Augmented Shortcuts for Vision Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer models have achieved great progress on computer vision tasks recently. The rapid development of vision transformers is mainly contributed by their high representation ability for extracting informative features from input images. However, the mainstream transformer models are designed with deep architectures, and the feature diversity will be continuously reduced as the depth increases, *i.e.*, feature collapse. In this paper, we theoretically analyze the feature collapse phenomenon and study the relationship between shortcuts and feature diversity in these transformer models. Then, we present an augmented shortcut scheme, which inserts additional paths with learnable parameters in parallel on the original shortcuts. To save the computational costs, we further explore an efficient approach that uses the block-circulant projection to implement augmented shortcuts. Extensive experiments conducted on benchmark datasets demonstrate the effectiveness of the proposed method, which brings about 1% accuracy increase of the state-of-the-art visual transformers without obviously increasing their parameters and FLOPs.

1 Introduction

Originating from the natural language processing filed, transformer models [28, 9] have recently made great progress in various computer vision tasks such as image classification [9, 27], object detection [3] and image processing [4]. Wherein, the ViT model [9] divides the input images as visual sequences and obtains an 88.36% top-1 accuracy which is competitive to the SOTA convolutional neural network (CNN) models (*e.g.*, EfficientNet [26]). Compared to CNNs which are usually customized for vision tasks with prior knowledge (*e.g.*, translation equivalence and locality), vision transformers introduce less inductive bias and have a larger potential to achieve better performance on different visual tasks. Besides, considering the high performance of transformers in various fields (*e.g.*, natural language processing and computer vision), we may only need to support transformer for processing different tasks which can significantly simplify the software/hardware design.

Besides the self-attention layers in vision transformers, a shortcut is often included to directly connect multiple layers with identity projection [12, 28, 9]. The introduce of shortcut is motivated by the architecture designs in CNNs, and has been demonstrated to be beneficial for a stable convergence and better performance. There is a series of works on interpreting and understanding the role of shortcut. For example, Balduzzi *et al.* [1] analyze the gradients of deep networks and show that shortcut connection can effectively alleviate the problem of gradient vanishing and exploding. Veit *et al.* [29] reckon ResNet [12] as the ensemble of a collection of paths with different lengths, and the gradient vanishing problem is addressed by short paths. From a theoretical perspective, Liu *et al.* [18] prove that shortcut connection can avoid network parameters trapped by spurious local optimum and help them converge to a global optimum. Apart from CNNs, shortcut connection is consequently

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

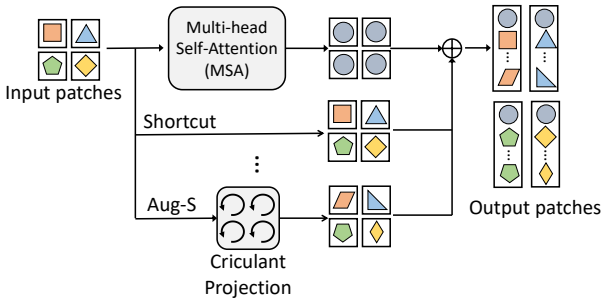


Figure 1: The diagram of MSA module equipped with augmented shortcuts. The original identity shortcut copies input feature while the augmented shortcuts (Aug-S) project features of each input patch to diverse representations.

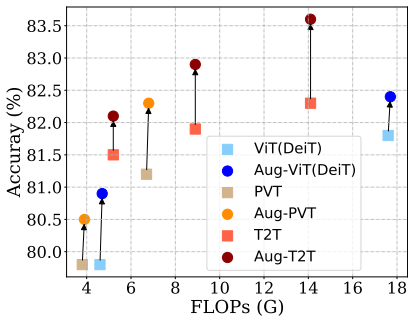


Figure 2: Performance improvement for using the proposed augmented shortcuts on the state-of-the-art vision transformer models. The performances of different models on ImageNet are provided.

widely-used in other deep neural networks such as transformer [28, 9], LSTM [11] and AlphaGo Zero [24].

In vision transformers, the shortcut connection bypasses the multihead self-attention (MSA) and multilayer perceptron (MLP) modules, which plays a critical role towards the success of vision transformers. A transformer without shortcut suffer extremely low performance (Table 1). Empirically, removing the shortcut results in features from different patches becoming indistinguishable as the network going deeper (shown in Figure 3(a)), and such features have limited representation capacity for the downstream prediction. We name this phenomenon as *feature collapse*. Fortunately, adding shortcut in transformers can alleviate the phenomenon (Figure 3(b)) and make the generated features diverse. However, the conventional shortcut simply copies the input feature to the output, limiting its ability for enhancing the feature's diversity.

In this paper, we introduce a novel augmented shortcut scheme for improving feature diversity in vision transformers (Figure 1). Besides the conventional identity shortcut, we propose to parallel the MSA module with multiple parameterized shortcuts, which provide more alternative paths to bypass the attention mechanism. In particular, an augmented shortcut connection is constructed as the sequence of a linear projection with learnable parameters and a nonlinear activation function. We theoretically demonstrate that the transformer model equipped with augmented shortcuts can avoid feature collapse and produce more diverse features. For the efficiency reason, we further replace the original dense matrices with block-circulant matrices, which have lower computational complexity in the Fourier frequency domain while high representation ability in the spatial domain. Owing to the compactness of circulant projection, the parameters and computational costs introduced by the augmented shortcuts are negligible compared to those of the MSA and MLP modules. We empirically evaluate the effectiveness of augmented shortcut with the ViT model [9, 27] and its SOTA variants (PVT [30], T2T [31]) on ImageNet dataset. Equipped with our augmented shortcut, the performances (top-1 accuracies) of these models can be enhanced by about 1% with comparable computational costs (*e.g.*, Figure 2).

2 Preliminaries and Motivation

The vision transformer (ViT) [9] adopts the typical architecture proposed in [28], with MSA and MLP modules alternatively stacked. For an image, ViT splits it into N patches and projects each patch into a d-dimension vector. Thus the input of a vision transformer can be seen as a matrix $Z_0 \in \mathbb{R}^{N\times d}$. An MSA module with H heads is defined as

$$\mathrm{MSA}(Z_l) = \mathrm{Concat}([A_{lh} Z_l W^v_{lh}]_{h=1}^{H}) W^o_l, \tag{1}$$

where $Z_l \in \mathbb{R}^{N\times d}$ is the features of the l-th layer, $A_{lh} \in \mathbb{R}^{N\times N}$ and $W^v_{lh} \in \mathbb{R}^{d\times(d/H)}$ are the corresponding attention map and value projection matrix in the h-th head, respectively. $\mathrm{Concat}(\cdot)$ denotes the concatenating for features of the H heads and $W^o_l \in \mathbb{R}^{d\times d}$ is the output projection matrix. The attention matrix A_{lh} is calculated by the self-attention mechanism, *i.e.*,

$$A_{lh} = \mathrm{softmax}\left(\frac{(Z_l W^q_{lh})(Z_l W^k_{lh})^\top}{\sqrt{d}}\right), \tag{2}$$

where $W^q_{lh} \in \mathbb{R}^{d\times(d/H)}$ and $W^k_{lh} \in \mathbb{R}^{d\times(d/H)}$ are the query and value projection matrices, respectively. Attention A_{lh} reflects the relation between different patches, and a larger value A^{ij}_{lh} indicate that patch i and patch j have a stronger relationship.

The MLP module extracts features from each patch independently, which is usually constructed by stacking two linear projection layers with weights $W^a \in \mathbb{R}^{d\times d_{hidden}}$, $W^b \in \mathbb{R}^{d_{hidden}\times d}$ and non-linear activation function σ, *i.e.*, $\mathrm{MLP}(Z_l) = \sigma(Z_l W^a)W^b$. A vision transformer model is constructed by stacking the MLP and MSA modules alternatively.

Depth	Top1 accuracy (%)
1	18.3
2	43.4
4	0.13
6	0.12
8	0.15
10	0.14
12	0.15

Table 1: Performance of the ViT model without shortcut on ImageNet.

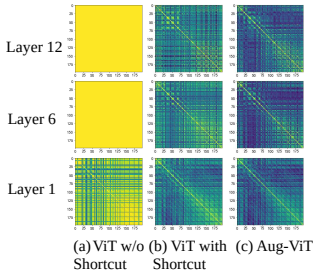


Figure 3: The similarity matrices over patches in the ViT-Base model.

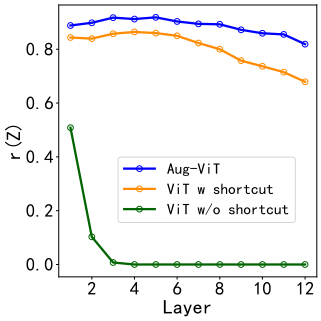


Figure 4: Diversity metric $r(Z_l)$ varies *w.r.t.* layers.

Besides MSA and MLP, the shortcut connection also plays a vital role to achieve high performance. Empirically, removing shortcut severely harms the accuracies of ViT models (as shown in Table 1). To explore the reason behind, we propose to analyze the intermediate features in the ViT-like models. Specially, we calculate the cosine similarity between different patch features and show the similarity matrices of low (Layer 1), middle (Layer 6) and deep layers (Layer 12) in Figure 3. Features from different patches in a layer quickly become indistinguishable as the network depth increasing. We call this phenomenon *feature collapse*, which greatly restrict the representation capacity and then prevent high performance.

The feature collapse mainly results from the attention mechanism, which aggregates features from different patches layer-by-layer. Denoting $\tilde{Z}_{lh}$ as the output feature after the attention map, the attention mechanism can be formulated as:

$$\tilde{Z}^i_{lh} = \sum_{j=1}^{N} A^{ij}_{lh} Z^j_l, \ \text{ s.t. } \sum_{j=1}^{N} A^{ij}_{lh} = 1, i = [1, 2, \cdots, N], \tag{3}$$

where the feature $\tilde{Z}^i_{lh}$ from the i-th patch is the weighted average of features from all other patches, and the weights are values in attention map A_{lh}. Though attention mechanisms capture global relationship, the diversity of patch features are reduced as well. Theoretically, the diversity of a feature Z_l can be measured by the difference between the feature and a rank-1 matrix, *i.e.*,

$$r(Z_l) = \|Z_l - \mathbf{1}\boldsymbol{z}_l^\top\|, \ \text{ where } \boldsymbol{z}_l = \mathrm{argmin}_{\boldsymbol{z}'_l}\|Z_l - \mathbf{1}\boldsymbol{z}_l'^\top\|. \tag{4}$$

where $\|\cdot\|$ denotes the matrix norm[1]. $\boldsymbol{z}_l, \boldsymbol{z}'_l \in \mathbb{R}^d$ are vectors and $\mathbf{1}$ is an all-ones vector. The rank of matrix $\mathbf{1}\boldsymbol{z}_l^\top$ is 1. Intuitively, if $\boldsymbol{z}_l \in \mathbb{R}^d$ can represent $Z_l \in \mathbb{R}^{N\times d}$ of N patches, the generated feature will be redundant. A larger $r(Z_l)$ implies stronger diversity of the given feature Z_l. We use $r(Z_l)$ as the diversity metric in the following. Because MSA module incurs the feature collapse, we focus on the a model stacked by attention modules and have the following theorem [8], *i.e.*,

Theorem 1 *Given a model stacked by the MSA modules, the diversity $r(X_l)$ of feature in the l-th layer can be bounded by that of input data Z_0,* i.e.,

$$r(Z_l) \leq \left(\frac{H\gamma}{\sqrt{d}}\right)^{\frac{3^l-1}{2}} r(Z_0)^{3^l}, \tag{5}$$

where H is number of heads, d is feature dimension and γ is a constant related to the norms of weight matrices in the MSA module.

[1]Here $\|\cdot\|$ is defined as the ℓ_1, ℓ_∞-composite norm for the convenience of theoretical derivation.

$H\gamma/\sqrt{d}$ and $r(Z_0)$ are usually smaller than 1, so the feature diversity $r(Z_l)$ will decrease rapidly as the network depth increases [8]. We also empirically show how $r(Z_l)$ varies *w.r.t.* the network depth in the ViT models (Figure 4), and the empirical results are accordant to the Theorem 1.

Fortunately, adding a shortcut connection parallel to the MSA module can empirically preserve the feature diversity especially in deep layers (see Figure 3(b) and Figure 4). The MSA module with shortcut can be formulated as:

$$\mathrm{ShortcutMSA}(Z_l) = \mathrm{MSA}(Z_l) + Z_l, \tag{6}$$

where the identity projection (*i.e.*, Z_l) is parallel to the MSA module. Intuitively, the shortcut connection bypasses the MSA module and provides another alternative path, where features can be directly delivered to the next layer without interference of other patches. Adding the shortcut connections can also theoretically improve the bound of feature diversity $r(Z_l)$ (as discussed in Section 3.1). The success of shortcut shows that bypassing the attention layers with extra paths is an effective way to enhance feature diversity and improve the performance of transformer-like models.

However, in general vision transformer models (*e.g.*, ViT [9], PVT [30], T2T [31]), there is only a single shortcut connection with identity projection for each MSA module, which only copies the input features to the outputs. This simple formulation may not have enough representation capacity to improve the feature diversity maximally. In the following chapters, we aim to refine the existing shortcut connections in vision transformers and explore efficient but powerful augmented shortcuts to produce visual features with higher diversity.

3 Approach

3.1 Augmented Shortcuts

We propose augmented shortcuts to alleviate the feature collapse by paralleling the original identity shortcut with more parameterized projections. The MSA module equipped with T augmented shortcuts can be formulated as:

$$\mathrm{AugMSA}(Z_l) = \mathrm{MSA}(Z_l) + Z_l + \sum_{i=1}^{T} \mathcal{T}_{li}(Z_l; \Theta_{li}),\ l \in [1, 2, \cdots, L], \tag{7}$$

where $\mathcal{T}_{li}(\cdot)$ is the i-th augmented shortcut connection of the l-th layer and Θ_{li} denotes its parameters. Besides the original shortcut, the augmented shortcuts provide more alternative paths to bypass the attention mechanism. Different from the identity projection directly copying the input patches to the corresponding outputs, the parameterized projection $\mathcal{T}_{li}(\cdot)$ can transform input features into another feature space. Actually, projections $\mathcal{T}_{li}(\cdot)$ will make different transformations on the input feature as long as their weight matrices Θ_{li} are different, and thus paralleling more augmented shortcuts has potential to enrich the feature space.

A simple formulation for $\mathcal{T}_{li}(\cdot)$ is the sequence of a linear projection and an activation function *i.e.*,

$$\mathcal{T}_{li}(Z_l; \Theta_{li}) = \sigma(Z_l \Theta_{li}),\ l \in [1, \cdots, L],\ i \in [1, 2, \cdots, T], \tag{8}$$

where $\Theta_{li} \in \mathbb{R}^{d \times d}$ is the weight matrix and σ is the non-linear activation function (*e.g.*, GELU). In Eq. 8, $\mathcal{T}_{li}(\cdot)$ tackles each patch independently and preserves their specificity, which is complement to the MSA modules aggregating different patches. Note that the identity mapping is a special case of Eq. 8, *i.e.*, $\sigma(x) = x$ and Θ_{li} is the identity matrix.

Recall that in a transformer-like model without shortcut, the upper bound of feature diversity $r(Z_l)$ decreases dramatically as the increase of network depth(Theorem 1). In the following, we analyze how the diversity $r(Z_l)$ changes *w.r.t.* the layer l in the model stacked by the AugMSA modules, and we has the follow theorem.

Theorem 2 *Given a model stacked by the AugMSA modules, the diversity $r(X_l)$ of feature in the l-th layer can be bounded by that of input data Z_0,* i.e.*,*

$$r(Z_l) \leq \max_{0 \leq m \leq l} \left(\frac{H\gamma}{\sqrt{d}}\right)^{\frac{3^m - 1}{2}} (2H\alpha_m)^{3^m(l-m)} r(Z_0)^{3^m}, \tag{9}$$

where $\alpha_m = 1 + \sum_{i=1}^{T} \lambda \|\Theta_{mi}\|$. Θ_{mi} is the weight matrix in the i-th augmented shortcut of the m-th layer, and λ is the Lipschitz constant of activation function $\sigma(\cdot)$.

Compared with Theorem 1, the augmented shortcuts introduce an extra term $(2H\alpha_m)^{3^m(l-m)}$, which will increase doubly exponentially as $2H\alpha_m$ is usually larger than 1. This tends to suppress the diversity decay incurred by attention mechanism. The term α_m ($0 \leq m \leq l$) is determined by the norms of weight matrices Θ_{mi} of the augmented shortcuts in the m-th layer, and then bound of diversity $r(Z_l)$ in the l-th layer can be affected by all the augmented shortcuts in the previous layers. For the ShortcutMSA module (Eq. 6) with only a identity shortcut, we have $\alpha_m = 1$. Adding more augmented shortcuts can increase the magnitude of α_m, which further improves the bound. Detailed proof for Theorem 2 is represented in the supplemental material.

Considering that shortcut connections exist in both MSA and MLP modules, the proposed augmented shortcuts can also be embedded into MLP similarly, *i.e.*,

$$\mathrm{AugMLP}(Z'_l) = \mathrm{MLP}(Z'_l) + Z'_l + \sum_{i=1}^{T} \mathcal{T}_{li}(Z'_l; \Theta'_{li}),\ l \in [1, 2, \cdots, L], \tag{10}$$

where Z'_l is the input feature of the MLP module in the l-th layer and Θ'_{li} denotes the parameters in augmented shortcuts. Paralleling the MLP module with the augmented shortcuts can further improve the diversity, which is analyzed detailedly in the supplemental material. Stacking the AugMSA and AugMLP modules constructs the *Aug-ViT* model, whose feature has stronger diversity as shown in Figure 3(c) and Figure 4.

3.2 Efficient Implementation via Circulant Projection

As discussed above, Paralleling multiple augmented shortcuts with the MSA and MLP modules in a vision transformer can improve the feature diversity for higher performance. However, directly implementing $\mathcal{T}_{li}(\cdot)$ (Eq. 8) involves a lot of matrix multiplications, which are computationally expensive. For example, given feature $Z_l \in \mathbb{R}^{n\times d}$ and weight matrix $\Theta_{li} \in \mathbb{R}^{d\times d}$, the matrix multiplication $Z_l\Theta_{li}$ consumes nd^2 FLOPs, where d is usually large in vision transformers (*e.g.*, 798 in ViT-B). Therefore, we propose to implement the augmented shortcuts with block-circulant matrices, whose computational costs are negligible compared to other modules in a vision transformer. In the following, we omit the subscripts l and i indicating layers and paths for brevity.

Circulant matrix [15, 7] is a typical structured matrix with extremely few parameters and low computational complexity on the Fourier domain. A circulant matrix $C \in \mathbb{R}^{d'\times d'}$ only has d' parameters and the product between C and a vector only has $\mathcal{O}(d' \log d')$ computational complexity via the fast Fourier transformation (FFT). The projection with circulant matrix also has theoretical guarantee to be a good approximation of unstructured projection and preserve the critical projection properties such as ℓ_2 distance and angles [13, 10]. Consequently, we take advantage of circulant matrices to implement augmented shortcuts. Specially, the original matrix Θ is split into b^2 sub-matrices $C^{ij} \in \mathbb{R}^{d'\times d'}$, *i.e.*,

$$\Theta = \begin{bmatrix} C^{11} & C^{12} & \cdots & C^{1b} \\ C^{21} & C^{22} & \cdots & C^{2b} \\ \vdots & \vdots & \vdots & \vdots \\ C^{b1} & C^{b2} & \cdots & C^{bb} \end{bmatrix}, \tag{11}$$

where d' is the size of sub-matrices and $bd' = d$. Each sub-matrix C^{ij} is a circulant matrix generated by circulating the elements in a d'-dimension vector $\boldsymbol{c}^{ij} = [c_1^{ij}, c_2^{ij}, \cdots, c_{d'}^{ij}]$, *i.e.*,

$$C^{ij} = circ(\boldsymbol{c}^{ij}) = \begin{bmatrix} c_1^{ij} & c_{d'}^{ij} & \cdots & c_3^{ij} & c_2^{ij} \\ c_2^{ij} & c_1^{ij} & c_{d'}^{ij} & & c_3^{ij} \\ \vdots & c_2^{ij} & c_1^{ij} & \ddots & \vdots \\ c_{d'-1}^{ij} & & \ddots & \ddots & c_d^{ij} \\ c_{d'}^{ij} & c_{d'-1}^{ij} & \cdots & c_2^{ij} & c_1^{ij} \end{bmatrix}. \tag{12}$$

For efficient implementation of the projection $\mathcal{T}(Z) = \sigma(Z\Theta)$, we first split the input Z into b slices $Z^j \in \mathbb{R}^{N\times d'}$, *i.e.*, $Z = [Z^1; Z^2; \cdots; Z^b]$, and then multiply each slice Z^j by a circulant matrix C^{ij}. The product between circulant matrix and vector in the original domain is equivalent

to element-wise multiplication in the Fourier domain, *i.e.*, $\mathcal{F}(Z^j C^{ij}) = \mathcal{F}(Z^j) \odot \mathcal{F}(\boldsymbol{c}^{ij})$, where $\mathcal{F}(\cdot)$ is the discrete Fourier transformation and $\odot$ denotes that each row in Z^j is multiplied by the according element in $\boldsymbol{c}^{ij}$. The discrete Fourier transform and its inverse transformation both can be efficiently calculated with fast Fourier transform FFT and its inverse IFFT with only $\mathcal{O}(d' \log d')$ computational complexity. The output is calculated as:

$$\mathcal{T}(Z)^i = \sigma\left(\sum_{j=1}^{b} Z^j C^{ij}\right) = \sigma\left(\sum_{j=1}^{b} \mathrm{IFFT}(\mathrm{FFT}(Z^j) \circ \mathrm{FFT}(\boldsymbol{c}^{ij}))\right), \tag{13}$$

where $\mathcal{T}(Z)^i \in \mathbb{R}^{N \times d'}$ is a slice of $\mathcal{T}(Z)$. Finally, $\mathcal{T}(Z)$ is obtained by concatenating slices $\mathcal{T}(Z)^i$, *i.e.*, $\mathcal{T}(Z) = [\mathcal{T}(Z)^1; \mathcal{T}(Z)^2; \cdots; \mathcal{T}(Z)^b]$.

Complexity Analysis. For a $d \times d$ matrix Θ split into multiple b^2 sub-matrices, the numbers of parameters is only bd, which has linear complexity with matrix size d. Note that the learnable parameters $\boldsymbol{c}^{ij}$ can be stored as its Fourier form, and both the FFT and IFFT operations in Eq. 13 are conducted one time. Thus the computational cost (*i.e.*, FLOPs) of Eq. 13 is about $(d \log(d/b) + 2bd)$, where $d \log(d/b)$ comes from the FFT and IFFT transformation and $2bd$ is consumed by the element-wisely operation in the Fourier domain [21, 2][2]. The size of matrix d in a transformer is usually large while the number of sub-matrices b is small, and thus the parameters and computation costs incurred by the augmented shortcuts can be negligible. For example, in the ViT-S model with $d = 384$, b is set to 4 in our implementation. The augmented shortcuts only adds 0.07 M parameters and 0.08 G FLOPs, which is negligible considering the ViT-S model has 22.1 M parameters and 4.6 G FLOPs.

4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed augmented shortcuts. The vision transformer [9] and its SOTA variants are equipped with the augmented shortcuts for performance improving. We first compare the performances of different models on the ImageNet dataset for the image classification task. Then ablation studies are conducted to analyze the algorithm. To validate its generalization ability, we further test the proposed Aug-ViT model on the object detection and transfer learning tasks.

4.1 Experiments on ImageNet

Dataset. ImageNet (ILSVRC-2012) dataset [6] contains 1.3 M training images and 50k validation images from 1000 classes, which is a widely used image classification benchmark.

Implementation details. We use the same training strategy of DeiT [27] for a fair comparison. Specifically, the model is trained with AdamW [19] optimizer for 300 epochs with batchsize 1024. The learning rate is initialized to 10^{-3} and then decayed with the cosine schedule. Label smoothing [25], DropPath [17] and repeated augmentation [14] are also implemented following DeiT [27]. The data augmentation strategy contains Rand-Augment [5], Mixup [33] and CutMix [32]. Besides the original shortcuts, two augmented shortcuts are added, where the hyper-parameter b for partitioning matrices is set to 4 empirically. The models are trained from scratch on ImageNet and no extra data are used. All experiments are conducted with PyTorch [23] on NVIDIA V100 GPUs.

Backbones. We apply the augmented shortcuts on multiple vision transformer models. ViT [9] is the typical transformer model for the vision tasks, which splits an image to multiple 16×16 patches. Deit [27] adopts the same architecture with ViT but improves the training strategy for better performance. T2T [31] and PVT [30] are two recently proposed SOTA variants of ViT. T2T [31] improves the process of producing patches by considering the structured information in images. PVT [30] designs a pyramid-like structure by partitioning the model into multiple stages. In Table 2, 'ViT(DeiT)-S' ,'ViT(DeiT)-B' and 'PVT-S', 'PVT-B' denote the DeiT and PVT models with different size. 'T2T-14', 'T2T-19' and 'T2T-24' are the T2T models with different depths.

[2]The constants are approximated by considering the relation between complex operation and real operation, as well as the symmetry property [2]

Table 2: Performance of different models on ImageNet.

Model	Resolution	Top1-Accuracy (%)	Params (M)	FLOPs (G)
ViT(DeiT)-S [9]	224×224	79.8	22.1	4.6
Aug-ViT-S	224×224	**80.9** (+1.1)	22.2	4.7
ViT(DeiT)-B [9]	224×224	81.8	86.4	17.6
Aug-ViT-B	224×224	**82.4** (+0.6)	86.5	17.7
ViT(DeiT)-B↑ [9]	384× 384	83.1	86.4	55.6
Aug-ViT-B↑	384× 384	**84.2** (+1.1)	86.5	55.8
PVT-S [30]	224×224	79.8	24.5	3.8
Aug-PVT-S [30]	224×224	**80.5** (+0.7)	24.6	3.9
PVT-M [30]	224×224	81.2	44.2	6.7
Aug-PVT-M [30]	224×224	**82.3** (+1.1)	44.3	6.8
T2T-14 [31]	224×224	81.5	21.5	5.2
Aug-T2T-14	224×224	**82.1** (+0.6)	21.6	5.3
T2T-19 [31]	224×224	81.9	39.2	8.9
Aug-T2T-19	224×224	**82.9** (+1.0)	39.3	9.0
T2T-24 [31]	224×224	82.3	64.1	14.1
Aug-T2T-24	224×224	**83.6** (+1.3)	64.2	14.3

Experimental Results. The validation accuracies of different models on ImageNet are shown in Table 2. [3] We firstly equip the ViT models with the augmented shortcuts and get Aug-ViT, which show large superiority to the plain counterparts, *i.e.*, more than 1% accuracy improvement without noticeable parameter and computational complexity increasing. For example, the proposed 'Aug-ViT-S' achieves 80.9% top-1 accuracy with 4.6G FLOPs, which suppresses the baseline ('ViT(DeiT)-S' with 4.6G FLOPs) by 1.2% top-1 accuracy while the computational cost is barely changed. For a large model with higher input image resolution (*e.g.*, ViT-B with input resolution 384×384) and high performance, equipping it with the augmented shortcuts can still further improve the performance (*e.g.*, 83.1% $\rightarrow$ 84.2%).

Besides the typical ViT model, the augmented shortcuts can be embedded into multiple vision transformers flexibly and improve their performance as well. For example, equipping the PVT-M with augmented shortcut can improve its accuracy form 81.2% to 82.3%. For the T2T-14 model, the performance improvement is even more obvious, *i.e.*, 1.3% accuracy improvement from 82.3% to 83.6%.

Performance Improvement *w.r.t.* Network Depth. It is interesting to see that the augmented shortcuts improve the performance of deeper models more obviously. 'T2T-14', 'T2T-19' and 'T2T-24' compose of the same blocks but have different depths. For the baseline, increasing the depth of T2T model from 14 to 24, the accuracy is only improved by 0.7% (from 81.5% to 82.3%). While with the augmented shortcuts, the Aug-T2T model with 24 layers can achieve 83.6%, which achieves more obvious performance improvement than models with 14 layers. We conjecture that it is because deeper models tend to suffer more serve feature collapse suffer more serve feature collapse as features from different patches are aggregated with more attention layers.

4.2 Ablation Studies

To better understand the proposed augmented shortcuts for vision transformers, we conduct extensive experiments to investigate the impact of each component. All the ablation experiments are conducted based on ViT(DeiT)-S model on the ImageNet dataset.

The number of augmented shortcuts. The performance varies *w.r.t.* the number of augmented shortcuts as shown in Table 3. Besides the original identity shortcut, adding only one augmented shortcut can significantly improves the performance of the ViT model (*e.g.*, 0.8% top-1 accuracy improvement compared to the baseline). Further increasing the number of augmented shortcuts will further improve the performance, and the improvement margin will be saturated gradually. We empirically find that two augmented shortcuts are enough to achieve obvious performance improvement.

[3] Note that DeiT [27] adopts the same model architecture with ViT [9] but achieves higher performance by adjusting the training strategy, which is used as the baseline model.

Table 3: The number of augmented shortcuts.

# Path	Top1-Acc. (%)	Params (M)	FLOPs (G)
0	79.8 (+0.0)	22.1	4.6
1	80.6 (+0.8)	+0.04	+0.04
2	80.9 (+1.1)	+0.07	+0.08
3	80.9 (+1.1)	+0.12	+0.12

Table 4: Location for implementing the augmented shortcuts.

Location	Top1-Acc. (%)	Params (M)	FLOPs (G)
Baseline	79.8 (+0.0)	22.1	4.6
MSA	80.6 (+0.8)	+0.035	+0.04
MLP	80.4 (+0.6)	+0.035	+0.04
MSA&MLP	80.9 (+1.1)	+0.07	+0.08

Table 5: Efficiency of the block cifculant projection.

Type	b	Top1-Acc. (%)	Params (M)	FLOPs (G)
Baseline	-	79.8	22.1	4.6
Block-Circulant	1	80.5 (+0.7)	+0.02	+0.07
Block-Circulant	2	80.7 (+0.9)	+0.03	+0.07
Block-Circulant	4	80.9 (+1.1)	+0.07	+0.08
Block-Circulant	8	80.9 (+1.1)	+0.15	+0.10
Unstructured	-	81.0 (+1.2)	+7.1	+1.4

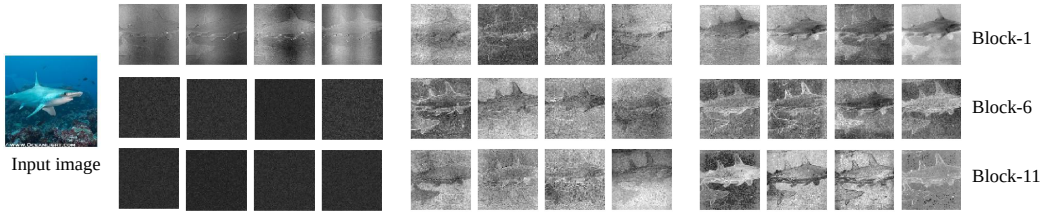
Table 6: Formulation of the augmented shortcuts. 'Act' denotes the activation function.

Form	Top1-Acc. (%)
Baseline	79.8
Act.	79.9
Circulant	80.6
Act. + Circulant	80.8
Circulant + Act.	80.9

Location for implementing the augmented shortcuts. As discussed before, the augmented shortcuts can be paralleled with both MSA and MLP modules to increase the feature diversity. Table 4 shows how the implementation location affects the final performance. Paralleling MSA with the augmented shortcuts significantly improves the performance (*e.g.*, 0.8% top-1 accuracy), which we attribute it to the increasing of feature diversity. Enhancing MLP module also bring the performance improvement, which is accordant to our analysis in supplementary materials. Combining them together can achieve the highest performance (1.1% accuracy improvement), which we adopt in our implementation.

Efficiency of the block-circulant projection. In the block-wisely circulant projection, the hyper-parameter b controls the number of sub-matrices C^{ij} partitioned by original matrix Θ. Table 5 shows how the performance varies *w.r.t.* the parameter b. A larger b implies the matrix Θ will be partitioned into more circulant matrix with smaller sizes, which brings more parameters and higher computational cost, as well as the performance improvement. The unstructured projection can also be used as augmented shortcuts and brings performance improvement, but it incurs obvious increasing of parameters and computational cost. Using the block-circulant projection with $b = 4$ can achieve very similar performance with the unstructured projection but has much few parameters (*e.g.*, 0.07M *vs.*7.1M), implying that the block-circulant projection is an efficient and effective formulation to transform features in the augmented shortcuts.

Formulation of the augmented shortcuts. The augmented shortcuts are implemented sequentially with the block-circulant projection and activation function (*e.g.*, GeLU). Table 6 shows the impact of each component. Only using the activation function introduces no learnable parameters, which only achieves similar accuracy with the baseline model, implying that the learnable parameters are vital to produce diverse features. If only the linear circulant projection is kept, multiple augmented shortcut can be merged to a single one. Swapping the sequential order of the circulant projection and activation function has negligible influence on the final performance.



(a) ViT w/o Shortcut (b) ViT with Shortcut (c) Aug-ViT

Figure 5: Feature Visualization of different models.

Feature Visualization. We intuitively show the features of different models in Figure 5, From top to bottom are features in low, middle and deep layers of the ViT-Small model. The input image is scaled to 1024×1024 for better visualization and the patch embeddings are reshaped to their spatial positions to construct the features maps. Without shortcut, the feature maps in deep layers conveys no effective information (Figure 5 (a)), and adding a shortcut connection make the feature maps informative ((b)). Compared with them, the features in the Aug-ViT model are further enriched, especially for the deep layers.

4.3 Object Detection with Pure Transformer

We also validate the effectiveness of the augmented shortcuts on the objection detection task. A pure transformer detector can be constructed by combining the vision transformer backbone and the DETR [3] head, and we equip the backbones with the augmented shortcuts. For a fair comparison, we follow the training strategy in PVT [30] and fine-tune the models for 50 epochs on the COCO train2017 dataset. Random flip and random scale are used as the data augmentation strategy. The results on COCO val2017 are shown in Table 7. The detector equipped with augmented shortcut achieve better performance than the base model. For the DeiT-S model with 33.9 AP, the augmented shortcut can improve 1.8% AP and achieve 35.7%.

Table 7: Results of pure transformer object detection on COCO val2017 dataset.

Backbone	Parmas(M)	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Backbone	41	50	32.3	53.9	32.3	10.7	33.8	53.0
DeiT-S [27]	38	50	33.9	54.7	34.3	11.0	35.4	56.6
Aug-ViT-S	38	50	**35.7**	**56.7**	**36.5**	**12.2**	**38.0**	**58.6**
PVT-S [30]	40	50	34.7	55.7	35.4	12.0	36.4	56.7
Aug-PVT-S	40	50	**35.1**	**55.8**	**35.9**	**12.5**	**36.5**	**57.3**

4.4 Transfer Learning

To validate the generalization ability of vision transformers equipped with the projection, we conduct experiments on the transfer learning tasks. Specifically, the models trained on ImageNet is further fine-tuned on the downstream tasks, containing superordinate-level image recognition dataset (CIFAR-10 [16], CIFAR-100 [16]) and fine-grained image recognition dataset(Oxford 102 Flowers [20] and Oxford-IIIT Pets [22]). Following [27], we fine-tune the models on images with resolution 384×384, and use the same fine-tuning strategy as [27]. Table 8 shows the performances of different models on the downstream tasks, and the model equipped with the augmented shortcut always achieves higher accuracies than the baseline on different tasks.

Table 8: Results on downstream tasks with ImageNet pre-training. All the models are fine-tuned with the image resolution 384×384.

Model	Parmas(M)	ImageNet	CIFAR-10	CIFAR-100	Flowers	Pets
ViT-B/16$_{\uparrow 384}$	86.4	77.9	98.1	87.1	89.5	93.8
DeiT-B$_{\uparrow 384}$	86.4	83.1	99.1	90.8	98.4	-
Aug-ViT-B$_{\uparrow 384}$	**86.4**	**84.1**	**99.2**	**91.3**	**98.8**	**95.1**

5 Conclusion

We presented augmented shortcuts for resolving the feature collapse issue in vision transformers. The augmented shortcuts are parallel with the original identity shortcuts, and each connection has its own learnable parameters to make diverse transformations on the input features. Efficient circulant projections are used to implement the augmented shortcuts, whose memory and computational cost are negligible compared with other components in the vision transformer. Similar to the widely used identity shortcuts, the augmented shortcuts do not depend on the specific architecture design either, which can be flexibly embedded into various variants of vision transformers (*e.g.*, ViT [9], T2T [31], PVT [30]) for enhancing their performance on different tasks such as image classification, object detection and transfer learning. In the future, we plan to research designing deeper vision transformers with the help of augmented shortcuts.

References

[1] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pages 342–350. PMLR, 2017.

[2] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Claude R Dietrich and Garry N Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

[8] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Robert M Gray. Toeplitz and circulant matrices: A review. 2006.

[11] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Aicke Hinrichs and Jan Vybíral. Johnson-lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011.

[14] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020.

[15] Irwin Kra and Santiago R Simanca. On circulant matrices. *Notices of the AMS*, 59(3):368–377, 2012.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[18] Tianyi Liu, Minshuo Chen, Mo Zhou, Simon S Du, Enlu Zhou, and Tuo Zhao. Towards understanding the importance of shortcut connections in residual networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[20] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[21] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

[22] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

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

[24] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[26] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[27] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[29] Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *arXiv preprint arXiv:1605.06431*, 2016.

[30] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[31] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[32] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[33] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract Section 1.
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See the supplemental materials.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the supplemental materials.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code will be available upon acceptance.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.

(b) Did you mention the license of the assets? [Yes] See Section 4.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]