other layers. Meanwhile, the context representation captures changes and evolution of features along the temporal axis, a critical aspect in determining feature freshness. In the general attention mechanism, essential information is generated at each layer, and the context representation is transferred to the next layer for calculating the attention output. In contrast, in layer attention, once tokens are generated, attention is calculated with a fixed context representation, diminishing the efficiency of the attention mechanism. Therefore, this paper aims to establish a novel method to restore the context representation, ensuring that the information fed into the layer attention is consistently dynamic.

3.3 Dynamic Layer Attention Architecture

To address the static issue of MRLA, we propose the use of a dynamic updating rule to extract the context representation and promptly update features at previous layers, resulting in a Dynamic Layer Attention (DLA) architecture. As illustrated in Figure 1(b), DLA consists of dual paths: forward and backward. In the forward path, a Recurrent Neural Network (RNN) is employed for context feature extraction. Let the RNN block be denoted as "Dyn", and the initial context as $\boldsymbol{c}^0$, respectively. $\boldsymbol{c}^0$ is randomly initialized. Given an input $\boldsymbol{x}^m \in \mathbb{R}^{C\times W\times H}$ where $m < l$, a Global Average Pooling (GAP) is applied to extract global features at m-th layer as follows:

$$\boldsymbol{y}^m = \text{GAP}(\boldsymbol{x}^m),\ \boldsymbol{y}^m \in \mathbb{R}^C. \tag{4}$$

The context representation is extracted as follows:

$$\boldsymbol{c}^m = \text{Dyn}(\boldsymbol{y}^m, \boldsymbol{c}^{m-1}; \theta^l). \tag{5}$$

where θ^l represents the shared trainable parameters of "Dyn". Once the context $\boldsymbol{c}^l$ is calculated, the features of each layer are simultaneously updated in the backward path as follows:

$$\begin{cases} \boldsymbol{d}^m = \text{Dyn}(\boldsymbol{y}^m, \boldsymbol{c}^l; \theta^l) \\ \boldsymbol{x}^m \leftarrow \boldsymbol{x}^m \odot \boldsymbol{d}^m \end{cases} \tag{6}$$

Referring to Equation (5), the forward context feature extraction is a step-by-step process with a computation complexity of $\mathcal{O}(n)$. Meanwhile, the feature updating in Equation (6) can be performed in parallel, resulting in a computation complexity of $\mathcal{O}(1)$. After updating $\boldsymbol{x}^m$, the basic version of DLA use Equation (2) to compute the layer attention, abbreviated as DLA-B. For the lightweight version of DLA, Simply update $\boldsymbol{o}^{l-1}$ and then use Equation (3) to obtain DLA-L.

Computation Efficiency. DLA possesses several advantages in its structural design. Firstly, global information is condensed to compute context information, a utility that has been validated in [Huang *et al.*, 2020]. Secondly, DLA employs shared parameters within a RNN block. Thirdly, the context $\boldsymbol{c}^l$ is separately fed into the feature maps at each layer in parallel. Both the forward and backward paths share the same parameters throughout the entire network. Finally, we introduce an efficient RNN block for calculating context representation, which will be elucidated in the following subsection. With these efficiently designed structural rules, the computation cost and network capacity are guaranteed.

3.4 Dynamic Sharing Unit

LSTM, as depicted in Figure 2(a), is designed for processing sequential data and learning temporal features, enabling it to capture and store information over long sequences. However, the fully connected linear transformation in LSTM significantly increases the network capacity when embedding LSTM as the recurrent block in DLA. To mitigate this capacity increase, a variant LSTM block named the DIA unit was proposed by [Huang *et al.*, 2020], as illustrated in Figure 2(b). Before feeding data into the network, DIA first utilizes a linear transformation followed by a ReLU activation function to reduce the input dimension. Additionally, DIA replaces the Tanh function with a Sigmoid function at the output layer.

LSTM and DIA generate two outputs, comprising a hidden vector $\boldsymbol{h}^m$ and a cell state vector $\boldsymbol{c}^m$. Typically, $\boldsymbol{h}^m$ is used as the output vector, and $\boldsymbol{c}^m$ serves as the memory vector. DLA is exclusively focused on extracting context characteristics from different layers, where the RNN block has no duty to transform its internal state feature to the outside. Consequently, we discard the output gate and merge the memory and hidden vector by omitting the $\boldsymbol{h}^m$ symbol. The proposed simplified RNN block is named Dynamic Sharing Unit (DSU). The workflow of the DSU is illustrated in Figure 2(c). Specifically, before adding $\boldsymbol{c}^{m-1}$ and $\boldsymbol{y}^m$, we first normalize $\boldsymbol{c}^{m-1}$ using an activation function $\sigma(\cdot)$. Here, we opt for the Sigmoid function ($\sigma(z) = 1/(1+e^{-z})$). Therefore, the input to DSU was compressed as follows:

$$\boldsymbol{s}^m = \text{ReLU}\left(\boldsymbol{W}_1 \left[\sigma(\boldsymbol{c}^{m-1}), \boldsymbol{y}^m\right]\right). \tag{7}$$

The hidden transformation, the input gate, and the forget gate can be represented by the following formula:

$$\begin{cases} \tilde{\boldsymbol{c}}^m = \text{Tanh}(\boldsymbol{W}_2^c \cdot \boldsymbol{s}^m + b^c) \\ \boldsymbol{i}^m = \sigma(\boldsymbol{W}_2^i \cdot \boldsymbol{s}^m + b^i) \\ \boldsymbol{f}^m = \sigma(\boldsymbol{W}_2^f \cdot \boldsymbol{s}^m + b^f) \end{cases} \tag{8}$$

Subsequently, we obtain

$$\boldsymbol{c}^m = \boldsymbol{f}^m \odot \boldsymbol{c}^{m-1} + \boldsymbol{i}^m \odot \tilde{\boldsymbol{c}}^m \tag{9}$$

To decrease the network parameters, let $\boldsymbol{W}_1 \in \mathbb{R}^{\frac{C}{r} \times 2C}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$, where r is the reduction ratio. DSU reduces the parameters to $5C^2/r$, which is fewer than $8C^2$ of LSTM and $10C^2/r$ of DIA.

4 Experiments

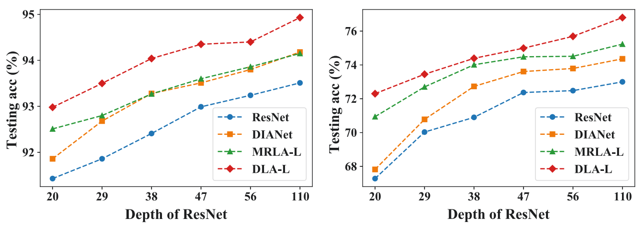
4.1 Image Classification

Experimental Setup. We conducted experiments on the CIFAR-10, CIFAR-100, and ImageNet-1K datasets using ResNets [He *et al.*, 2016] as the backbone network for image classification. For the CIFAR-10 and CIFAR-100 datasets, we employed standard data augmentation strategies [Huang *et al.*, 2016]. The training process involved random horizontal flipping of images, padding each side by 4 pixels, and then randomly cropping to 32×32. Normalization with mean and standard deviation adjustment was implemented, and training hyperparameters such as batch size, initial learning rate, and weight decay followed the recommendations of the original ResNets [He *et al.*, 2016]. To address hyperparameter uncertainty, we conducted five runs of experiments. For the ImageNet-1K dataset, we adopted the same data augmentation strategy and hyperparameter settings outlined in [He *et al.*, 2016] and [He *et al.*, 2019]. During training, images were randomly cropped to 224×224 with horizontal flipping. In the testing phase, images were resized to 256×256, then centrally cropped to a final size of 224×224. The optimization process utilized an SGD optimizer with a momentum of 0.9 and weight decay of 1e-4. The initial learning rate was set to 0.1 and decreased according to the MultiStepLR schedule over 100 epochs for a batch size of 256, consistent with the ResNet approach [He *et al.*, 2016]. Meanwhile, the reduction ratio r was set to 4 for the CIFAR-10 and CIFAR-100 datasets, and 20 for the ImageNet-1K dataset, consistent with the settings in DIANet [Huang *et al.*, 2020].

Results on CIFAR. The experimental results, showcasing Accuracy±Std, are presented in Table 1. These results underscore the significant superiority of our DLA-B and DLA-L models over other challenging networks, including SE [Hu *et al.*, 2018], ECA [Wang *et al.*, 2020a], DIANet [Huang *et al.*, 2020], MRLA [Fang *et al.*, 2023] on the CIFAR-10 and CIFAR-100 datasets. In comparison with the baselines, DLA-L's top-1 accuracy on CIFAR-10 surpasses ResNets by 1.32%, 1.60%, and 1.62%, and even outperforms them on CIFAR-100 by 4.96%, 2.94%, and 3.41%. Furthermore, both DLA-B and DLA-L outperform MRLA-B and MRLA-L, which are typical static layer attention models. It is noteworthy that DLA-L-20 (embedding ResNet-20) achieves fewer parameters than ResNet-56 while maintaining comparable top-1 accuracy on the CIFAR-10 (92.96% vs. 92.95%) and CIFAR-100 (72.26% vs. 72.32%) datasets. Furthermore, DLA-L-56 outperforms ResNet-110 by 1.51% and 2.46% on CIFAR-10 and CIFAR-100 datasets, respectively. As depicted in Figure 4, ResNet, DIANet, and MRLA-L exhibit a relatively slow growth in capabilities with increasing network depth. In contrast, the proposed DLA demonstrates a faster increase in test accuracy on the CIFAR-10 and CIFAR-100 datasets as the network depth increases. This observation verifies that strengthening layer interaction through DLA is more beneficial in a deeper network structure.

Data / Model	CIFAR-10		CIFAR-100	
	#P(M)	**Top-1**	**#P(M)**	**Top-1**
ResNet-20	0.22	91.64±0.18	0.24	67.30±0.28
SE	0.24	91.29±0.24	0.27	68.93±0.35
ECA	0.22	91.63±0.16	0.24	67.23±0.24
DIANet	0.44	91.43±0.14	0.46	67.67±0.22
MRLA-B	0.23	92.15±0.23	0.25	71.44±0.49
DLA-B (ours)	0.41	**92.47±0.10**	0.43	**72.01±0.37**
MRLA-L	0.23	92.65±0.08	0.25	71.46±0.27
DLA-L (ours)	0.41	**92.96±0.18**	0.43	**72.26±0.29**
ResNet-56	0.59	92.95±0.18	0.61	72.32±0.36
SE	0.66	93.60±0.18	0.68	73.51±0.28
ECA	0.59	93.72±0.14	0.61	72.63±0.35
DIANet	0.81	93.88±0.21	0.83	73.87±0.27
MRLA-L	0.62	94.28±0.26	0.65	74.18±0.17
DLA-L (ours)	0.80	**94.55±0.13**	0.82	**75.46±0.26**
ResNet-110	1.15	93.04±0.33	1.17	73.00±0.36
SE	1.28	94.09±0.11	1.30	75.01±0.20
ECA	1.15	93.76±0.31	1.17	73.97±0.36
DIANet	1.37	94.48±0.08	1.39	75.31±0.16
MRLA-L	1.21	94.49±0.31	1.24	75.16±0.24
DLA-L (ours)	1.39	**94.66±0.23**	1.41	**76.41±0.36**

Table 1: Testing accuracy (%) on CIFAR-10 and CIFAR-100 datasets. "#P(M)" means the number of parameters (million).



(a) Testing acc. on CIFAR-10 (b) Testing acc. on CIFAR-100

Figure 4: Comparison of testing accuracy of different models with different depths.

Results on ImageNet-1K. We compare the DLA-L architecture with other challenging methods using ResNets as baselines. Experimental results, shown in Table 2, indicate that our model significantly outperforms other models. Firstly, our DLA-L exhibits an increase of 1.9% and 1.5% in top-1 accuracy to ResNet-50 and ResNet-101, respectively.

Model	Params	FLOPs	Top-1	Top-5
ResNet-50	25.6M	4.1B	76.1	92.9
SE	28.1M	4.1B	76.7	93.4
CBAM	28.1M	4.2B	77.3	93.7
A^2	34.6M	7.0B	77.0	93.5
AA	27.1M	4.5B	77.7	93.8
all GC	29.4M	4.2B	77.7	93.7
ECA	25.6M	4.1B	77.5	93.7
DIANet	28.4M	-	77.2	-
RLA$_g$	25.9M	4.5B	77.2	93.4
MRLA-L	25.7M	4.2B	77.7	93.8
DLA-L (ours)	27.2M	4.3B	**78.0**	**94.0**
ResNet-101	44.5M	7.8B	77.4	93.5
SE	49.3M	7.8B	77.6	93.9
CBAM	49.3M	7.9B	78.5	94.3
AA	47.6M	8.6B	78.7	94.4
ECA	44.5M	7.8B	78.7	94.3
RLA$_g$	45.0M	8.4B	78.5	94.2
MRLA-L	44.9M	7.9B	78.7	94.4
DLA-L (ours)	47.8M	8.1B	**78.9**	**94.5**

Table 2: Comparisons of accuracy (%) on the ImageNet-1K validation set (All results of the following methods are captured from their original papers).

Then, when compared to the channel attention method, DLA-L-50 has 0.9M fewer parameters than SE-50 and CBAM-50 [Woo *et al.*, 2018], but its top-1 accuracy is higher by 1.3% and 0.7%. DLA-L-101 has 1.5M fewer parameters than SE-101 and CBAM-101 [Woo *et al.*, 2018] while achieving a 1.3% and 0.4% top-1 accuracy increase, respectively. Meanwhile, when compared to the lightweight model, DLA-L-50 and DLA-L-101 remains a 0.5% and 0.2% higher top-1 accuracy than ECA-50 and ECA-101. DLA-L also outperforms other popular layer interaction models. With 1.2M fewer parameters than DIANet-50 , DLA-L-50 achieves a 0.8% higher top-1 accuracy than DIANet-50. Although DLA-L introduces more parameters compared to RLA$_g$ [Zhao *et al.*, 2021], it achieves a 0.8% and 0.4% higher top-1 accuracy than RLA$_g$-50 and RLA$_g$-101, respectively. Additionally, DLA-L also surpasses recent models such as A^2 [Chen *et al.*, 2018b], AA [Bello *et al.*, 2019], and GC [Cao *et al.*, 2019]. Finally, in comparison with static layer attention, DLA-L exhibits an increase of 0.3% and 0.2% in top-1 accuracy to MRLA-L-50 and MRLA-L-101 with a tolerable increase in the number of parameters. In summary, through comparisons with well-known channel attention models, layer interaction models, and various other challenging models, we have validated that our proposed DLA architecture serves as a more effective layer interaction model, outperforming other models in the domain of image classification.

4.2 Object Detection

Experimental Setup. In the context of object detection, our approach was evaluated on the COCO2017 dataset using the Faster R-CNN [Ren *et al.*, 2015] and Mask R-CNN [He *et al.*, 2017] frameworks as detectors. The implementations of

Methods	Detectors	Params	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
ResNet-50 [He *et al.*, 2016]	Faster R-CNN	41.5M	36.4	58.2	39.2	21.8	40.0	46.2
SE [Hu *et al.*, 2018]		44.0M	37.7	60.1	40.9	22.9	41.9	48.2
ECA [Wang *et al.*, 2020a]		41.5M	38.0	60.6	40.9	23.4	42.1	48.0
RLA$_g$ [Zhao *et al.*, 2021]		41.8M	38.8	59.6	42.0	22.5	42.9	49.5
BA [Zhao *et al.*, 2022]		44.7M	39.5	61.3	43.0	24.5	43.2	50.6
MRLA-L [Fang *et al.*, 2023]		41.7M	40.4	61.5	44.0	24.2	44.1	52.7
DLA-L (ours)		44.2M	**40.6**	**61.6**	**44.2**	**24.5**	**44.2**	**52.9**
ResNet-101 [He *et al.*, 2016]		60.5M	38.7	60.6	41.9	22.7	43.2	50.4
SE [Hu *et al.*, 2018]		65.2M	39.6	62.0	43.1	23.7	44.0	51.4
ECA [Wang *et al.*, 2020a]		60.5M	40.3	62.9	44.0	24.5	44.7	51.3
RLA$_g$ [Zhao *et al.*, 2021]		60.9M	41.2	61.8	44.9	23.7	45.7	53.8
MRLA-L [Fang *et al.*, 2023]		60.9M	42.0	63.1	45.7	25.0	45.8	55.4
DLA-L (ours)		63.4M	**42.3**	**63.3**	**45.8**	**25.2**	**46.0**	**55.5**
ResNet-50 [He *et al.*, 2016]	Mask R-CNN	44.2M	37.2	58.9	40.3	34.1	55.5	36.2
SE [Hu *et al.*, 2018]		46.7M	38.7	60.9	42.1	35.4	57.4	37.8
ECA [Wang *et al.*, 2020a]		44.2M	39.0	61.3	42.1	35.6	58.1	37.7
NL [Wang *et al.*, 2018]		46.5M	38.0	59.8	41.0	34.7	56.7	36.6
GC [Cao *et al.*, 2019]		54.4M	39.9	62.2	42.9	36.2	58.7	38.3
RLA$_g$ [Zhao *et al.*, 2021]		44.4M	39.5	60.1	43.4	35.6	56.9	38.0
BA [Zhao *et al.*, 2022]		47.3M	40.5	61.7	44.2	36.6	58.7	38.6
MRLA-L [Fang *et al.*, 2023]		44.3M	41.2	62.3	45.1	37.1	59.1	39.6
DLA-L (ours)		46.9M	**41.4**	**62.5**	**45.3**	**37.2**	**59.3**	**39.7**
ResNet-101 [He *et al.*, 2016]		63.2M	39.4	60.9	43.3	35.9	57.7	38.4
SE [Hu *et al.*, 2018]		67.9M	40.7	62.5	44.3	36.8	59.3	39.2
ECA [Wang *et al.*, 2020a]		63.2M	41.3	63.1	44.8	37.4	59.9	39.8
NL [Wang *et al.*, 2018]		65.5M	40.8	63.1	44.5	37.1	59.9	39.2
GC [Cao *et al.*, 2019]		82.2M	41.7	63.7	45.5	37.6	60.5	39.8
RLA$_g$ [Zhao *et al.*, 2021]		63.6M	41.8	62.3	46.2	37.3	59.2	40.1
MRLA-L [Fang *et al.*, 2023]		63.5M	42.8	63.6	46.5	38.4	60.6	41.0
DLA-L (ours)		66.1M	**42.9**	**63.8**	**46.7**	**38.6**	**60.9**	**41.2**

Table 3: The object detection results on the COCO val2017 with Faster R-CNN and Mask R-CNN.

all detectors were carried out using the MMDetection toolkit [Chen *et al.*, 2019], following the default settings. In the preprocessing stage, the shorter side of input images was resized to 800 pixels. The optimization process employed SGD with a weight decay of 1e-4, momentum of 0.9, and a batch size of 8. The models underwent training for a total of 12 epochs, starting with an initial learning rate of 0.01. Learning rate adjustments occurred at the 8th and 11th epochs, with a reduction by a factor of 10 each time.

Results on COCO2017. As shown in Table 3, when utilizing Faster R-CNN as the detectors, our proposed DLA-L demonstrates a remarkable improvement in average precision (AP) of 4.2% and 3.6% on ResNet-50 [He *et al.*, 2016] and ResNet-101 [He *et al.*, 2016], respectively, which validates the outstanding capability of the DLA architecture in enhancing object detection performance. Notably, DLA-L outperforms other models, when compared to the challenging channel attention block, DLA-L-50 exhibits similar parameter counts to SE-50 [Hu *et al.*, 2018] but achieves a higher AP by 2.9% and a 3.3% improvement on AP_{75}. When compared to representative layer interaction models, DLA-L continues to exhibit excellence. Despite introducing more parameters compared to RLA$_g$ [Zhao *et al.*, 2021], there is an excellent increase in AP by 1.8% and 1.1% on DLA-L-50 and DLA-L-101, respectively. Meanwhile, in contrast to a static layer attention model, MRLA-L [Fang *et al.*, 2023], our DLA-L achieves a respective increase of 0.2% and 0.3% in AP. When utilizing Mask R-CNN as the detector, our DLA-L also outperforms the aforementioned models. Additionally, it surpasses NL [Wang *et al.*, 2018], GC [Cao *et al.*, 2019], BA [Zhao *et al.*, 2022], and other models, showcasing the remarkable potential of the DLA architecture in facilitating dynamic modification in inter-layer information, even with the introduction of a tolerable parameter increment.

5 Ablation Study

5.1 Evaluating Different RNN Blocks in DLA-L

In order to validate the effectiveness of our proposed DSU block in implementing dynamic layer attention, we incorporated the original RNN block, DIA block and LSTM block as plugins into DLA to replace DSU block for comparative experiments. Due to limitations in computational resources, our ablation experiments were conducted using ResNet-110 as the baseline on CIFAR-100. Each experiment was run five times, and the results were expressed in the form of Accuracy $\pm$ Std.

Block	Params	Top-1 acc. (%)
Original RNN	1.41M	73.65$\pm$0.23
LSTM	1.93M	75.32$\pm$0.34
DIA	1.46M	75.60$\pm$0.36
DSU	1.39M	**76.41$\pm$0.36**

Table 4: Testing accuracy of different RNN blocks in DLA-110 on the CIFAR-100 dataset.

As illustrated in Table 4, the dynamic layer attention implemented by the four RNN blocks exhibits varying performance. The original RNN blocks show the poorest performance, with even a little top-1 accuracy increase than the baseline. LSTM block and the DIA block demonstrate comparable performance, outperforming baseline by 2.32% and 2.60%, respectively. However, the LSTM block has an additional parameter count of 0.47M compared to the DIA block, making it impractical applying LSTM in DLA. On the other hand, our proposed DSU block, utilizing even fewer parameters, achieves superior results. In comparison to the DIA block, we have 0.07M fewer parameters, leading to a 0.81% improvement in top-1 accuracy. Therefore, it can be inferred that our proposed DSU is the most effective block among existing RNN blocks for implementing DLA.

5.2 Evaluating the Introduced σ Function in DSU

Compared to other RNN blocks, our DSU block first normalizes c^{m-1} using an activation function, where we employ the Sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ for this purpose. To contrast the role of the σ function utilized in DSU, we will substitute it with various activation functions, including Identity mapping, Tanh, and ReLU functions.

Model	Function	Top-1 acc. (%)
DLA-L-56	Identity mapping	74.63$\pm$0.14
	Tanh	74.88$\pm$0.21
	ReLU	74.95$\pm$0.40
	Sigmoid	**75.46$\pm$0.26**
DLA-L-110	Identity mapping	75.37$\pm$0.38
	Tanh	75.67$\pm$0.20
	ReLU	75.99$\pm$0.29
	Sigmoid	**76.41$\pm$0.36**

Table 5: Testing accuracy of different activation functions used for normalizing c^{m-1} in DLA-L on CIFAR-100 dataset.

As shown in Table 5, we conducted experiments on the CIFAR-100 dataset using DLA-L-56 and DLA-L-110. Firstly, we observed in DSU that normalizing c^{m-1} is essential compared to Identity Mapping. The inclusion of an activation function resulted in significantly higher top-1 accuracy compared to Identity mapping. Secondly, among various activation functions, the Sigmoid function has been proven effective in scaling c^{m-1} to the range of $(0, 1)$, facilitating better fusion with y^m as input for the DSU block. Experimental results confirm this observation, achieving impressive top-1 accuracy of 75.46% and 76.41% on DLA-L-56 and DLA-L-110, respectively, far surpassing the baseline.

5.3 Evaluating DSU Block in Layer-wise Information Integration

We also attempt to evaluate that whether the proposed DSU block could be deployed to other layer interaction methods. [Huang *et al.*, 2020] introduced a Layer-wise Information Integration (LII) architecture that shared a RNN block throughout different network layers. And DIANet serves as a simple and effective form of LII architecture, demonstrating notable achievements in image classification. We evaluated the performance of the proposed DSU, LSTM, and DIA blocks, when integrating them into the LII. Additionally, we employed ResNets as the backbone and conduct experimental comparisons on the CIFAR-100 dataset.

Block	Params	Top-1 acc. (%)
ResNet-56	0.61M	72.32$\pm$0.36
LII (LSTM)	1.31M	69.28$\pm$0.44
LII (DIA)	0.83M	73.87$\pm$0.27
LII (DSU)	0.79M	**74.23$\pm$0.22**
ResNet-110	1.17M	73.00$\pm$0.36
LII (LSTM)	1.86M	71.31$\pm$0.33
LII (DIA)	1.39M	**75.31$\pm$0.16**
LII (DSU)	1.35M	75.14$\pm$0.21

Table 6: Testing accuracy of different RNN blocks in layer-wise information integration on the CIFAR-100 dataset.

As shown in Table 6, the experimental results demonstrate that the RNN blocks exhibits favorable performance when applied to LII. On the CIFAR-100 dataset, our DSU block shows improvements of 1.91% and 2.14% on ResNet-56 and ResNet-110, respectively. While the introduced parameters are 0.04M smaller than DIA block, DSU block outperforms DIA block by 0.36% on ResNet-56. Meanwhile, DSU block performs comparably with DIA block on ResNet-110. On the contrary, LSTM block has the highest number of parameters but performs the worst. Therefore, it could be concluded that DSU block is also a challenging method in LII, which could achieve comparable results to DIA block in LII architecture while having fewer parameters.

6 Conclusion

In this paper, we first unveil the inherent static nature of existing layer attention mechanisms and analyze their limitations in facilitating feature extraction through layer interaction. To address these limitations and restore the dynamic features of attention mechanisms, we propose and construct a framework called Dynamic Layer Attention (DLA). For implementing DLA, we design a novel RNN block, named Dynamic Sharing Unit (DSU). Experimental evaluations in the domains of image classification and object detection demonstrate that our framework outperforms static layer attention significantly in promoting layer interaction.

Acknowledgments

This work was supported by the National Major Science and Technology Projects of China under Grant 2018AAA0100201, the National Natural Science Foundation of China under Grant 62206189, and the China Postdoctoral Science Foundation under Grant 2023M732427.

References

[Bello *et al.*, 2019] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, pages 3286–3295, 2019.

[Cao *et al.*, 2019] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV*, pages 0–0, 2019.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[Chen *et al.*, 2018a] Dapeng Chen, Hongsheng Li, Tong Xiao, Shuai Yi, and Xiaogang Wang. Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding. In *CVPR*, pages 1169–1178, 2018.

[Chen *et al.*, 2018b] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A^ 2-nets: Double attention networks. *NeurIPS*, 2018.

[Chen *et al.*, 2019] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[Chen *et al.*, 2023] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *ICCV*, pages 12312–12321, 2023.

[Fang *et al.*, 2023] Yanwen Fang, Yuxi Cai, Jintai Chen, Jingyu Zhao, Guangjian Tian, and Guodong Li. Cross-layer retrospective retrieving via layer attention. *arXiv preprint arXiv:2302.03985*, 2023.

[Han *et al.*, 2021] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 7436–7456, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[He *et al.*, 2019] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, pages 558–567, 2019.

[He *et al.*, 2020] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, pages 1735–1780, 1997.

[Hu *et al.*, 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[Huang *et al.*, 2016] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[Huang *et al.*, 2018] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *CVPR*, pages 2752–2761, 2018.

[Huang *et al.*, 2019] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019.

[Huang *et al.*, 2020] Zhongzhan Huang, Senwei Liang, Mingfu Liang, and Haizhao Yang. Dianet: Dense-and-implicit attention network. In *AAAI*, pages 4206–4214, 2020.

[Jiao *et al.*, 2023] Jiayu Jiao, Yu-Ming Tang, Kun-Yu Lin, Yipeng Gao, Jinhua Ma, Yaowei Wang, and Wei-Shi Zheng. Dilateformer: Multi-scale dilated transformer for visual recognition. *IEEE Transactions on Multimedia*, 2023.

[Li and Chen, 2020] Duo Li and Qifeng Chen. Deep reinforced attention learning for quality-aware visual recognition. In *ECCV*, pages 493–509, 2020.

[Li *et al.*, 2019] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, pages 510–519, 2019.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

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

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[Srivastava *et al.*, 2015] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *NeurIPS*, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[Wang *et al.*, 2018] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[Wang *et al.*, 2020a] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *CVPR*, pages 11534–11542, 2020.

[Wang *et al.*, 2020b] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. *NeurIPS*, pages 2432–2444, 2020.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018.

[Xu *et al.*, 2017] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *ICCV*, pages 4733–4742, 2017.

[Yang *et al.*, 2020] Le Yang, Yizeng Han, Xi Chen, Shiji Song, Jifeng Dai, and Gao Huang. Resolution adaptive networks for efficient inference. In *CVPR*, pages 2369–2378, 2020.

[Yang *et al.*, 2021] Le Yang, Haojun Jiang, Ruojin Cai, Yulin Wang, Shiji Song, Gao Huang, and Qi Tian. Condensenet v2: Sparse feature reactivation for deep networks. In *CVPR*, pages 3569–3578, 2021.

[Zhao *et al.*, 2021] Jingyu Zhao, Yanwen Fang, and Guodong Li. Recurrence along depth: Deep convolutional neural networks with recurrent layer aggregation. *NeurIPS*, pages 10627–10640, 2021.

[Zhao *et al.*, 2022] Yue Zhao, Junzhou Chen, Zirui Zhang, and Ronghui Zhang. Ba-net: Bridge attention for deep convolutional neural networks. In *ECCV*, pages 297–312, 2022.