Supplementary of Stripe Observation Guided Inference Cost-free Attention Mechanism

Zhongzhan Huang[1], Shanshan Zhong[1], Wushao Wen[1], Jinghui Qin[2], and Liang Lin[1,3]

[1] Sun Yat-sen University
[2] Guangdong University of Technology
[3] Peng Cheng Laboratory
{huangzhzh23,zhongshsh5}@mail2.sysu.edu.cn
Correspondence: linliang@ieee.org

Abstract. Structural re-parameterization (SRP) is a novel technique series that boosts neural networks without introducing any computational costs in inference stage. The existing SRP methods have successfully considered many architectures, such as normalizations, convolutions, etc. However, the widely used but computationally expensive attention modules cannot be directly implemented by SRP due to the inherent multiplicative manner and the modules' output is input-dependent during inference. In this paper, we statistically discover a counter-intuitive phenomenon *Stripe Observation* in various settings, which reveals that channel attention values consistently approach some constant vectors during training. It inspires us to propose a novel attention-alike SRP, called ASR, that allows us to achieve SRP for a given network while enjoying the effectiveness of the attention mechanism. Extensive experiments conducted on several standard benchmarks show the effectiveness of ASR in generally improving the performance of various scenarios without any elaborated model crafting. We also provide experimental evidence for how the proposed ASR can enhance model performance. https://github.com/zhongshsh/ASR.

Keywords: Structural Re-parameterization · Attention Mechanism

Table 1: The significant decrease in inference speed (Frames Per Second) by using attention modules. "#P." denotes the number of parameters.

	CIFAR100		STL10			CIFAR100		STL10	
Model	#P. (M)	Speed	#P. (M)	Speed	Model	#P. (M)	Speed	#P. (M)	Speed
ResNet164	1.73	1944	1.70	255	ResNet164	1.73	1944	1.70	255
+IE [31, 39]	1.74	1505 (↓ 22.56%)	1.72	170 (↓ 33.26%)	+SRM [29]	1.76	1387 (↓ 28.64%)	1.74	162 (↓ 36.40%)
+CBAM [41]	1.93	793 (↓ 59.21%)	1.90	127 (↓ 50.23%)	+DIA [25]	1.95	1092 (↓ 43.82%)	1.92	154 (↓ 39.61%)
+SE [19]	1.93	1469 (↓ 24.42%)	1.91	173 (↓ 32.08%)	+SPA [13]	3.86	1080 (↓ 44.45%)	3.83	180 (↓ 29.36%)
+ASR (SE)	1.73	1942 (∼ 0.00%)	1.70	255 (∼ 0.00%)	+ASR (SPA)	1.73	1946 (∼ 0.00%)	1.70	253 (∼ 0.00%)

A ASR for different network layers in inference phase

In the main text, we find that the "attention values" $\mathbf{v}_{\psi,\theta}$ generated by ASR are some constant vector, for the various common-used modules $\mathbf{B}_{\hat{\theta}}$ in the backbone, we can seamlessly find the corresponding transformation g such that

$$\mathbf{B}_{\hat{\theta}} \odot \mathbf{v}_{\psi,\theta} = \mathbf{B}_{g[\hat{\theta},\psi,\theta]}, \tag{1}$$

(1) For the convolutional layer, if $\mathbf{B}_{\hat{\theta}}$ is a convolutional layer $\mathcal{C}$ with kernels $\mathbf{K}$ and bias $\mathbf{b}$, then we have

$$\begin{aligned} \mathcal{C}(\mathbf{x};\mathbf{K},\mathbf{b}) \odot \mathbf{v}_{\psi,\theta} &= (\mathbf{x} * \mathbf{K}) \odot \mathbf{v}_{\psi,\theta} + \mathbf{b} \odot \mathbf{v}_{\psi,\theta} \\ &= \mathbf{x} * (\mathbf{K} \odot \mathbf{v}_{\psi,\theta}) + \mathbf{b} \odot \mathbf{v}_{\psi,\theta} \\ &= \mathcal{C}(\mathbf{x};\mathbf{K} \odot \mathbf{v}_{\psi,\theta}, \mathbf{b} \odot \mathbf{v}_{\psi,\theta}), \\ &\equiv \mathcal{C}(\mathbf{x};\mathbf{K}',\mathbf{b}') \end{aligned} \tag{2}$$

where $*$ denote convolution and $\mathbf{K} \odot \mathbf{v}_{\psi,\theta}$ means that the the product of i-th elements of $\mathbf{v}_{\psi,\theta}$ and i-th kernel of $\mathbf{K}$. Since the existing SRP methods mainly merge various neural network layers into a convolutional layer, therefore ASR is compatible with most of these SRP methods.

(2) For the normalization layer κ, like batch normalization [37], instance normalization [28], group normalization [42], etc., they generally can be formulated as

$$\kappa(x;\mu,\sigma,\gamma,\beta) = \frac{x-\mu}{\sigma} \odot \gamma + \beta, \tag{3}$$

where $\mu, \sigma, \gamma, \beta$ are the parameters of each kind of normalization method. For ASR, the Eq.(1) can be rewrittern as

$$\begin{aligned} \kappa(\mathbf{x};\mu,\sigma,\gamma,\beta) \odot \mathbf{v}_{\psi,\theta} &= \frac{(\mathbf{x}-\mu) \odot \gamma \odot \mathbf{v}_{\psi,\theta}}{\sigma} + \beta \odot \mathbf{v}_{\psi,\theta} \\ &= \kappa(\mathbf{x};\gamma \odot \mathbf{v}_{\psi,\theta}, \beta \odot \mathbf{v}_{\psi,\theta}), \\ &\equiv \kappa(\mathbf{x};\mu,\sigma,\gamma',\beta') \end{aligned} \tag{4}$$

(3) For the fully connected layer $f(x) = Wx$, we have

$$\begin{aligned} f(x) \odot \mathbf{v}_{\psi,\theta} &= Wx \odot \mathbf{v}_{\psi,\theta} \\ &= (W \odot \mathbf{v}_{\psi,\theta})x \equiv W'x \end{aligned} \tag{5}$$

(4) For the transformer-based attention layer T, we have

$$\begin{aligned} T(x;W_Q,W_K,W_V) \odot \mathbf{v}_{\psi,\theta} &= \frac{W_Q x (W_K x)^T}{\sqrt{d_k}} W_V x \odot \mathbf{v}_{\psi,\theta} \\ &= \frac{W_Q x (W_K x)^T}{\sqrt{d_k}} (W_V \odot \mathbf{v}_{\psi,\theta}) x \\ &\equiv \frac{W_Q x (W_K x)^T}{\sqrt{d_k}} W'_V x \qquad \text{Since Eq.(5)} \\ &= T(x;W_Q,W_K,W'_V) \end{aligned}$$

B Introdcution of implementation details

In Section B.1, we present the experimental details, followed by an explanation in Section B.2 on how ASR is incorporated into various backbones.

B.1 Experiment details

Unless otherwise specified, we follow the settings of [4,8,15,18,33,35]. Specifically, all models using STL10, CIFAR10, and CIFAR100 datasets with epoch set to 164. During training, we apply standard data augmentation techniques such as normalization, random cropping, and horizontal flipping. The batch size of CIFAR10, CIFAR100, and STL10 is 128, 128, 16, respectively. The other hyper-parameter settings of CIFAR10, CIFAR100, STL10 and ImageNet are shown in Table 2 and Table 3 respectively. The patch size of ViT is 16.

	ResNet83	ResNet164	VGG19	ShuffleNetV2	MobileNet	RepVGG	ResNet-ACNet
optimizer	SGD (0.9)	SGD (0.9)	SGD (0.9)	SGD (0.9)	SGD (0.9)	SGD (0.9)	SGD (0.9)
schedule	81/122	81/122	60/120/160	60/120/160	60/120/160	130	cosine annealing
weight decay	1.00E-04	1.00E-04	5.00E-04	5.00E-04	5.00E-04	1.00E-04	1.00E-04
gamma	0.1	0.1	0.2	0.2	0.2	0.1	0.333
lr	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table 2: Implementation details for **CIFAR10/100, STL10** image classification. Normalization and standard data augmentation (random cropping and horizontal flipping) are applied to the training data.

	ResNet34	ResNet50	ResNet101	ViT-S	ViT-B	ViT-B ↑ 384
optimizer	SGD (0.9)	SGD (0.9)	SGD (0.9)	AdamW	AdamW	AdamW
schedule	30/60/90	30/60/90	30/60/90	cosine annealing	cosine annealing	cosine annealing
weight decay	1.00E-04	1.00E-04	1.00E-04	5.00E-02	5.00E-02	1.00E-08
gamma	0.1	0.1	0.1	-	-	-
lr	0.1	0.1	0.1	5.00E-04	5.00E-04	5.00E-06
epoch	100	100	100	300	300	30
batch size	128	128	128	256	256	64

Table 3: Implementation details for **ImageNet 2012** image classification. Normalization and standard data augmentation (random cropping and horizontal flipping) are applied to the training data. The random cropping of size 224 by 224 is used in these experiments.

B.2 Application details

In this section, we provide detailed information on how we apply ASR to different architectures during the training phase. We focus on ResNet and ResNet-ACNet,

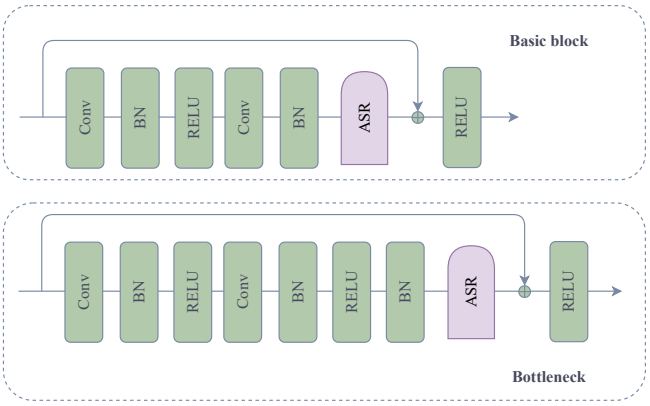

Fig. 1: ASR in the blocks of ResNet during the training phase.

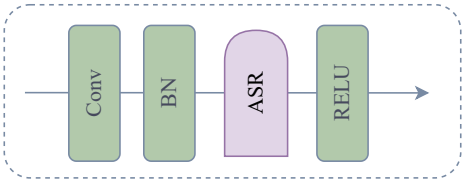

Fig. 2: ASR in the block of VGG during the training phase.

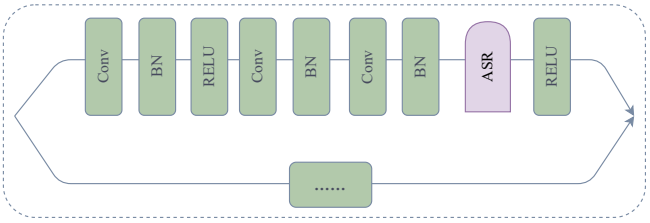

Fig. 3: ASR in the block of ShuffleNetV2 during the training phase.

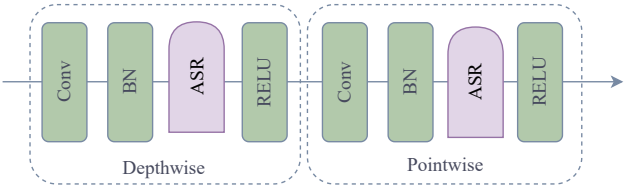

Fig. 4: ASR in the block of MobileNet during the training phase.

Name	Explanation
optimizer	Optimizer
depth	The depth of the network
schedule	Decrease learning rate at these epochs
wd	Weight decay
gamma	The multiplicative factor of learning rate decay
lr	Initial learning rate

Table 4: The additional explanation.

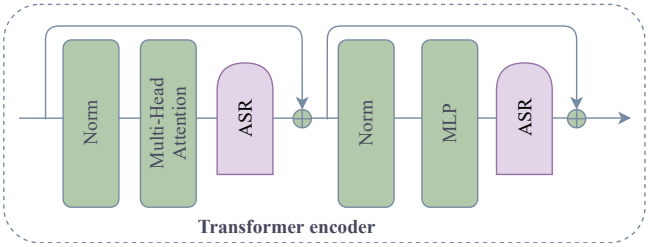

Fig. 5: ASR in the transformer encoder of ViT during the training phase.

VGG and RepVGG, ShuffleNetV2, MobileNet, and ViT. For each architecture, we specify the location of ASR within the block and provide a visualization to facilitate understanding. Normally, the placement of ASR is typically situated post the individual layers within the model architecture. Specifically, it is conventionally positioned subsequent to batch normalization and prior to the integration of residual connections and ReLU. In essence, ASR plays the role of calibration for the outputs of each stratum within the model architecture.

ResNet and ResNet-ACNet are popular convolutional neural network architectures that use basic blocks or bottlenecks. In our experiments, we insert ASR after the last batch normalization layer and before the residual addition operation, as shown in Fig. 1. We use Sigmoid as the activation function $\sigma(\cdot)$ of ASR, and we set the initial value of the learnable vector ψ to 0.1.

VGG and RepVGG are two VGG-type convolutional neural network architectures that are widely used in computer vision applications. In our experiments, we insert ASR between the batch normalization layer and ReLU activation function, as shown in Fig. 2. This location allows ASR to process the feature maps before they are passed to the next convolutional layer, which helps to reduce the batch noise [31] and distortion caused by the convolution. We use Sigmoid as the activation function $\sigma(\cdot)$ of ASR, and we set the initial value of the learnable vector ψ to 0.1.

ShuffleNetV2 is a lightweight convolutional neural network architecture that uses a dual-branch structure for its block. In our experiments, we insert ASR between the last batch normalization layer and ReLU activation function in the

residual branch, as shown in Fig. 3. We use Sigmoid as the activation function $\sigma(\cdot)$ of ASR, and we set the initial value of the learnable vector ψ to 0.1.

MobileNet is another lightweight convolutional neural network architecture that uses depthwise and pointwise convolutional layers in each block. In our experiments, we insert ASR between BN and ReLU in both layers, as demonstrated in Fig. 4. This placement allows ASR to capture the non-linearity of both convolutional layers. We use Sigmoid as the activation function $\sigma(\cdot)$ of ASR, and we set the initial value of the learnable vector ψ to 0.1.

ViT. For ViT, we apply ASR to the transformer encoder, as illustrated in Fig. 5. ASR is inserted after multi-head attention and MLP. Moreover, distinct from other backbones, the transformer-based model ViT diverges from the conventional notion of channels. Instead, tokens and their associated features constitute the vectors from ViT. Therefore, we apply ASR to the dimension of the features. The activation function $\sigma(\cdot)$ employed within ASR is Tanh, and the initial value of the learnable vector ψ is set to 1e-3. Correspondingly, the initial values of the network layers within ASR are also configured to 1e-3.

C The initialization of ASR

In this section, we analyze the initialization of the input $\psi \in R^{C\times 1\times 1}$ in ASR. We conduct experiments by initializing ψ with values ranging from 0.1 to 0.6 and evaluate the performance of ASR on the CIFAR100 dataset as shown in Table 5. Our findings suggest that an appropriate initialization value is crucial for the performance of ASR. Specifically, we find that ASR initialized with a value of 0.1 achieves the highest accuracy of 74.83% and 74.77% on the test set, while the accuracy of ASR with other initialization values varies from 74.03% to 74.73%. These results indicate that choosing an appropriate initialization value can significantly impact the performance of ASR, and initializing ASR with a value of 0.1 leads to the best performance on the CIFAR100 dataset.

Initialization	0.1	0.2	0.3	0.4	0.5	0.6
ASR (SE)	**74.83**	<u>74.56</u>	74.15	74.17	74.03	74.15
ASR (IE)	**74.77**	74.55	<u>74.73</u>	74.25	74.45	74.24

Table 5: Top-1 accuracy (%) of different initialization values on ASR's performance. The backbone is ResNet83, and the dataset is CIFAR100. Bold and underline indicate the best results and the second best results, respectively.

D Different numbers of ASR inserted at the same position

In this section, we provide additional experimental results on the impact of inserting different numbers of ASR at the same position in ResNet164 and ViT.

Specifically, we evaluate the performance of ResNet164 with 1, 2, 3, and 4 ASR modules inserted at the positions shown in Fig. 1. We report the top-1 and top-5 accuracy on the CIFAR100 validation set.

As shown in Table 6, for ResNet164, the performance generally improves as the number of ASR modules increases when $\delta < 3$. These results suggest that inserting multiple ASR modules at the same position in ResNet164 may further enhance the performance.

Module	$\delta = 1$		$\delta = 2$		$\delta = 3$		$\delta = 4$	
	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.
ASR (SE)	75.36	93.53	75.87	93.99	75.72	94.15	75.22	93.56
ASR (IE)	75.58	93.84	75.71	93.81	75.45	94.07	74.56	93.73
ASR (SRM)	75.23	93.68	75.45	94.06	75.61	94.21	75.11	93.78
ASR (SPA)	75.12	93.44	75.43	93.66	75.81	94.03	75.62	93.46

Table 6: The accuracy (%) of ResNet164 with different numbers of ASR inserted at the same position on CIFAR100.

E Related works

Attention mechanism selectively focuses on the most informative components of a network via self-information processing and has gained a promising performance on vision tasks [31]. For example, SENet [19] proposes the channel attention mechanism, which adjusts the feature map with channel view, and CBAM [41] considers both channel and spatial attention for adaptive feature refinement. Recently, more works [2, 10, 12, 17, 23, 27, 30, 34, 44, 49, 51] are proposed to optimize spatial attention and channel attention. Most of the above works regard attention mechanism as an additional module, and with the development of the transformer [38], a large number of works [9, 22, 36, 43, 47, 48] regard the attention as important parts of the backbone network.

Structural re-parameterization enables different architectures to be mutually converted through the equivalent transformation of parameters [20]. For instance, a branch of 1×1 convolution and a branch of 3×3 convolution can be transferred into a single branch of 3×3 convolution [8]. In the training phase, multi-branch [7, 8] and multi-layer [1, 14] topologies are designed to replace the vanilla layers for augmenting models. Afterward, during inference, the training-time complex models are transferred to simple ones for faster inference. Cao et al [1] have discussed how to merge a depthwise separable convolution kernel during training. Thanks to the efficiency of structural re-parameterization, it has gained great importance and has been utilized in various tasks [21,32,46,50] such as compact model design [9], architecture search [3, 24, 45], pruning [5, 16, 26], image recognition [6], and super-resolution [11, 40].

F The visualizations of the first-order difference (absolute value) for attention value over epoch

To provide further evidence for the claim made in our paper that most of the attention values almost converge at the first learning rate decay (30 epochs), we present additional visualizations of the first-order difference (absolute value) in attention value over epoch for different structures, attention modules, datasets, and training settings (including learning rate and weight decay). Each figure includes four subplots that show the evolution of attention value for different images. The horizontal axis indicates the number of epochs, while the vertical axis represents the order of random channel ID. Unless otherwise specified, we adopt ResNet83-SE as our baseline, CIFAR100 as the default dataset, and schedule learning rate as the default learning rate, with weight decay set to 1e-4.
Different backbones. Fig. 6 and Fig. 7 show the first-order difference (absolute value) in attention value over epoch for ResNet83 and ResNet164, respectively. Both backbones exhibit the same trend, indicating that most of the attention values almost converge at the first learning rate decay (30 epochs).
Different attention modules. Fig. 6 and Fig. 8 present the first-order difference (absolute value) in attention value over epoch for two attention modules SE and IE, respectively. Although IE has some cchannel that converge more slowly than others, most of the channel attention values almost converge at 30 epochs.
Different datasets. Fig. 6 and Fig. 9 compare the attention values of ResNet83-SE on CIFAR100 and STL10 during the training process. Although ResNet83-SE exhibits greater attention value fluctuations in the initial stages on the STL10, our results still align with the findings presented in our paper.
Different training setting. We also compare the first-order difference (absolute value) in attention value over epoch for ResNet83-SE under different training settings. Fig. 6 and Fig. 10 show the results of using schedule learning rate and cosine learning rate, respectively. Fig. 11, Fig. 12, and Fig. 13 correspond to weight decay values of 2e-4, 3e-4, and 4e-4, respectively. In all cases, we obtain results consistent with our paper's findings. We also observe that larger weight decay values lead to faster attention value convergence.

G The results about the batch noise attack

We conduct experiments on three types of noise attacks to empirically verify the ability of ASR in regulating noise to improve model robustness, including batch noise, constant noise, and random noise. We consider the style transfer task, which generally adopts the instance normalization (IN) without batch noise, rather than BN, as adding batch noise would significantly reduce the quality of generated images due to noise interference.

In this section, we present additional results on the batch noise attack to support the conclusions of our paper. As shown in Fig. 14, Fig. 15, and Fig. 16, with batch noise (BN), there are more blurriness compared to without batch noise (IN). However, when ASR is applied to BN, the aforementioned issues

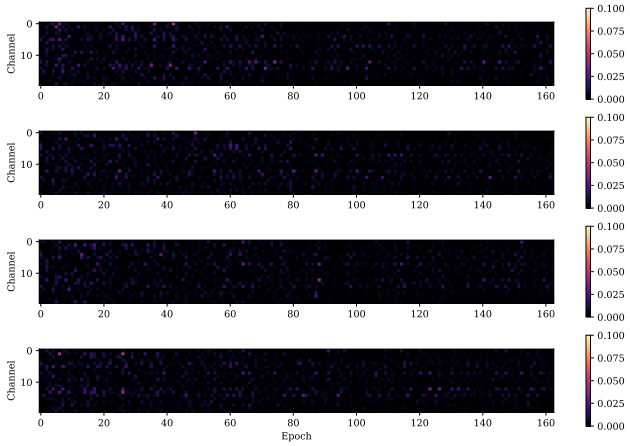

Fig. 6: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 1e-4) over epoch on CIFAR100. Zoom in for best view.

are significantly reduced. This suggests that ASR can effectively alleviate the adverse effects of noise introduced by batch normalization, resulting in image quality comparable to that of IN without batch noise.

H More examples of Stripe Observation

In this section, we present additional examples to support the conclusions of our paper that after passing through the attention module, the channel attention values of different images tend to approach a certain value within the same channel, resulting in a “stripe structure". We present in Fig. 17, Fig. 18, Fig. 19, Fig. 20, Fig. 21, Fig. 22, and Fig. 23 the visualization of attention values for different structures, attention modules, datasets, and training settings (including learning rate and weight decay values). The horizontal axis represents the order of random channels, while the vertical axis represents the order of random images. Corresponding to Appendix F, we use ResNet83-SE as the baseline, with the default dataset being CIFAR100, learning rate being schedule learning rate, and weight decay being 1e-4. All figures show an obvious "stripe structure," which is consistent with our conclusion in the paper that the attention values of different images tend to converge to a certain value within the same channel.

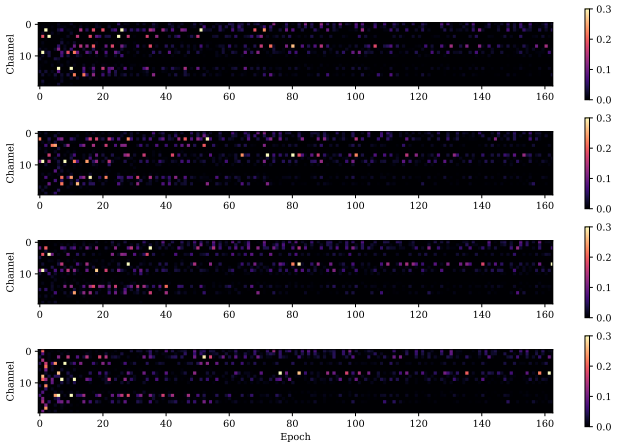

Fig. 7: The visualization of the first-order difference (absolute value) for attention value of ResNet164-SE (weight decay: 1e-4) over epoch on CIFAR100. Zoom in for best view.

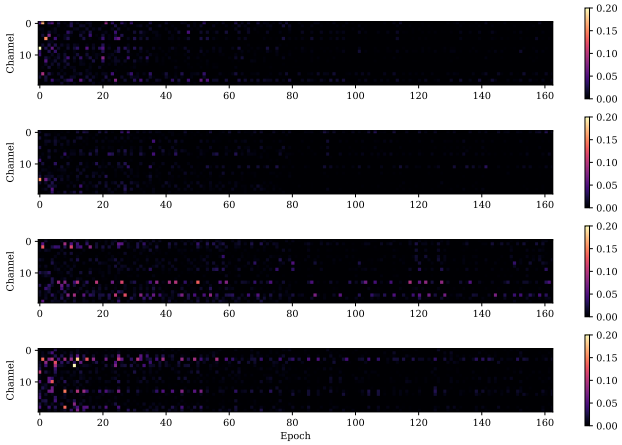

Fig. 8: The visualization of the first-order difference (absolute value) for attention value of ResNet83-IE (weight decay: 1e-4) over epoch on CIFAR100. Zoom in for best view.

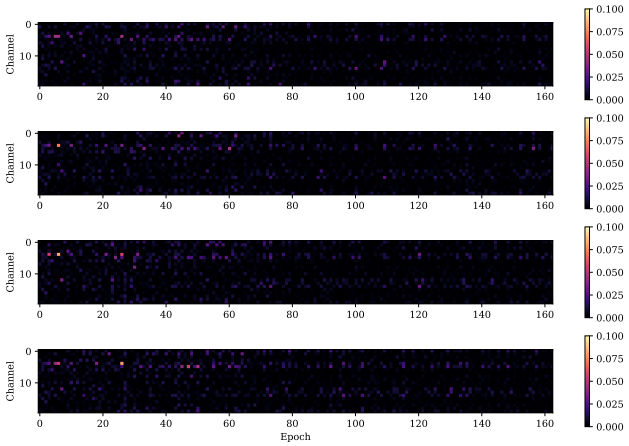

Fig. 9: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 1e-4) over epoch on STL10. Zoom in for best view.

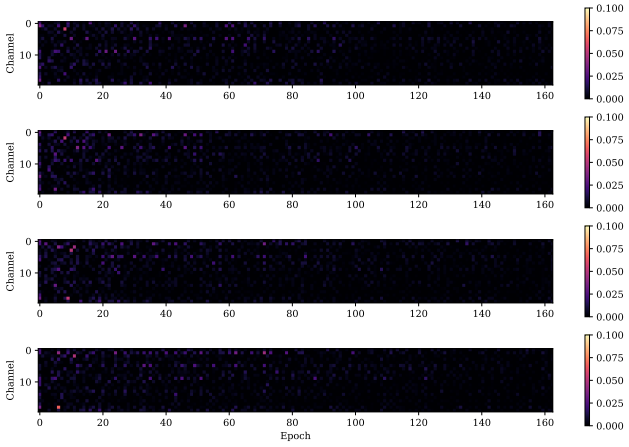

Fig. 10: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 1e-4) based on cosine learning rate over epoch on CIFAR100. Zoom in for best view.

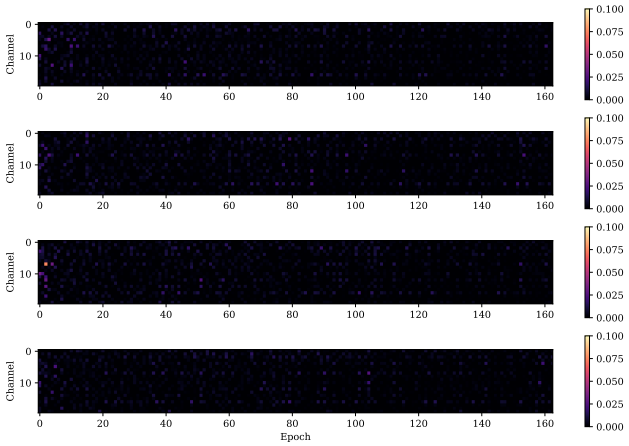

Fig. 11: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 2e-4) over epoch on CIFAR100. Zoom in for best view.

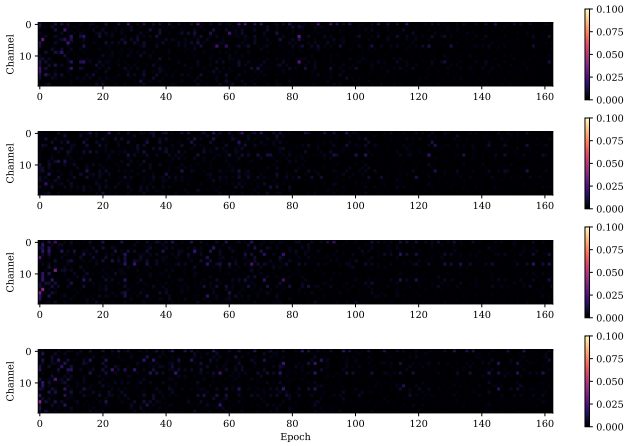

Fig. 12: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 3e-4) over epoch on CIFAR100. Zoom in for best view.

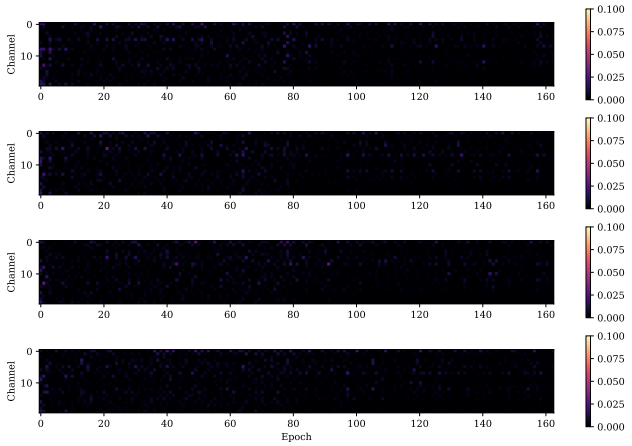

Fig. 13: The visualization of the first-order difference (absolute value) for attention value of ResNet83-SE (weight decay: 4e-4) over epoch on CIFAR100. Zoom in for best view.

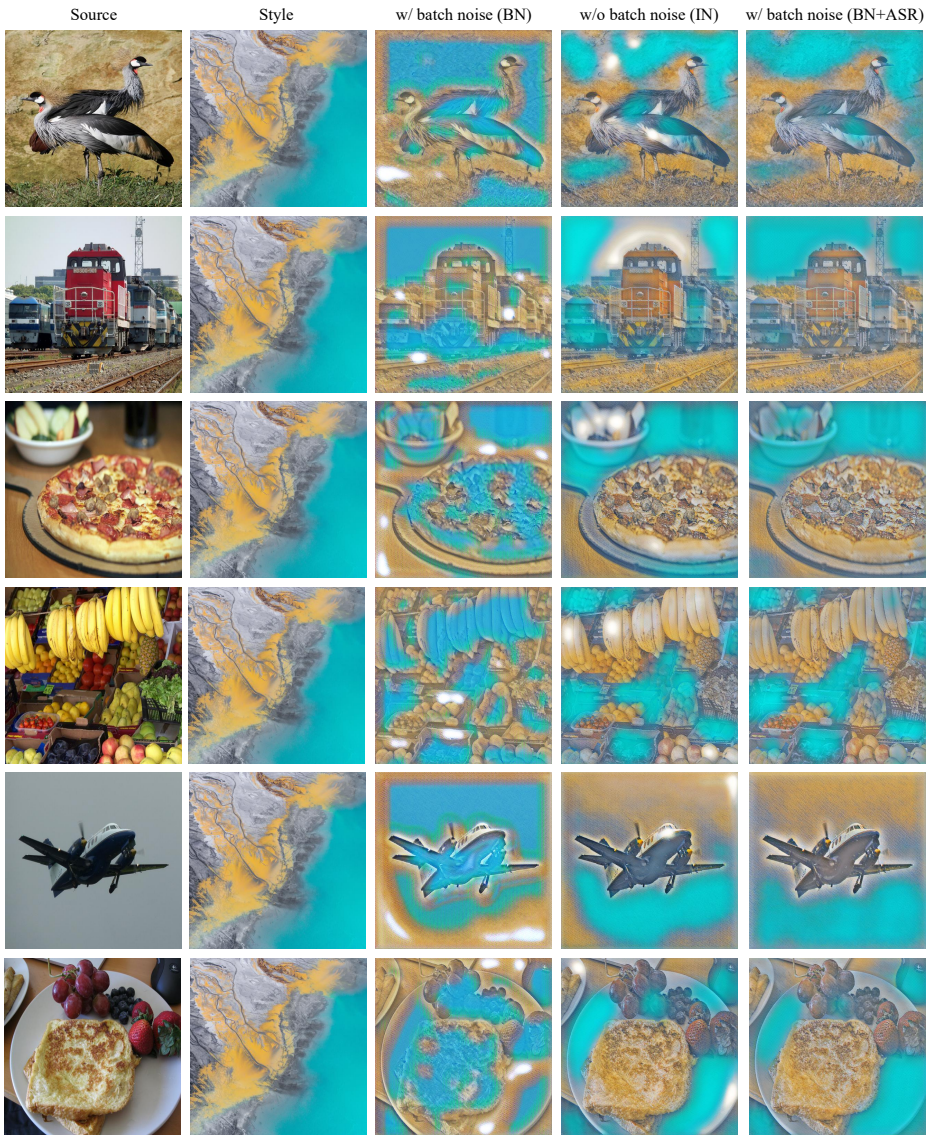

Fig. 14: The results about the batch noise attack. Zoom in for best view.

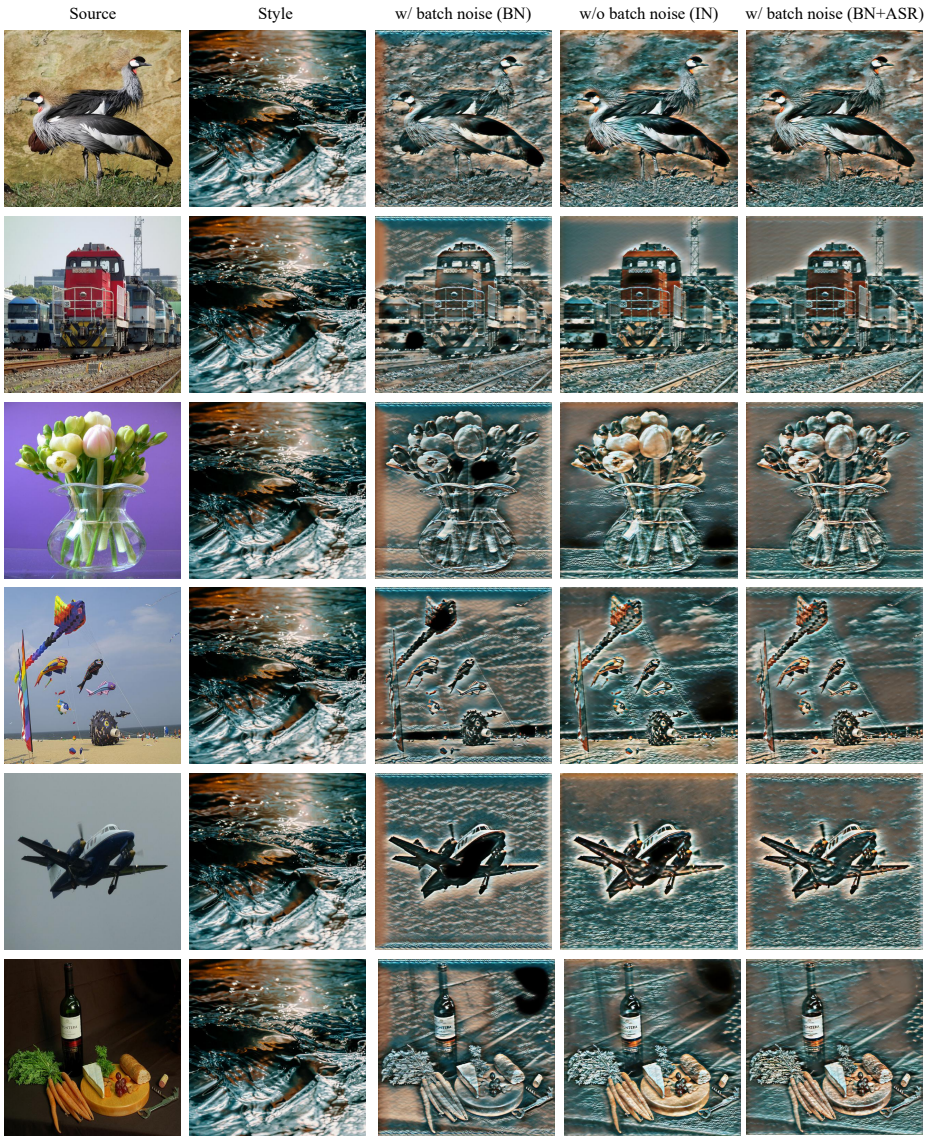

Fig. 15: The results about the batch noise attack. Zoom in for best view.

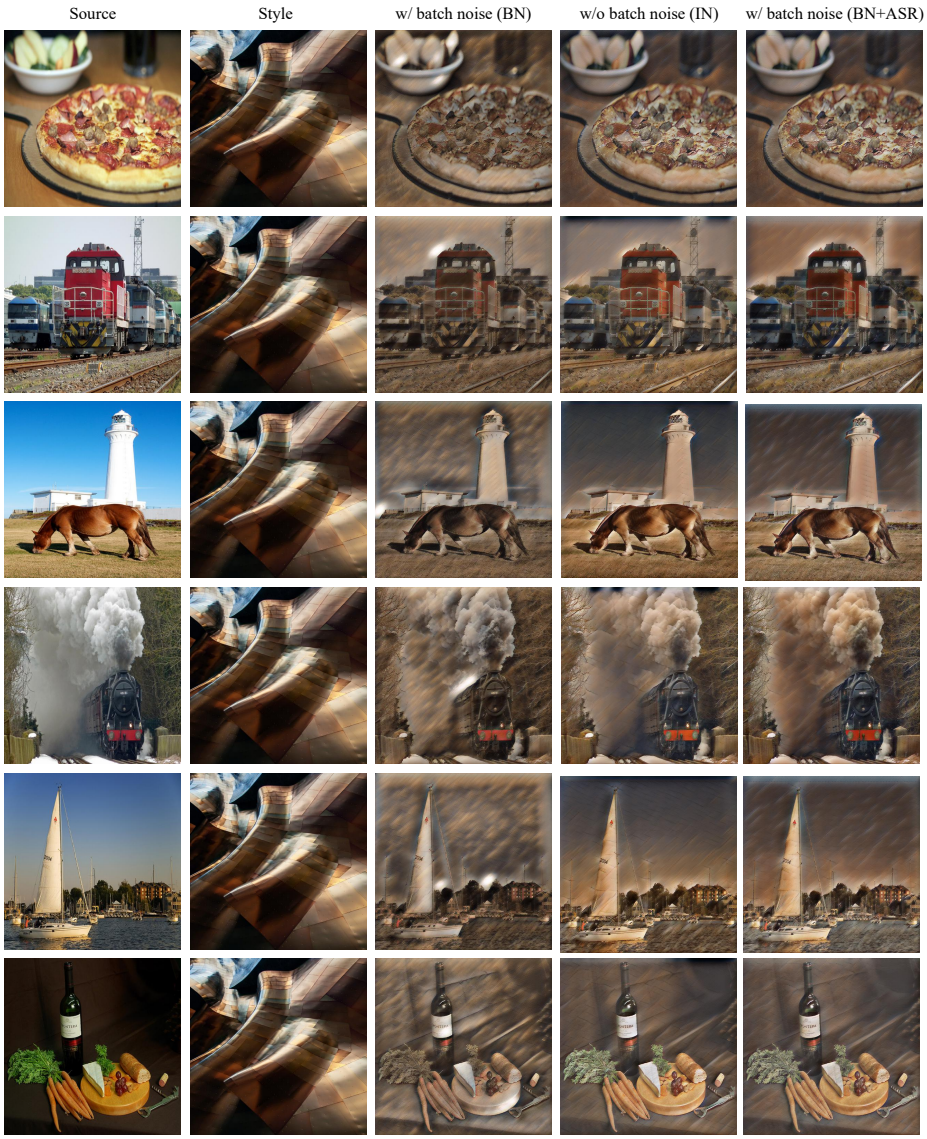

Fig. 16: The results about the batch noise attack. Zoom in for best view.

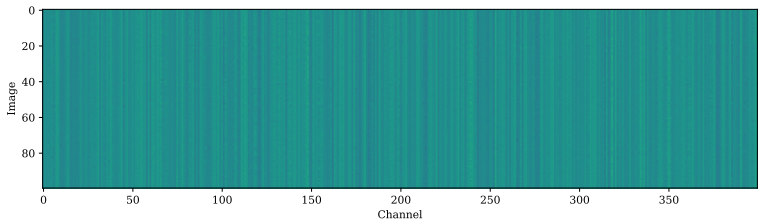

Fig. 17: The attention values of different images from ResNet83-SE on CIFAR100. Zoom in for best view.

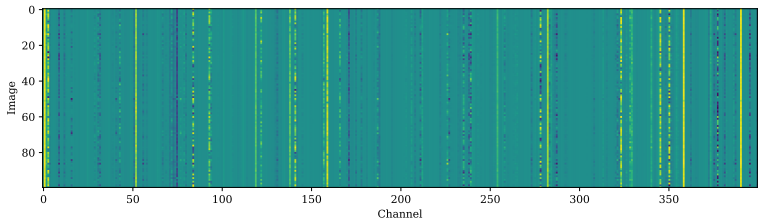

Fig. 18: The attention values of different images from ResNet164-SE on CIFAR100. Zoom in for best view.

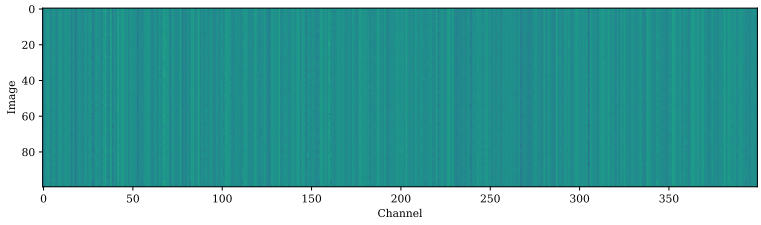

Fig. 19: The attention values of different images from ResNet83-SE based on cosine learning rate on CIFAR100. Zoom in for best view.

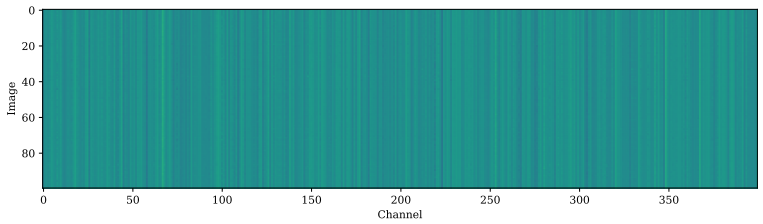

Fig. 20: The attention values of different images from ResNet83-SE on STL10. Zoom in for best view.

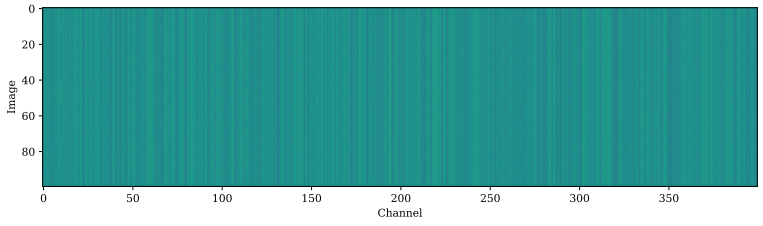

Fig. 21: The attention values of different images from ResNet83-SE (weight decay: 2e-4) on CIFAR100. Zoom in for best view.

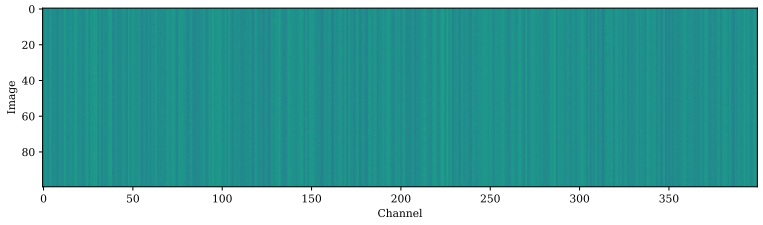

Fig. 22: The attention values of different images from ResNet83-SE (weight decay: 3e-4) on CIFAR100. Zoom in for best view.

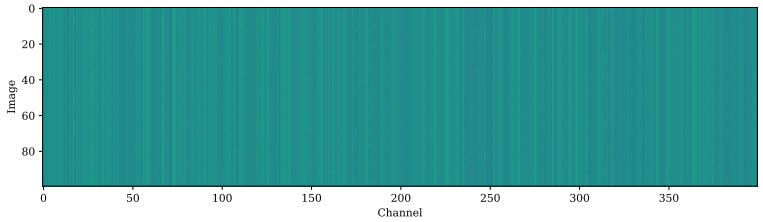

Fig. 23: The attention values of different images from ResNet83-SE (weight decay: 4e-4) on CIFAR100. Zoom in for best view.

References

1. Cao, J., Li, Y., Sun, M., Chen, Y., Lischinski, D., Cohen-Or, D., Chen, B., Tu, C.: Do-conv: Depthwise over-parameterized convolutional layer. arXiv 2006.12030 (2020)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: IEEE Conf. Comput. Vis. Worksh. pp. 0–0 (2019)
3. Chen, S., Chen, Y., Yan, S., Feng, J.: Efficient differentiable neural archetcture search with meta kernels. arXiv 1912.04749 (2019)
4. Ding, X., Guo, Y., Ding, G., Han, J.: Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1911–1920 (2019)
5. Ding, X., Hao, T., Tan, J., Liu, J., Han, J., Guo, Y., Ding, G.: Lossless cnn channel prunning via decoupling remembering and forgetting. In: ICCV (2021)
6. Ding, X., Xia, C., Zhang, X., Chu, X., Han, J., Ding, G.: Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. arXiv preprint arXiv:2105.01883 (2021)
7. Ding, X., Zhang, X., Han, J., Ding, G.: Diverse branch block: Building a convolution as an inception-like unit. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10886–10895 (2021)
8. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13733–13742 (2021)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3146–3154 (2019)

11. Gao, S., Zheng, C., Zhang, X., Liu, S., Wu, B., Lu, K., Zhang, D., Wang, N.: Rcbsr: re-parameterization convolution block for super-resolution. In: Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II. pp. 540–548. Springer (2023)
12. Gao, Z., Xie, J., Wang, Q., Li, P.: Global second-order pooling convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3024–3033 (2019)
13. Guo, J., Ma, X., Sansom, A., McGuire, M., Kalaani, A., Chen, Q., Tang, S., Yang, Q., Fu, S.: Spanet: Spatial pyramid attention network for enhanced image recognition. In: 2020 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2020)
14. Guo, S., Alvarze, J.M., Salzmann, M.: Expandnets: linear over re-parameterization to train compact convolutional networks. In: NeurIPS (2020)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 770–778 (2016)
16. He, W., Huang, Z., Liang, M., Liang, S., Yang, H.: Blending pruning criteria for convolutional neural networks. In: Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part IV 30. pp. 3–15. Springer (2021)
17. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. arXiv preprint arXiv:2103.02907 (2021)
18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
19. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7132–7141 (2018)
20. Hu, M., Feng, J., Hua, J., Lai, B., Huang, J., Gong, X., Hua, X.S.: Online convolutional re-parameterization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 568–577 (2022)
21. Huang, T., You, S., Zhang, B., Du, Y., Wang, F., Qian, C., Xu, C.: Dyrep: Bootstrapping training with dynamic re-parameterization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 588–597 (2022)
22. Huang, Z., Liang, M., Qin, J., Zhong, S., Lin, L.: Understanding self-attention mechanism via dynamical system perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1412–1422 (2023)
23. Huang, Z., Liang, M., Zhong, S., Lin, L.: Attns: Attention-inspired numerical solving for limited data scenarios. In: Forty-first International Conference on Machine Learning (2024)
24. Huang, Z., Liang, S., Liang, M., He, W., Yang, H., Lin, L.: The lottery ticket hypothesis for self-attention in convolutional neural network. arXiv preprint arXiv:2207.07858 (2022)
25. Huang, Z., Liang, S., Liang, M., Yang, H.: Dianet: Dense-and-implicit attention network. In: AAAI. pp. 4206–4214 (2020)
26. Huang, Z., Shao, W., Wang, X., Lin, L., Luo, P.: Rethinking the pruning criteria for convolutional neural network. Advances in Neural Information Processing Systems **34**, 16305–16318 (2021)
27. Huang, Z., Zhou, P., Yan, S., Lin, L.: Scalelong: Towards more stable training of diffusion model via scaling network long skip connection. Advances in Neural Information Processing Systems **36**, 70376–70401 (2023)

28. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
29. Lee, H., Kim, H.E., Nam, H.: Srm: A style-based recalibration module for convolutional neural networks. In: Int. Conf. Comput. Vis. pp. 1854–1862 (2019)
30. Liang, M., Zhou, J., Wei, W., Wu, Y.: Balancing between forgetting and acquisition in incremental subpopulation learning. In: European Conference on Computer Vision. pp. 364–380. Springer (2022)
31. Liang, S., Huang, Z., Liang, M., Yang, H.: Instance enhancement batch normalization: An adaptive regulator of batch noise. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4819–4827 (2020)
32. Luo, J., Si, W., Deng, Z.: Few-shot learning for radar signal recognition based on tensor imprint and re-parameterization multi-channel multi-branch model. IEEE Signal Processing Letters **29**, 1327–1331 (2022)
33. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
34. Qin, Z., Zhang, P., Wu, F., Li, X.: Fcanet: Frequency channel attention networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 783–792 (2021)
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
36. Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Izacard, G., Joulin, A., Synnaeve, G., Verbeek, J., et al.: Resmlp: Feedforward networks for image classification with data-efficient training. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
37. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inform. Process. Syst. pp. 5998–6008 (2017)
39. Wang, J., Chen, Y., Yu, S.X., Cheung, B., LeCun, Y.: Recurrent parameter generators. arXiv preprint arXiv:2107.07110 (2021)
40. Wang, X., Dong, C., Shan, Y.: Repsr: Training efficient vgg-style super-resolution networks with structural re-parameterization and batch normalization. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 2556–2564 (2022)
41. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: Eur. Conf. Comput. Vis. pp. 3–19 (2018)
42. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
43. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10819–10829 (2022)
44. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
45. Zhang, M., Yu, X., Rong, J., Ou, L.: Repnas: Searching for efficient re-parameterizing blocks. arXiv 2109.03508 (2021)

46. Zhang, R., Wei, J., Lu, W., Zhang, L., Ji, Y., Xu, J., Lu, X.: Cs-rep: Making speaker verification networks embracing re-parameterization. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7082–7086. IEEE (2022)
47. Zhong, S., Huang, Z., Wen, W., Yang, Z., Qin, J.: Esa: Excitation-switchable attention for convolutional neural networks. Neurocomputing **557**, 126706 (2023)
48. Zhong, S., Wen, W., Qin, J.: Spem: Self-adaptive pooling enhanced attention module for image recognition. In: International Conference on Multimedia Modeling. pp. 41–53. Springer (2023)
49. Zhong, S., Wen, W., Qin, J., Chen, Q., Huang, Z.: Lsas: Lightweight sub-attention strategy for alleviating attention bias problem. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 2051–2056. IEEE (2023)
50. Zhou, H., Liu, L., Zhang, H., He, H., Zheng, N.: Cmb: A novel structural re-parameterization block without extra training parameters. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2022)
51. Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J.: An empirical study of spatial attention mechanisms in deep networks. In: Int. Conf. Comput. Vis. pp. 6688–6697 (2019)

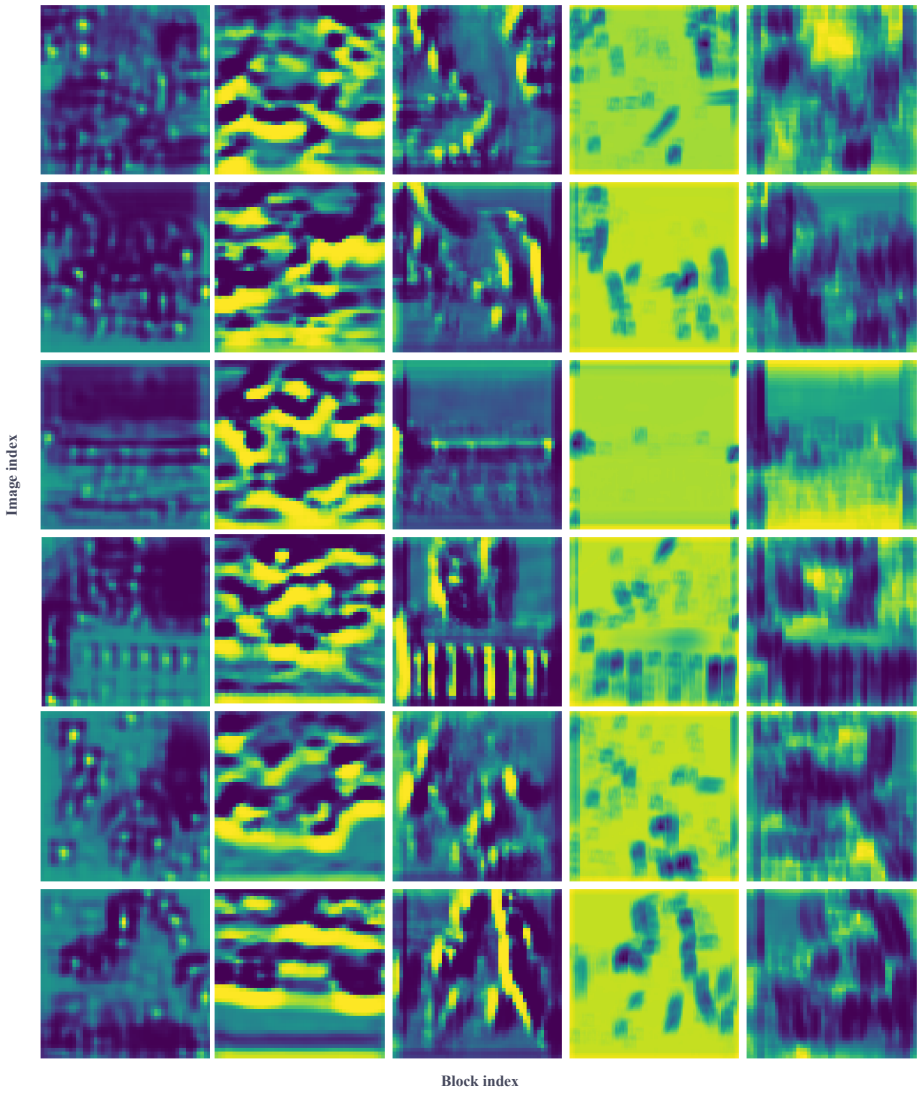

Fig. 24: The visualization about the spatial attention from CBAM. We randomly select six images from STL10 and extract the spatial attention values of five blocks from ResNet83-CBAM. Each row in the visualization represents the spatial attention values of different blocks for the same image, while each column represents the spatial attention values of different images for the same block. Zoom in for best view.