doi:10.20944/preprints202210.0092.v1

Article

Incoherence of Deep Isotropic Neural Networks Increase Their Performance on Image Classification

Wenfeng Feng *,‡, **Xin Zhang** ‡, **Qiushuang Song and Guoying Sun**

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China
* Correspondence: fengwf@hpu.edu.cn; Tel.: +86-139-3911-9371
‡ These authors contributed equally to this work.

Abstract: Although neural network architectures are critical for their performance, how the structural characteristics of a neural network affect its performance has still not been fully explored. We here map architectures of neural network to directed acyclic graphs, and find that incoherence, a structural characteristic to measure the order of directed acyclic graphs, is a good indicator for the performance of corresponding neural networks. Therefore we propose a deep isotropic neural network architecture by folding a chain of same blocks then connecting the blocks with skip connections at different distances. Our models, named FoldNet, have two distinguishing features compared with traditional residual neural netowrks. First, the distances between block pairs connected by skip connections increase from always equal to one to specially selected different values, which lead to more incoherent graphs and let the neural network explore larger receptive fields and thus enhance its multi-scale representation ability. Second, the number of direct paths increases from one to multiple, which leads to a larger proportion of shorter paths and thus improve the direct propagation of information throughout the entire network. Image classification results on CIFAR-10 and Tiny ImageNet benchmarks suggested that our new network architecture performs better than traditional residual neural networks.

Keywords: complex network; neural network architecture; isotropic architecture; image classification

1. Introduction

An artificial neural network is a computing system consisting of many simple, highly interconnected processing elements, i.e. neurons, which process information by their dynamic state response to external inputs [1]. How the neurons are connected is believed to be crucial for the performance of artificial neural networks.

Recent advances in computer vision models have partially confirmed such hyperthesis. For example, the effectiveness of ResNet [2,3] and DenseNet [4] is largely due to the skip connections between blocks; the performance of the learned architectures in neural architecture search is also largely due to their connection structures [5–8].

In spite of the architecture of neural networks is critically important, there is still no consistent way to model it till now. This makes it impossible to theoretically measure the effect of network architectures on their performance, and also makes the design of neural network architectures basically based on intuition try and error. Even if the recent architectures learned by automatically searching in large architecture space are also results of try and error methods [5–9].

On the other hand, the theory of complex networks has been used to model networked systems for decades [10]. If we consider neural networks as networked systems, we can use the theory of complex networks to model neural networks and characterize the effect of network architectures on their performance. Recently, Testolin et al. [11] explain deep belief neural networks by techniques in the field of complex networks; Xie et al. [12] show the efficiency of the neural network structures that are randomly generated by three classical random graph models, i.e. the Erdős-Rényi (ER) [13], Barabási-Albert (BA) [14], and Watts-Strogatz (WS) [15] models.

We here first map the architectures of residual neural networks to directed acyclic graphs then explore the incoherence of DAGs. We find that the incoherence parameter q increases with both the depth of residual neural networks and the folding length d (explained in subsection 3.2). We also find that the proportion of shorter paths in DAGs increases with the folding length d.

Therefore we **fold** the chain-like architecture of ResNet to form an accordion-like neural network architecture, named FoldNet. The new network has multiple direct paths across the whole network, compared with one direct path in traditional residual networks. It also has a higher degree of disorder and a larger proportion of shorter paths in the corresponding DAG. We experimentally show that these structural features of FoldNet let it explore extremely deep network and lead to high performance.

2. Related work

2.1. Isotropic architectures of neural network

The exploration of network architectures has been a part of neural network research since their initial discovery. In computer vision, the architecture of convolutional neural networks has been explored from their depth [2–4,16], width [17], cardinality [18], etc. Their building blocks also extended from residual blocks [2,3,17,18] to many variants of efficient blocks [19–23], such as the depthwise separable convolutional block, etc.

Recently, a new paradigm of isotropic architectures of neural network have emerged partially inspired by the state-of-the-art attention-based transformer architectures in vision [24,25]. Contrary to pyramid-shaped architectures, isotropic architectures have equal size and shape for all elements throughout the network. In isotropic neural networks, images is first divided into sequences of patches, which are then passed into a chain of repeated same blocks.

The blocks of isotropic architectures are divided into three categories depending on their inner operations: attention-based blocks [24,25], CNN-based blocks [26,27] and MLP-based blocks [28,29]. We here focus on CNN-based blocks and leave attention-based blocks and MLP-based blocks for the future work.

2.2. Degree of order of DAGs: trophic coherence

Directed Acyclic Graphs (DAGs) is a representation of partially ordered sets [30]. The extent to which the nodes of a DAG are organized in levels can be measured by *trophic coherence*, a parameter that is originally defined in food webs and then shown to be closely related to many structural and dynamical aspects of complex systems [31–33].

For a directed acyclic graph given by $n \times n$ adjacency matrix A, with elements $a_{ij} = 1$ if there is a directed edge from node i to node j, and $a_{ij} = 0$ if not. The in- and out-degrees of node i are $k_i^{in} = \sum_j a_{ji}$ and $k_i^{out} = \sum_j a_{ij}$, respectively. The first node ($i = 1$) can never have ingoing edges, thus $k_1^{in} = 0$. Similarly, the last node ($i = n$) can never have outgoing edges, thus $k_n^{out} = 0$.

The trophic level s_i of node i is defined as

$$s_i = 1 + \frac{1}{k_i^{in}} \sum_j a_{ji} s_j, \tag{1}$$

if $k_i^{in} > 0$, or $s_i = 1$ if $k_i^{in} = 0$. In other words, the trophic level of the first node is $s = 1$ as it has no incoming edge, while other nodes are assigned the average trophic level of their in-neighbors, plus one. Thus, for any DAG, the trophic level of each node can be easily obtained by solving the linear system of Eq. 1.

Johnson et al. [31] characterize each edge in an DAG with a trophic distance: $x_{ij} = s_i - s_j$. They then consider the distribution of trophic distances over the network, $p(x)$. The homogeneity of $p(x)$ is called trophic coherence: the more similar the trophic distances of all the edges, the more coherent the network. They measure the degree of coherence with the standard deviation of $p(x)$, which is referred to as an *incoherence parameter*: $q = \sqrt{\langle x^2 \rangle - 1}$.

We map architectures of neural networks to directed acyclic graphs, measure the degree of order of directed acyclic graphs using incoherence parameter q, then explore the relationship between the performance of neural networks on image classification and the incoherence of corresponding directed acyclic graphs.

2.3. Effective paths in neural networks

Veit et al. [34] interpreted residual networks as a collection of many paths of differing lengths. The gradient magnitude of a path decreases exponentially with the number of blocks it went through in the backward pass. The total gradient magnitude contributed by paths of each length can be calculated by multiplying the number of paths with that length, and the expected gradient magnitude of the paths with the same length. Thus most of the total gradient magnitude is contributed by paths of shorter length even though they constitute only a tiny part of all paths through the network. These shorter paths are called *effective paths*. The larger the proportion of effective paths, the better performance, with other conditions unchanged.

We find that more incoherent directed acyclic graphs have larger proportion of shorter paths, which improve the direct propagation of information throughout the whole network.

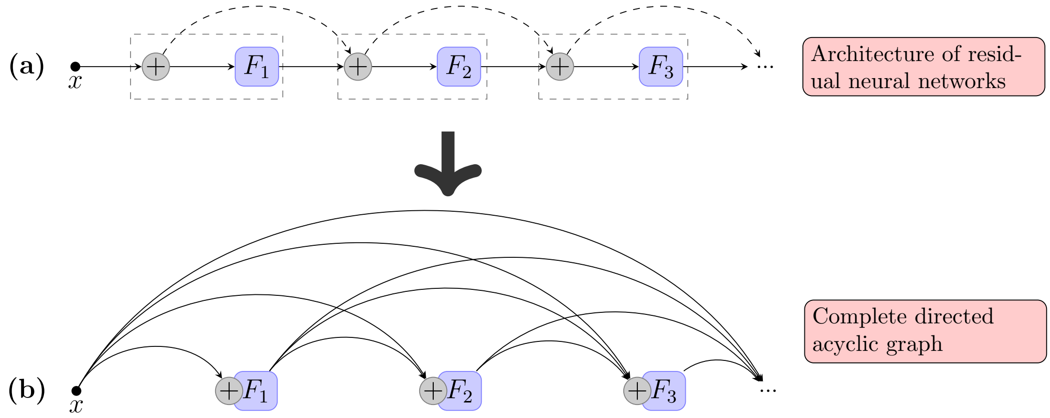
3. FoldNet

3.1. Mapping residual neural network architectures to directed acyclic graphs

In order to evaluate the effect of structural characteristics of neural networks on their performance, we first need to map the architectures of neural networks to directed acyclic graphs. The mapping from the architectures of neural networks to general graphs is flexible. We here intentionally chose a simple mapping, i.e. nodes in graphs represent non-linear transformations among data, while directed edges in graphs represent data flows which send data from one node to another node. Such mapping separates the effect of network structure on performance from the effect of non-linear transformations on performance, since all the weights in neural networks are mapped to the nodes of graphs while all the connection structures are mapped to the edges of graphs.

Consider a batch of images $\mathbf{x}$ that is passed through a residual convolutional neural network. The network comprises L layers, each of which implements a non-linear transformation $F_l(\cdot)$, where $l \geq 1$ indexes the layer. $F_l(\cdot)$ can be a composite function of operations such as batch normalization (BN), rectified linear units (ReLU), pooling, or convolution (Conv) [35,36]. Residual neural networks [2,3] have a skip connection for every layer that bypasses the non-linear transformations with an identity function. Fig. 1a outlines the network structure, where all the dashed lines representing skip connections form the direct path. The skip connections in residual neural networks allow the forward activations and the backward gradients to flow directly through the identity function without information loss, that is the origin of their high performance.

Under the above mapping rule, the architecture of residual neural networks is mapped to a complete directed acyclic graph (Fig. 1b).



Architecture of residual neural networks

Complete directed acyclic graph

Figure 1. Example of mapping from residual neural networks to directed acyclic graphs. (**a**) An example of the architecture of residual neural networks. The F_i nodes represent non-linear transformations among data, the circles with plus signs inside represent summation on all ingoing data. (**b**) The complete directed acyclic graph mapped from the residual neural network. The nodes are composition of summation and non-linear transformations, the lines represent data flows among nodes.

3.2. *Improving incoherence of DAGs by folding residual neural networks*

We observe that all the skip connections in residual neural networks only connect adjacent layers, i.e. the distances between any two layers connected by skip connections always equal to one, that may restrict its represent capability. Thus we **fold** the backbone chain of residual neural networks back and forth to form an accordion-like architecture, as shown in Fig. 2a, 2b.

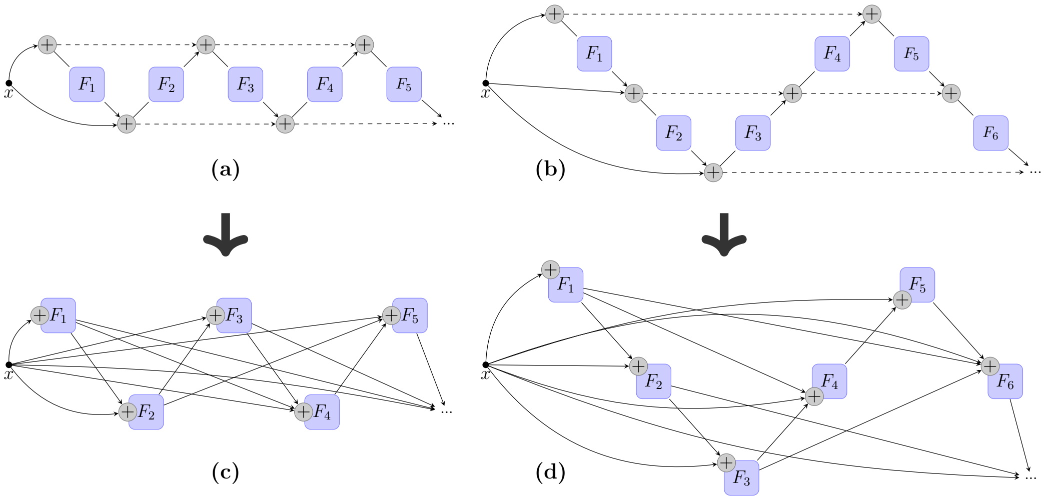


Figure 2. Example of mapping from FoldNet to directed acyclic graphs. (**a**) An example of FoldNet-2. (**b**) An example of FoldNet-3. The F_i nodes represent non-linear transformations among data, the circles with plus signs inside represent summation on all ingoing data. (**c**) The directed acyclic graph mapped from FoldNet-2. (**d**) The directed acyclic graph mapped from FoldNet-3. The nodes are composition of summation and non-linear transformations, the lines represent data flows among nodes.

Such an accordion-like structure extend the chain-like structure of residual neural networks from two aspects. First, the number of direct paths increases from one to multiple, while the particular number of direct paths is determined by the so-called "folding length". Second, the distances between layers connected by skip connections are different with each other, while the particular values of distances are also determined by the so-called "folding length". For example, in Fig. 2b where the "folding length" equal to 3, so there are 3 direct paths, the distances between layers connected by skip connections equal to 2 or 4. Thus we incorporate a new control parameter d to represent the folding length. For convenience, we name such a folded neural network as FoldNet-d where d is the folding length. In FoldNet-d, the number of direct paths equal to d, the distances of skip connections are integers in the set $[2, 4, \ldots, 2(d-1)]$. When $d = 1$, the model degenerated to the traditional residual neural networks. Fig. 1a, Fig. 2a, Fig. 2b illustrated the architectures of FoldNet-1, FoldNet-2, FoldNet-3 respectively.

According to the mapping rule of the previous subsection, FoldNet-2 and FoldNet-3 could be mapped to directed acyclic graphs as shown in (Fig. 2). We next explore the incoherence and path lengths of DAGs. As shown in the main plot in Fig. 3, We find that the incoherence parameter q increases with the number of nodes in DAGs, which equal to the number of layers (or depth) of corresponding neural networks. We also find that the incoherence parameter q increases with the folding length d. The inset plot in Fig. 3 shows the cumulative distribution function (CDF) of path lengths in DAGs when the number of nodes equal to 50. The inset plot indicate that the proportion of shorter paths increases with the folding length d.

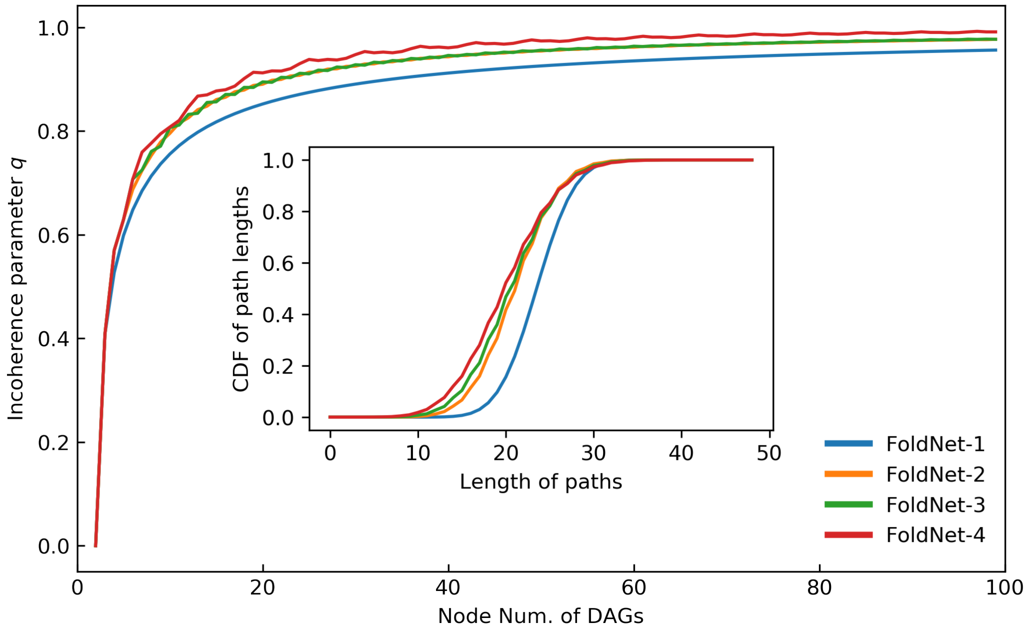


Figure 3. Incoherence and path lengths of DAGs. The main plot illustrates the relationship between incoherence parameter q and number of nodes of DAGs and folding length d. The inset plot show the relationship between path lengths and folding length d.

The comparison of incoherence and path lengths between FoldNet-d where $d \in [2, 3, 4]$ and traditional residual neural networks where $d = 1$ show that FoldNet-d have a higher degree of disorder and a larger proportion of shorter paths, and we argue that these two features together bring better performance of FoldNet-d.

3.3. Architecture design

FoldNet model can be formally expressed by the following equation:

$$\mathbf{x}_l = F_l(\mathbf{x}_{l-1}) + \mathbf{x}_{l-i}, \tag{2}$$

where the output of the current layer $l \geq 1$, $\mathbf{x}_l$, equal to the summation of the non-linear transformation of the output of the previous layer $F_l(\mathbf{x}_{l-1})$ and the output of a previous layer $l - i$, $\mathbf{x}_{l-i}$. i is the distance between the current layer l and a previous layer $l - i$ which is connected to the current layer by a skip connection. The distance i is determined by the current layer index l and the folding length d. It should be noted that if $d = 1$, then $i = 1$, where FoldNet is exactly same with the traditional residual neural networks. For the case of the folding length $d > 1$, if the current layer index is less than the folding length, $l < d$, then the previous layer $\mathbf{x}_{l-i}$ is always equal to $\mathbf{x}_0$. Otherwise, the distance i is determined by:

$$i = 2(1 + (l-1) \bmod (d-1)), \quad \text{when} \ \ d > 1 \wedge l \geq d. \tag{3}$$

The distances of skip connections i are constant and always equal to one in traditional residual networks, while in FoldNet, they are variable values determined by the current layer index l and the folding length d using equation 3. The variable distances allow the model to merge and fuse a larger number of previous images that have different sizes of receptive fields, and thus enhance its multi-scale representation ability.

As illustrated in Fig. 4a, FoldNet consists of a patch embedding layer followed by repeated applications of a folding block. After many applications of this block, we perform global average pooling to get a feature vector which are then passed to a linear classifier and a softmax function to predict the probabilities of all classes.

The patch embedding layer with patch size p and hidden dimension h can be implemented as convolution with c_{in} input channels (equal to 3 for RGB images), h output channels, kernel size p and stride p.

The folding block includes $d-1$ non-linear transformations F_i as shown in the red dashed rectangles in Fig. 4a. Each non-linear transformations F_i itself consists of depthwise convolution followed by pointwise convolution, and each of the convolutions is followed by an activation GELU and post-activation BatchNorm, as illustrated in Fig. 4b. The depthwise convolution is grouped convolution with kernel size $k \times k$ and groups equal to the number of channels h, the pointwise convolution is convolution with kernel size 1×1.

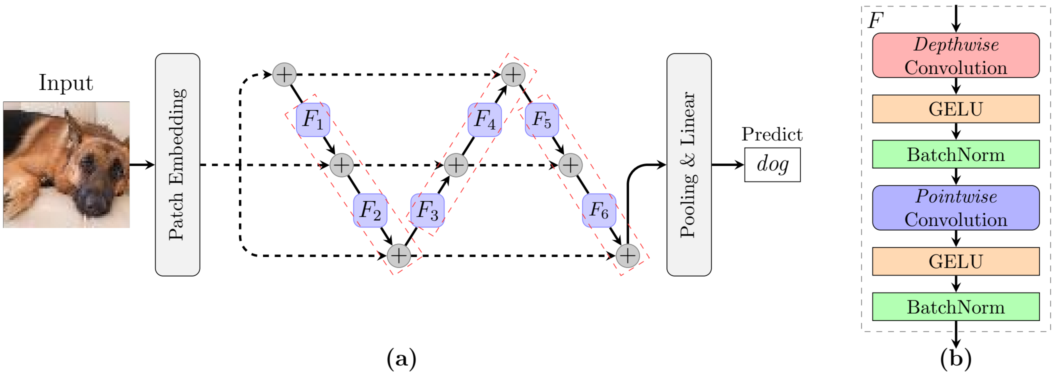


Figure 4. (a) Architecture of the FoldNet model. FoldNet starts with the patch embedding layer, continues with multiple folding blocks shown by the red dashed rectangles, then followed by the pooling and the linear softmax classifier. Here, the depth $n = 6$, the folding length $d = 3$, and the number of folding blocks equal to $n/(d-1) = 3$. (b) Detail of the non-linear transformations F, including a depthwise convolution followed by an GELU activation and a post-activation BatchNorm, after that followed by a pointwise convolution, another GELU activation, and another post-activation BatchNorm.

Therefore, the architecture of FoldNet is mainly determined by five hyper-parameters: (1) the "width" or hidden dimension h, (2) the depth n or the number of repetitions of non-linear transformation F, (3) the folding length d or the number of non-linear transformations per block, (4) the patch size p which controls the internal resolution of the model, (5) the kernel size k of the depthwise convolution.

4. Experiments

4.1. Experimental setup

We train for image classification on the CIFAR-10 and Tiny ImageNet datasets. The CIFAR-10 dataset consists of colored natural images with 32×32 pixels drawn from 10 classes. The training and test sets contain 50,000 and 10,000 images respectively. The Tiny ImageNet dataset is a modified subset of the original ImageNet dataset. It consists of colored natural images with 64 × 64 pixels drawn from 200 different classes instead of 1000 classes in ImageNet dataset. The training and test sets contain 100,000 examples and 10,000 examples respectively.

We implement FoldNet using the Pytorch framework, and evaluate it using the Pytorch Lightning library. We use the free online P100 GPU provided by Kaggle Kernels to train and evaluate our models on image classification. Kaggle Kernels implement a limit on each user's GPU use of 30 hours/week and 10 hours/session. We also use the free and paid online GPUs provided by paperspace.com when the free GPU of Kaggle Kernels couldn't fulfill our requirement on GPUs.

Due to our limited compute, we only consider hyper-parameters that are critical for the performance of FoldNet, and keep all other hyper-parameters constant. FoldNet only changes the connecting way of skip connections among the layers in residual neural networks, and is a macro design methodology of neural network architectures. Thus we focus on the depth n and the folding length d that reflect the macro design of FoldNet, while keep the hidden dimension h, the patch size p and the kernel size k that reflect the micro design in the layer level of FoldNet, their optimized values. We set the patch size $p = 2$ and the kernel size $k = 5$ suggested in the related isotropic model ConvMixer [27]. We set the hidden dimension $h \in \{64, 256\}$.

For CIFAR-10 dataset, we train FoldNet for 100 epochs with a batch size of 256. For Tiny ImageNet dataset, due to our limited compute, we train FoldNet for 50 epochs with a batch size of 128. For both CIFAR-10 and Tiny ImageNet, we use AdamW [37] with a learning rate of 1e-2 and a weight decay of 0.1. There is a 10-epoch linear warmup with initial learning rate of 1e-5 and a cosine decaying schedule afterward. For data augmentation, we include the RandomResizedCrop, RandomHorizontalFlip, RandAugment[38] and ColorJitter.

4.2. Experimental Results of CIFAR-10

In order to evaluate the effect of the depth n and the folding length d of FoldNet on its performance on image classification, we evaluate the depths in a sequence $[16, 24, 32, 40, 48]$, and for each depth n, we evaluate the folding length in a sequence $[1, 2, 3, 4]$. There are totally 24 evaluations as listed in Tab. 1. For each evaluation, we run 3 times, and report the mean value of the maximum validation accuracy of 3-runnings as the performance measurement.

Table 1. Hyper-parameter values for CIFAR-10 dataset. The depth n equal to the number of folding blocks times $d - 1$. The patch size p and kernel size k are fixed as $p = 2$ and $k = 5$. The hidden dimension h is chosen from the set [64, 256].

Folding length d	**Num. of folding blocks**	**Corresponding depth n**
1	[16, 24, 32, 40, 48]	[16, 24, 32, 40, 48]
2	[16, 24, 32, 40, 48]	[16, 24, 32, 40, 48]
3	[8, 9, 12, 13, 16, 17, 24]	[16, 18, 24, 26, 32, 34, 48]
4	[5, 7, 9, 11, 13, 15, 16]	[15, 21, 27, 33, 39, 45, 48]

Figure 5 depicts the validation accuracy of FoldNet-1, FoldNet-2, FoldNet-3 and FoldNet-4 when hidden dimension $h = 64$. As shown in the figure, the performance of all the FoldNet models increase with the depth n of FoldNet, and FoldNet-2, FoldNet-3 and FoldNet-4 where $d > 1$ perform better than FoldNet-1 where $d = 1$ at all the depths. As we have shown in Fig. 3 that the incoherence of DAGs is strong positive correlated with the depth n and folding length d of the corresponding neural networks, therefore we could infer the strong positive correlation between the incoherence of DAGs and the classification accuracy of the corresponding neural networks. In particular, FoldNet-2 with depth $n = 48$ has 0.293M parameters and can achieve 93.95% top-1 accuracy on CIFAR-10 after 100 epochs which increase by 0.43% compared with FoldNet-1 with depth $n = 48$. We also show the validation accuracy curves of FoldNet-1, FoldNet-2, FoldNet-3 and FoldNet-4 when depth $n = 48$ in Fig. 6 to compare their performance in detail.

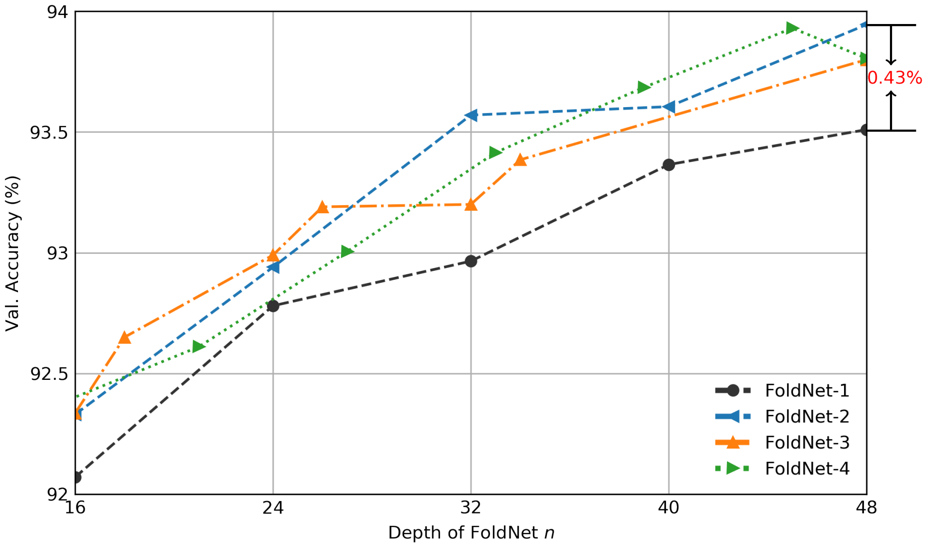


Figure 5. Validation accuracy of FoldNet-d for CIFAR-10 dataset when hidden dimension $h = 64$. x axis is the depth of neural network n, y axis is the validation accuracy percentage. The validation accuracy of FoldNet-2 increase by 0.43% compared with FoldNet-1 when depth $n = 48$.

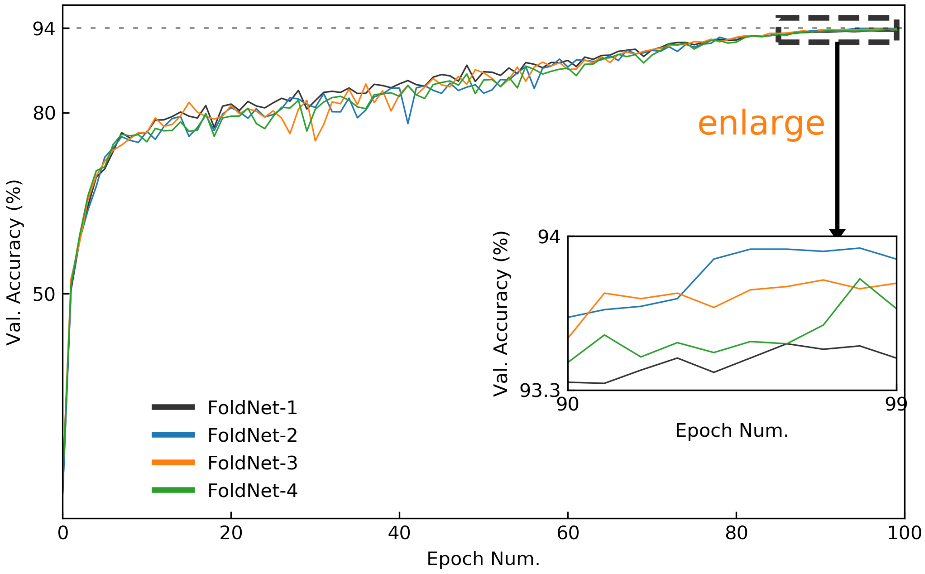


Figure 6. Validation accuracy curves of FoldNet for CIFAR-10 dataset when hidden dimension $h = 64$ and network depth $n = 48$. The validation accuracy curves of the last 10 epochs are enlarged to compare accuracy of FoldNet-d more clearly.

We also show the validation accuracy of FoldNet-1, FoldNet-2, FoldNet-3 and FoldNet-4 when hidden dimension $h = 256$ in Fig. 7. Similar to the case of $h = 64$ in Fig. 5, the performance of all the FoldNet models increase with the depth n of FoldNet, and FoldNet-2, FoldNet-3 and FoldNet-4 where $d > 1$ perform better than FoldNet-1 where $d = 1$ at

doi:10.20944/preprints202210.0092.v1

almost all the depths. In particular, FoldNet-2 with depth $n = 48$ has 3.5M parameters and can achieve 95.56% top-1 accuracy on CIFAR-10 after 100 epochs which increase by 0.25% compared with FoldNet-1 with depth $n = 48$. We also show the validation accuracy curves of FoldNet-1, FoldNet-2, FoldNet-3 and FoldNet-4 when depth $n = 48$ in Fig. 8 to compare their performance in detail. 236 237 238 239 240

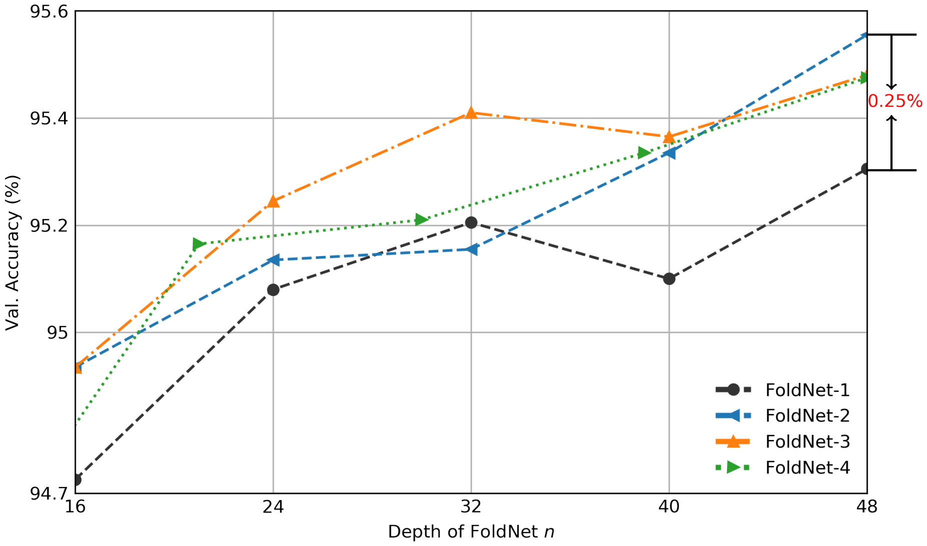


Figure 7. Validation accuracy of FoldNet-d for CIFAR-10 dataset when hidden dimension $h = 256$. x axis is the depth of neural network n, y axis is the validation accuracy percentage. The validation accuracy of FoldNet-2 increase by 0.25% compared with FoldNet-1 when depth $n = 48$.

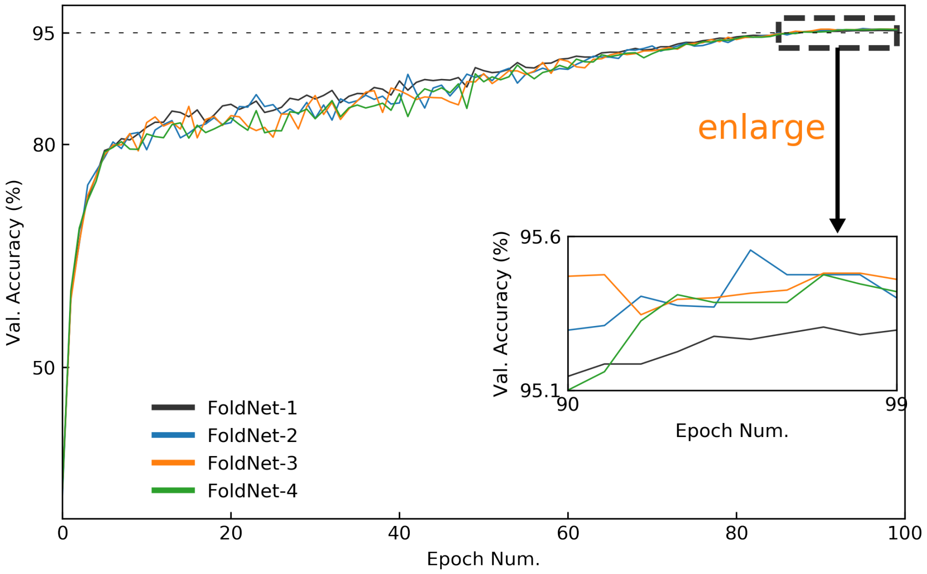


Figure 8. Validation accuracy curves of FoldNet for CIFAR-10 dataset when hidden dimension $h = 256$ and network depth $n = 48$. The validation accuracy curves of the last 10 epochs are enlarged to compare accuracy of FoldNet-d more clearly.

4.3. Experimental Results of Tiny ImageNet 241

Due to our limited compute, we only evaluate the performance of FoldNet-1, FoldNet-2, FoldNet-3 and FoldNet-4 when hidden dimension $h = 256$ and network depth $n = 32$. As shown in Fig. 9, FoldNet-3 has 2.4M parameters and can achieve 67.55% top-1 accuracy on Tiny ImageNet after only 50 epochs which increase by 0.55% compared with FoldNet-1. 242 243 244 245

We further explore the performance of FoldNet-3 by increasing the hidden dimension to $h = 512$ and the depth to $n = 48$ and training for 85 epochs. Such a FoldNet-3 has 13.7M parameters and can achieve 70.13% top-1 accuracy on Tiny ImageNet after 85 epochs as shown in Fig. 10. This result is competitive compared with the state-of-art results on Tiny ImageNet [39], which achieved 70.24% top-1 accuracy on Tiny ImageNet using a ResNet-like model with 100.5M parameters after 300 epochs. 246 247 248 249 250 251

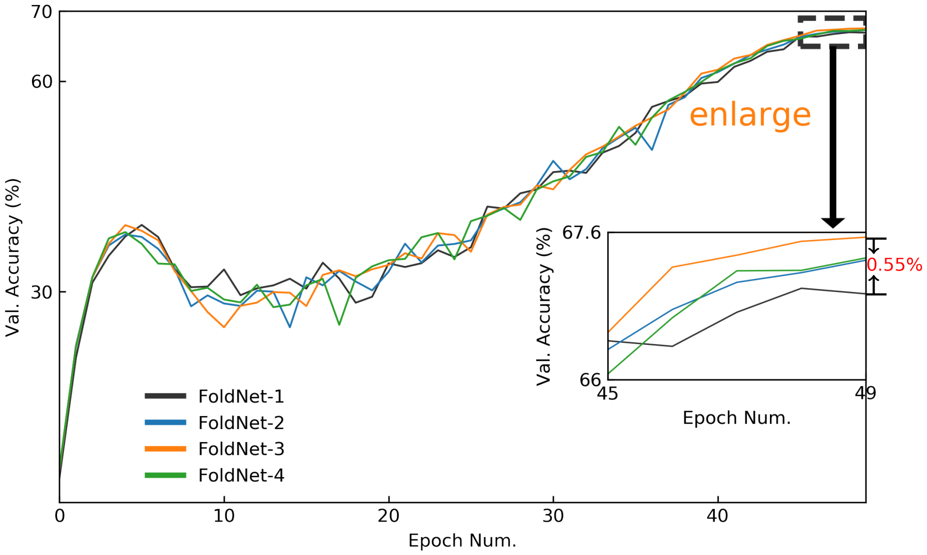


Figure 9. Validation accuracy curves of FoldNet for Tiny ImageNet dataset when hidden dimension $h = 256$ and network depth $n = 32$. The validation accuracy curves of the last 5 epochs are enlarged to compare accuracy of FoldNet-d more clearly.

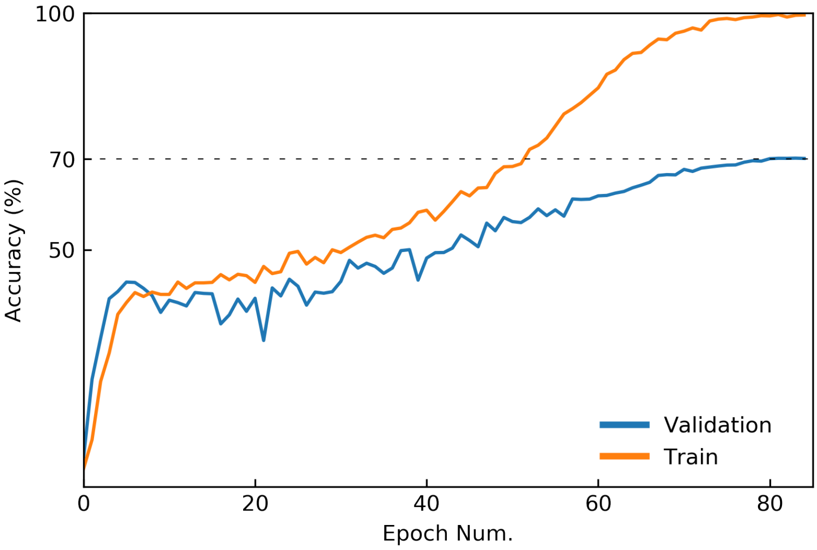


Figure 10. Accuracy curves of FoldNet-3 for Tiny ImageNet dataset when hidden dimension $h = 512$ and network depth $n = 48$.

5. Discussion 252

In this paper, we attempt to apply the insights from the field of complex networks about the structural features of a network affecting its dynamics to deep neural networks. To this end, we map the architectures of deep neural networks to directed acyclic graphs (DAGs), then find out the relationship between the structural characteristics of neural 253 254 255 256 networks and corresponding DAGs. We found a strong positive corelation between the incoherence of DAGs and the depth n and folding length d of corresponding neural netowrks. Thus, we propose a deep isotropic neural network architecture FoldNet by folding a chain of same blocks whose corresponding DAGs are more incoherent.

We evaluate the effect of FoldNet on image classification by varying their depth n and folding length d. We found a positive correlation between the depth and folding length of FoldNet and their accuracy. Therefore, we infer that the incoherence of DAGs has a positive corresponding with the accuracy of the corresponding neural netowrks on image classification. FoldNet achieves the competitive results on Tiny ImageNet dataset with much less parameters.

We recognize that the performance of a neural network may be affected by multiple structural features at the same time, rather than one, for example incoherence in our case. DAGs have other structural features, such as the number of paths in DAG, that can affect the performance of the corresponding neural netowrks. Our future work will explore in this direction.

Author Contributions: Conceptualization, W.F. and X.Z.; methodology and X.Z., W.F.; software, W.F.; validation, Q.S. and G.S.; formal analysis, W.F.; investigation, W.F.; resources, Q.S. and G.S.; data curation, Q.S. and G.S.; writing—original draft preparation, W.F. and X.Z.; writing—review and editing, X.Z.; visualization, Q.S. and G.S.; supervision, W.F.; project administration, W.F.; funding acquisition, W.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program of New Century Excellent Talents in University of China (no. NCET-11-0942) and the Program of National Natural Science Foundation of China (no. 60703053).

Data Availability Statement: The CIFAR-10 dataset refered to in this study are openly available in "Learning multiple layers of features from tiny images" at https://www.cs.toronto.edu/ kriz/cifar.html, accessed on 9 April 2022; the Tiny ImageNet dataset refered to in this study are openly available at https://www.kaggle.com/c/tiny-imagenet, accessed on 9 April 2022.

Acknowledgments: We gratefully acknowledge the P100 GPU support provided by kaggle.com and the A4000, A5000 GPU support provided by paperspace.com.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DAG	Directed Acyclic Graphs
s_i	trophic level of node i in DAG
A	adjacency matrix of DAG
k_i^{in}	in-degrees of node i in DAG
k_i^{out}	out-degrees of node i in DAG
q	incoherence parameter of DAG
$\mathbf{x}_l$	output of layer l in FoldNet
F_l	non-linear transformations of layer l in FoldNet
d	folding length of FoldNet
h	hidden dimension of FoldNet
n	number of non-linear transformations F in FoldNet
p	patch size
k	kernel size of depthwise convolution
CNN	Convolutional Neural Network
MLP	Multiple Layer Network

References

1. Caudill, M. Neural Networks Primer, Part I. *AI Expert* **1987**, *2*, 46–52.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]* **2015**. arXiv: 1512.03385.

3. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the Computer Vision – ECCV 2016. Springer, Cham, 2016, Lecture Notes in Computer Science, pp. 630–645.
4. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv:1608.06993 [cs]* **2016**. arXiv: 1608.06993.
5. Li, L.; Talwalkar, A. Random Search and Reproducibility for Neural Architecture Search. *arXiv:1902.07638 [cs, stat]* **2019**. arXiv: 1902.07638.
6. Pham, H.; Guan, M.Y.; Zoph, B.; Le, Q.V.; Dean, J. Efficient Neural Architecture Search via Parameter Sharing. *arXiv:1802.03268 [cs, stat]* **2018**. arXiv: 1802.03268.
7. Sciuto, C.; Yu, K.; Jaggi, M.; Musat, C.; Salzmann, M. Evaluating the Search Phase of Neural Architecture Search. *arXiv:1902.08142 [cs, stat]* **2019**. arXiv: 1902.08142.
8. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. *arXiv:1707.07012 [cs, stat]* **2017**. arXiv: 1707.07012.
9. Zoph, B.; Le, Q.V. Neural Architecture Search with Reinforcement Learning. *arXiv:1611.01578 [cs]* **2016**. arXiv: 1611.01578.
10. Newman, M. *Networks: An Introduction*; Oxford University Press, Inc.: New York, NY, USA, 2010.
11. Testolin, A.; Piccolini, M.; Suweis, S. Deep learning systems as complex networks **2018**.
12. Xie, S.; Kirillov, A.; Girshick, R.; He, K. Exploring Randomly Wired Neural Networks for Image Recognition **2019**.
13. Erdos, P.; Rényi, A.; et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **1960**, *5*, 17–60.
14. Albert, R.; Barabási, A.L. Statistical mechanics of complex networks. *Reviews of modern physics* **2002**, *74*, 47.
15. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world'networks. *nature* **1998**, *393*, 440–442.
16. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* **2014**. arXiv: 1409.1556.
17. Zagoruyko, S.; Komodakis, N. Wide Residual Networks **2016**.
18. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv:1611.05431 [cs]* **2017**. arXiv: 1611.05431.
19. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv:1610.02357 [cs]* **2016**. arXiv: 1610.02357.
20. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv:1905.11946 [cs, stat]* **2019**. arXiv: 1905.11946.
21. 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:1704.04861 [cs]* **2017**. arXiv: 1704.04861.
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv:1801.04381 [cs]* **2018**. arXiv: 1801.04381.
23. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning **2016**.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
25. 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**.
26. Sandler, M.; Baccash, J.; Zhmoginov, A.; Howard, A. Non-discriminative data or weak model? on the relative importance of data and model resolution. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.
27. Trockman, A.; Kolter, J.Z. Patches are all you need? *arXiv preprint arXiv:2201.09792* **2022**.
28. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems* **2021**, *34*, 24261–24272.
29. 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. *arXiv preprint arXiv:2105.03404* **2021**.
30. Karrer, B.; Newman, M.E.J. Random graph models for directed acyclic networks. *Physical Review E* **2009**, *80*. arXiv: 0907.4346, https://doi.org/10.1103/PhysRevE.80.046110.
31. Johnson, S.; Domínguez-García, V.; Donetti, L.; Muñoz, M.A. Trophic coherence determines food-web stability. *arXiv:1404.7728 [cond-mat, q-bio]* **2014**. arXiv: 1404.7728, https://doi.org/10.1073/pnas.1409077111.
32. Domínguez-García, V.; Johnson, S.; Muñoz, M.A. Intervality and coherence in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **2016**, *26*, 065308. arXiv: 1603.03767, https://doi.org/10.1063/1.4953163.
33. Klaise, J.; Johnson, S. From neurons to epidemics: How trophic coherence affects spreading processes. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **2016**, *26*, 065310. arXiv: 1603.00670, https://doi.org/10.1063/1.4953160.
34. Veit, A.; Wilber, M.; Belongie, S. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. In Proceedings of the Proceedings of the 30th International Conference on Neural Information Processing Systems; Curran Associates Inc.: USA, 2016; NIPS'16, pp. 550–558.
35. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* **2015**. arXiv: 1502.03167.

36. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324. https://doi.org/10.1109/5.726791.
37. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* **2017**.
38. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 702–703.
39. Ramé, A.; Sun, R.; Cord, M. Mixmo: Mixing multiple inputs for multiple outputs via deep subnetworks. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 823–833.