DeepViT: Towards Deeper Vision Transformer

Daquan Zhou[1], Bingyi Kang[1], Xiaojie Jin[2], Linjie Yang[2],
Xiaochen Lian[2], Zihang Jiang[1], Qibin Hou[1], Jiashi Feng[1]
[1]National University of Singapore,   [2]ByteDance US AI Lab
{zhoudaquan21, xjjin0731, lianxiaochen, yljatthu, andrewhoux}@gmail.com
jzihang, kang, elefjia@nus.edu.sg

Abstract

Vision transformers (ViTs) have been successfully applied in image classification tasks recently. In this paper, we show that, unlike convolution neural networks (CNNs) that can be improved by stacking more convolutional layers, the performance of ViTs saturate fast when scaled to be deeper. More specifically, we empirically observe that such scaling difficulty is caused by the attention collapse *issue: as the transformer goes deeper, the attention maps gradually become similar and even much the same after certain layers. In other words, the feature maps tend to be identical in the top layers of deep ViT models. This fact demonstrates that in deeper layers of ViTs, the self-attention mechanism fails to learn effective concepts for representation learning and hinders the model from getting expected performance gain. Based on above observation, we propose a simple yet effective method, named* Re-attention*, to re-generate the attention maps to increase their diversity at different layers with negligible computation and memory cost. The proposed method makes it feasible to train deeper ViT models with consistent performance improvements via minor modification to existing ViT models. Notably, when training a deep ViT model with 32 transformer blocks, the Top-1 classification accuracy can be improved by 1.6% on ImageNet. Code is publicly available at* `https://github.com/zhoudaquan/dvit_repo`*.*

1. Introduction

Recent studies [7, 37] have demonstrated that transformers [38] can be successfully applied to vision tasks [18] with competitive performance compared with convolutional neural networks (CNNs) [9, 35]. Different from CNNs that aggregate global information by stacking multiple convolutions (*e.g.*, 3×3) [9, 10], vision transformers (ViTs) [7] take advantages of the self-attention (SA) mechanism [38] to capture spatial patterns and non-local dependencies. This allows ViTs to aggregate rich global information
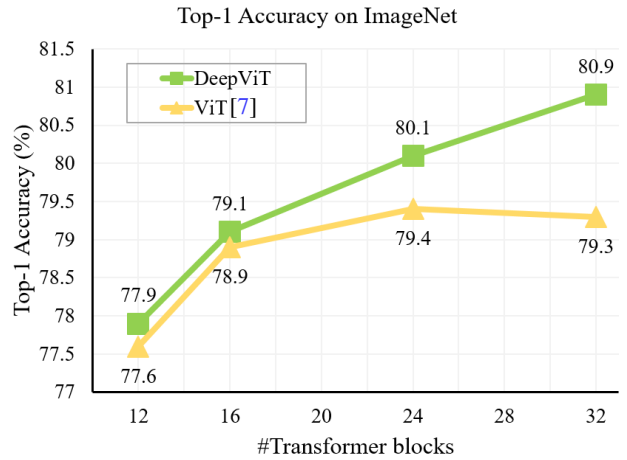

Figure 1: Top-1 classification performance of vision transformers (ViTs) [7] on ImageNet with different network depth {12, 16, 24, 32}. Directly scaling the depth of ViT by stacking more transformer blocks cannot monotonically increase the performance. Instead, the model performance saturates when going deeper. In contrast, with the proposed Re-attention, our DeepViT model successfully achieves better performance when it goes deeper.

without handcrafting layer-wise local feature extractions as CNNs and thus achieves better performance. For example, as shown in [37], a 12-block ViT model with 22M learnable parameters achieves better results than the ResNet-101 model which has more than 30 bottleneck convolutional blocks in ImageNet classification.

The recent progress of deep CNN models is largely driven by training very deep models with a large number of layers which is enabled by novel model architecture designs [9, 41, 31, 20, 47]. This is because a deeper CNN can learn richer and more complex representations for the input images and provide better performance on vision tasks [1, 45, 29]. Thus, how to effectively scale CNNs to be deeper is an important theme in recent deep learning fields, which stimulates the techniques like residual learning [9].

Considering the remarkable performance of shallow

ViTs, a natural question arises: *can we further improve performance of ViTs by making it deeper, just like CNNs?* Though it seems to be straightforward at the first glance, the answer may not be trivial since ViT is essentially different from CNNs in its heavy reliance on the self-attention mechanism. To settle the question, we investigate in detail the scalability of ViTs along depth in this work.

We start with a pilot study on ImageNet to investigate how the performance of ViT changes with increased model depth. In Fig. 1, we show the performance of ViTs [7] with different block numbers (green line), ranging from 12 to 32. As shown, as the number of transformer blocks increases, the model performance does not improve accordingly. To our surprise, the ViT model with 32 transformer blocks performs even worse than the one with 24 blocks. This means that directly stacking more transformer blocks as performed in CNNs [9] is inefficient at enhancing ViT models. We then dig into the cause of this phenomenon. We empirically observed that as the depth of ViTs increases, the attention maps, used for aggregating the features for each transformer block, tend to be overly similar after certain layers, which makes the representations stop evolving after certain layers. We name this specific issue as *attention collapse*. This indicates that as the ViT goes deeper, the self-attention mechanism becomes less effective for generating diverse attentions to capture rich representations.

To resolve the attention collapse issue and effectively scale the vision transformer to be deeper, We present a simple yet effective self-attention mechanism, named as Re-attention. Our Re-attention takes advantage of the multi-head self-attention(MHSA) structure and regenerates attention maps by exchanging the information from different attention heads in a learnable manner. Experiments show that, Without any extra augmentation and regularization policies, simply replacing the MHSA module in ViTs with Re-attention allows us to train very deep vision transformers with even 32 transformer blocks with consistent improvements as shown in Fig. 1. In addition, we also provide ablation analysis to help better understand of the role of Re-attention in scaling vision transformers.

To sum up, our contributions are as follows:

- We deeply study the behaviour of vision transformers and observe that they cannot continuously benefit from stacking more layers as CNNs. We further identify the underlying reasons behind such a counter-intuitive phenomenon and conclude it as *attention collapse* for the first time.

- We present Re-attention, a simple yet effective attention mechanism that considers information exchange among different attention heads.

- To the best of our knowledge, we are the first to successfully train a 32-block ViT on ImageNet-1k from

scratch with consistent performance improvement. We show that by replacing the self-attention module with our Re-attention, new state-of-the-art results can be achieved on the ImageNet-1k dataset without any pre-training on larger datasets.

2. Related Work

2.1. Transformers for Vision Tasks

Transformers [38] are initially used for machine translation which replace the recurrence and convolutions entirely with self-attention mechanisms [28, 14, 48, 11, 39, 17] and achieve outstanding performance. Later, transformers become the dominant models for various natural language processing (NLP) tasks [2, 26, 6, 21]. Motivated by their success on the NLP tasks, recent researchers attempted to combine the self-attention mechanism into CNNs for computer vision tasks [40, 3, 4, 32, 24, 50, 49].. Those achievements also stimulate interests of the community in building purely transformer-based models (without convolutions and inductive bias) for vision tasks. The vision transformer (ViT) [7] is among the first attempt that uses the pure transformer architecture to achieve competitive performance with CNNs on the image classification task. However, due to the large model complexity, ViT needs to be pre-trained on larger-scale datasets (e.g., JFT300M) for performing well on the ImageNet-1k dataset. To solve the data efficiency issue, DeiT [37] deploys knowledge distillation to train the model with a larger pre-trained teacher model. In this manner, vision transformer can perform well on ImageNet-1k without the need of pre-training on larger dataset. Differently, in this work, we target at a different problem with ViT, *i.e.*, how to effectively scale ViT to be deeper. We propose a new design for the self-attention mechanism so that it can perform well on vision tasks without the need of extra data, teacher networks, and the domain specific inductive bias.

2.2. Depth Scaling of CNNs

Increasing the network depth of a CNN model is deemed to be an effective way to improve the model performance [30, 33, 34, 10, 15]. However, very deep CNNs are generally harder to train to perform significantly better than the shallow ones in the past [8, 44]. How to effectively scale up the CNNs in depth was a long-standing and challenging problem [16]. The recent progress of CNNs largely benefits from novel architecture design strategies that make training deep CNNs more effective [9, 35, 13, 12, 36, 52]. Transformer-alike models have modularized architectures and thus can be easily made deeper by repeating the basic transformer blocks or using larger embedding dimensions [2, 19]. However, those straightforward scaling strategies only work well with larger datasets and stronger augmentation policies [51, 46, 43] to alleviate the brought training
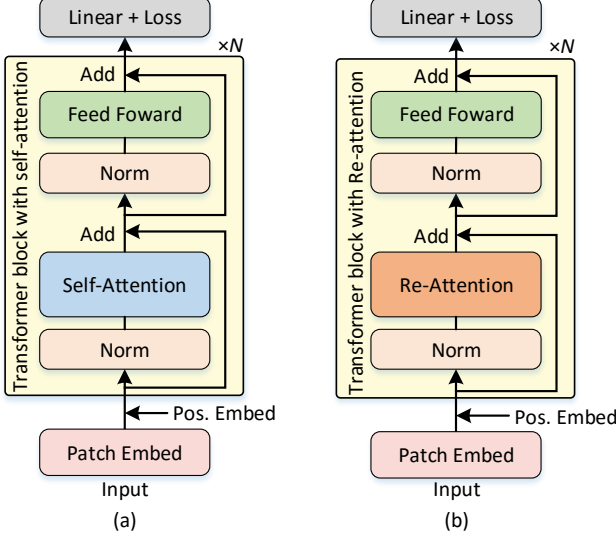
Figure 2: Comparison between the (a) original ViT with N transformer blocks and (b) our proposed DeepViT model. Different from ViT, DeepViT replaces the self-attention layer within the transformer block with the proposed Re-attention which effectively addresses the attention collapse issue and enables training deeper ViTs. More details are given in Sec. 4.2.

difficulties. In this paper, we observed that with the same dataset, the performance of vision transformers do saturate as the network depth rises. We rethink the self-attention mechanism and present a simple but effective approach to address the difficulties in scaling vision transformers.

3. Revisiting Vision Transformer

A vision transformer (ViT) model [37, 7], as depicted in Fig. 2(a), is composed of three main components: a linear layer for patch embedding (*i.e.*, mapping the high-resolution input image to a low-resolution feature map), a stack of transformer blocks with multi-head self-attention and feed-forward layers for feature encoding, and a linear layer for classification score prediction. In this section, we first review its unique transformer blocks, in particular the self-attention mechanism, and then we provide studies on the collapse problem of self-attention.

3.1. Multi-Head Self-Attention

Transformers [38] were extensively used in natural language for encoding a sequence of input word tokens into a sequence of embeddings. To comply with such sequence-to-sequence learning structure when processing images, ViTs first divide an input image into multiple patches uniformly and encode each patch into a token embedding. Then, all these tokens, together with a class token, are fed into a stack of transformer blocks.

Each transformer block consists of a multi-head self-

attention (MHSA) layer and a feed-forward multi-layer perceptron (MLP). The MHSA generates a trainable associate memory with a query (Q) and a pair of key (K)-value (V) pairs to an output via linearly transforming the input. Mathematically, the output of a MHSA is calculated by:

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^{\top}/\sqrt{d})V, \quad (1)$$

where $\sqrt{d}$ is a scaling factor based on the depth of the network. The output of the MHSA is then normalized and fed into the MLP to generate the input to the next block. In the above self-attention, Q and K are multiplied to generate the attention map, which represents the correlation between all the tokens within each layer. It is used to retrieve and combine the embeddings in the value V. In the following, we particularly analyze the role of the attention map in scaling the ViT. For convenience, we use $\mathbf{A} \in \mathbb{R}^{H \times T \times T}$ to denote the attention map, with H being the number of SA heads and T the number of tokens. For the h-th SA head, the attention map is computed as $\mathbf{A}_{h,:,:} = \text{Softmax}(Q_h K_h^{\top}/\sqrt{d})$ with Q_h and K_h from the corresponding head. When the context is clear, we omit the subscript h.

3.2. Attention Collapse

Motivated by the success of deep CNNs [9, 30, 35, 36], we conduct systematic study in the changes of the performance of ViTs as depth increases. Without loss of generality, we first fix the hidden dimension and the number of heads to 384 and 12 respectively[1], following the common practice in [37]. Then we stack different number of transformer blocks (varying from 12 to 32) to build multiple ViT models corresponding to different depths. The overall performances for image classification are evaluated on ImageNet [18] and summarized in Fig. 1. As evidenced by the performance curve, we surprisingly find that the classification accuracy improves slowly and saturates fast as the model goes deeper. More specifically, we can observe that the improvement stops after employing 24 transformer blocks. This phenomenon demonstrates that existing ViTs have difficulty in gaining benefits from deeper architectures.

Such a problem is quite counter-intuitive and worth exploration, as similar issues (*i.e.*, how to effectively train a deeper model) have also been observed for CNNs at its early development stage [9], but properly solved later [9, 10]. By taking a deeper look into the transformer architecture, we would like to highlight that the self-attention mechanism plays a key role in ViTs, which makes it significantly different from CNNs. Therefore, we start with investigating how the self-attention, or more concretely, the generated attention map $\mathbf{A}$ varies as the model goes deeper.

---

[1]Similar phenomenon can also be found when we vary the hidden dimension size according to our experiments.
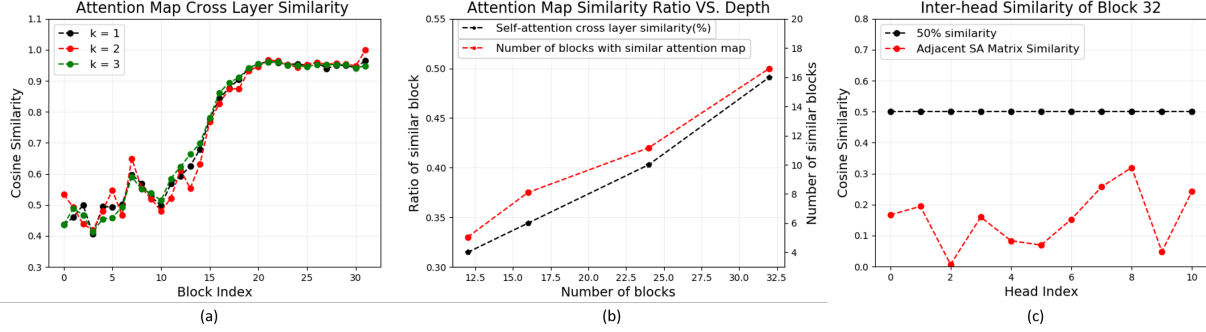
Figure 3: (a) The similarity ratio of the generated self-attention maps across different layers. The visualization is based on ViT models with 32 blocks pre-trained on ImageNet. For visualization purpose, we plot the ratio of token-wise attention vectors with similarity in Eqn. (2) larger than the average similarity within nearest k transformer blocks. As can be seen, the similarity ratio is larger than 90% for blocks after the 17th one. (b) The ratio of similar blocks to the total number of blocks increases when the depth of the ViT model increases. (c) Similarity of attention maps from different heads within the same block. The similarity between different heads within the blocks is all lower than 30% and they present sufficient diversity.

To measure the evolution of the attention maps over layers, we compute the ==following cross-layer similarity between the attention maps from different layers:==

$$M_{h,t}^{p,q} = \frac{\mathbf{A}_{h,:,t}^{p}{}^{\top}\mathbf{A}_{h,:,t}^{q}}{\|\mathbf{A}_{h,:,t}^{p}\|\|\mathbf{A}_{h,:,t}^{q}\|}, \qquad (2)$$

where $M^{p,q}$ is the cosine similarity matrix between the attention map of layers p and q. Each element $M_{h,t}^{p,q}$ measures the similarity of attention for head h and token t. Consider one specific self-attention layer and its h-th head, $\mathbf{A}_{h,:,t}^{*}$ is a T-dimensional vector representing how much the input token t contributes to each of the T output tokens. $M_{h,t}^{p,q}$, thus, provides an appropriate measurement on how the contribution of one token varies from layer p to q. When $M_{h,t}^{p,q}$ equals one, it means that token t plays exactly the same role for self-attention in layers p and q.

Given Eqn. (2), we then train a ViT model with 32 transformer blocks on ImageNet-1k and investigate the above similarity among all the attention maps. As shown in Fig. 3(a), the ratio of similar attention maps in M after the 17th block is larger than 90% . This indicates that the learned attention maps afterwards are similar and the transformer block may degenerate to an MLP. As a result, further stacking such degenerated MHSA may introduce the model rank degeneration issue (*i.e.*, the rank of the model parameter tensor from multiplying the layer-wise parameters together will decrease) and limits the model learning capacity. This is also validated by our analysis on the degeneration of learned features as shown below. Such observed attention collapse could be one of the reasons for the observed performance saturation of ViTs. To further validate the existence of this phenomenon for ViTs with different depths, we conduct the same experiments on ViTs with 12, 16, 24 and 32 transformer blocks respectively and calculate the number of blocks with similar attention maps. The re-
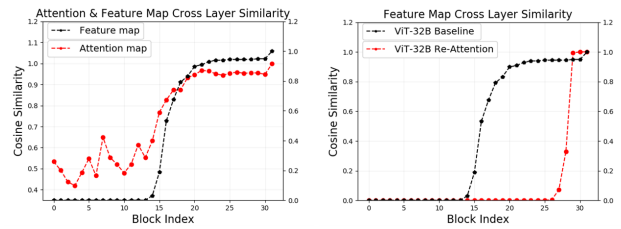

Figure 4: (**Left**): Cross layer similarity of attention map and features for ViTs. The black dotted line shows the cosine similarity between feature maps of the last block and each of the previous blocks. The red dotted line shows the ratio of similar attention maps of adjacent blocks. The visualization is based on a 32-block ViT model pre-trained on ImageNet-1k. (**Right**): Feature map cross layer cosine similarity for both the original ViT model and ours. As can be seen, replacing the original self-attention with Re-attention could reduce the feature map similarity significantly.

sults shown in Fig. 3(b) clearly demonstrate the ratio of the number of similar attention map blocks to the total number of blocks increases when adding more transformer blocks.

To understand how the attention collapse may hurt the ViT model performance, we further study how it affects feature learning of the deeper layers. For a specific 32-block ViT model, we compare the final output features with the outputs of each intermediate transformer block by investigating their cosine similarity. The results in Fig. 4 demonstrate that the similarity is quite high and the learned features stop evolving after the 20th block. There is a close correlation between the increase of attention similarity and feature similarity. This observation indicates that attention collapse is responsible for the non-scalable issue of ViTs.

4. Re-attention for Deep ViT

As revealed above, one major obstacle in scaling up ViT to a deeper one is the attention collapse problem. In this sec-
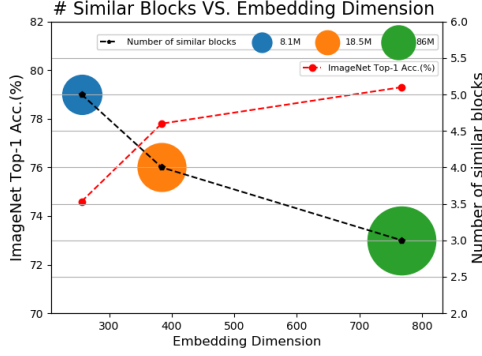
Figure 5: Impacts of embedding dimension on the similarity of generated self-attention map across layers. As can be seen, the number of similar attention maps decreases with increasing embedding dimension. However, the model size also increases rapidly.

Table 1: Top-1 accuracy on ImageNet-1k dataset of vision transformer with different embedding dimensions. The number of model parameters increase quadratically with the embedding dimension. The number of similar attention map blocks with different embedding dimensions are shown in Figure 5

.

Model	#Blocks	Embed Dim.	#Param. (M)	Top-1 Acc.(%)
ViT	12	256	8.15	74.6
ViT	12	384	18.51	77.86
ViT	12	512	33.04	78.8
ViT	12	768	86	79.3

tion, we present two solution approaches, one is to increase the hidden dimension for computing self-attention and the other one is a novel re-attention mechanism.

4.1. Self-Attention in Higher Dimension Space

One intuitive solution to conquer attention collapse is to increase the embedding dimension of each token. This will augment the representation capability of each token embedding to encode more information. As such, the resultant attention maps can be more diverse and the similarity between each block's attention map could be reduced. Without loss of generality, we verify this approach empirically by conducting a set of experiments based on ViT models with 12 blocks for quick experiments. Following previous transformer based works [38, 7], four embedding dimensions are selected, ranging from 256 to 768. The detailed configurations and the results are shown in Tab. 1.

From Fig. 5 and Tab. 1, one can see that the number of blocks with similar attention maps is reduced and the attention collapse is alleviated by increasing the embedding dimension. Consequently, the model performance is also increased accordingly. This validates our core hypothesis—the attention collapse is the main bottleneck for scaling ViT. Despite its effectiveness, increasing the embedding di-

mension also increases the computation cost significantly and the brought performance improvement tends to diminish. Besides, a larger model (with higher embedding dimension) typically needs more data for training, suffering the over-fitting risk and decreased efficiency.

4.2. Re-attention

It has been demonstrated in Sec. 3 that the similarity between attention maps across different transformer blocks is high, especially for deep layers. However, we find the similarity of attention maps from different heads of the same transformer block is quite small, as shown in Fig. 3(c). Clearly, different heads from the same self-attention layer focus on different aspects of the input tokens. Based on this observation, we propose to establish cross-head communication to re-generate the attention maps and train deep ViTs to perform better.

Concretely, we use the attention maps from the heads as basis and generate a new set of attention maps by dynamically aggregating them. To achieve this, we define a learnable transformation matrix $\Theta \in \mathbb{R}^{H \times H}$ and then use it to mix the multi-head attention maps into re-generated new ones, before being multiplied with V. Specifically, the Re-attention is implemented by:

$$\text{Re-Attention}(Q, K, V) = \text{Norm}(\Theta^\top(\text{Softmax}(\frac{QK^\top}{\sqrt{d}})))V, \tag{3}$$

where transformation matrix Θ is multiplied to the self-attention map $\mathbf{A}$ along the head dimension. Here Norm is a normalization function used to reduced the layer-wise variance. Θ is end-to-end learnable.

Advantages: The advantages of the proposed Re-attention are two-fold. First of all, compared with other possible attention augmentation methods, such as randomly dropping some elements of the attention map or tuning SoftMax temperature, our Re-attention exploits the interactions among different attention heads to collect their complementary information and better improves the attention map diversity. This is also verified by our following experiments. Furthermore, our Re-attention is effective and easy to implement. It needs only a few lines of code and negligible computational overhead compared to the original self-attention. Thus it is much more efficient than the approach of increasing embedding dimension.

5. Experiments

In this section, we first conduct experiments to further demonstrate the attention collapse problem. Then, we give extensive ablation analysis to show the advantages of the proposed Re-attention. By incorporating Re-attention into the transformers, we design two modified version of vision transformers and name them as deep vision transform-

ers (DeepViT). Finally, we compare the proposed DeepViT models against the latest state-of-the-arts (SOTA).

5.1. Experiment Details

To make a fair comparison, we first tuned a set of parameters for training the ViT base model and then use the same set of hyper-parameters for all the ablation experiments. Specifically, we use AdamW optimizer [23] and cosine learning rate decay policy with an initial learning rate of 0.0005. We use 8 Telsa-V100 GPUs and train the model for 300 epochs using Pytorch [25] library. The batch size is set to 256. We use 3 epochs for learning rate warm-up [22]. We also use some augmentation techniques such as mixup [46] and random augmentation [5] to boost the performance of baseline models following [47]. When comparing with other methods, we adopt the same set of hyper-parameters as used by the target models. We report results on the ImageNet dataset [18]. For all experiments, the image size is set to be 224×224. To study the scaling capability of current transformer blocks, we set the embedding dimension to 384 and the expansion ratio 3 for the MLP layers. We use 12 heads for all the models. More detailed configurations are shown in Tab. 2.

Table 2: Baseline model specifications. All ablation experiments are based on the ViT models with different number of blocks. The '#B' in 'ViT-#B' denotes the number of transformer blocks in the model.

Model	#Blocks	#Embeddings	MLP Size	Params. (M)
ViT-16B	16	384	1152	24.46
ViT-24B	24	384	1152	36.26
ViT-32B	32	384	1152	48.09

5.2. More Analysis on Attention Collapse

In this section, we show more analysis on the attention map similarity and study how the collapsed attention maps affect the model performance.

Attention reuse: As discussed above, when the model goes deeper, the attention maps of the deeper blocks become highly similar. This implies that adding more blocks on a deep ViT model may not improve the model performance. To further verify this claim, we design an experiment to reuse the attention maps computed at an early block of ViT to replace the ones after it. Specifically, we run experiments on the ViT models with 24 blocks and 32 blocks but share the Q and K values (and the resulted attention maps) of the last "unique" block to all the blocks afterwards. The "unique" block is defined as the block whose attention map's similarity ratio with adjacent layers is smaller than 90%. More implementation details can be found in the supplementary material. The results are shown in Tab. 3. Surprisingly, for a ViT model with 32 transformer blocks, when

we use the same Q and K values for the last 15 blocks, the performance degradation is negligible. This implies the attention collapse problem indeed exists and reveals the inefficacy in adding more blocks when the model is deep.

Table 3: ImageNet top-1 accuracy of the ViT models with shared self-attention maps. '#Shared blocks' denotes the number of the transformer blocks that share the same attention map.

#Blocks	#Embeddings	#Shared blocks	Top-1 Acc. (%)
24	384	0	79.3
24	384	11	78.7
32	384	0	79.2
32	384	15	79.2

Visualization: To more intuitively understand the attention map collapse across layers, we visualize the learned attention maps from different blocks of the original ViT [7]. We take a 32-block ViT model as an example and pre-train it on ImageNet. The visualization of the attention maps with original MHSA and Re-attention are shown in Fig. 6. It can be observed that the original MHSA learns the local relationship among the adjacent patches in the shallow blocks and the attention maps tend to expand to cover more patches gradually. In the deep blocks, the MHSA learns nearly uniform global attention maps with high similarity. Differently, after implementing Re-attention, the attention maps at deep blocks keep the diversity and have small similarities from adjacent blocks.

5.3. Analysis on Re-attention

In this subsection, we present two straightforward modifications to the current self-attention mechanism as baselines. We then conduct a series of comparison experiments to show the advantages of our proposed Re-attention.

Re-attention v.s. Self-attention: We first evaluate the effectiveness of Re-attention by comparing to the pure ViT models using the same set of training hyper-parameters. We directly replace the self-attention module in ViT with Re-attention and show the results in Tab. 4 with different number of transformer blocks. As can be seen, the vanilla ViT architecture suffers performance saturation when adding more transformer blocks. This phenomenon coincides with our observations that the number of blocks with similar attention maps increases with the depth as shown in Fig. 3(b). Interestingly, when replacing the self-attention with our proposed Re-attention, the number of similar blocks are all reduced to be zero and the performance rises consistently as the model depth increases. The performance gain is especially significant for deep ViT with 32 blocks. This might be explained by the fact that the 32 block ViT model has the largest number of blocks with similar attention maps and the improvements should be proportional to the number sim-
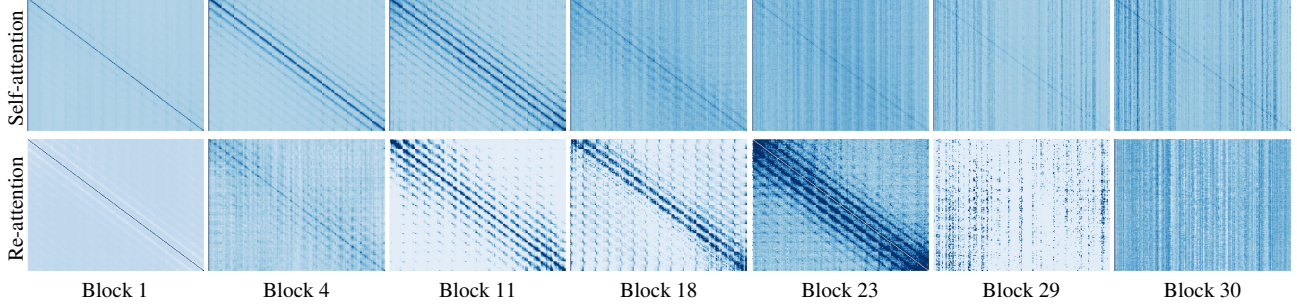
Figure 6: Attention map visualization of the selected blocks of the baseline ViT model with 32 transformer blocks. The first row is based on original Self-attention module and the second is based on Re-attention. As can be seen, the model only learns local patch relationship at its shallow blocks with the rest of attention values near to zero. Though their the scope increases gradually as the block goes deeper, the attention maps tend to become nearly uniform and thus lose diversity. After adding Re-attention, the originally similar attention maps are changed to be diverse as shown in the second row. Only at the last block's attention map, a nearly uniform attention map is learned.
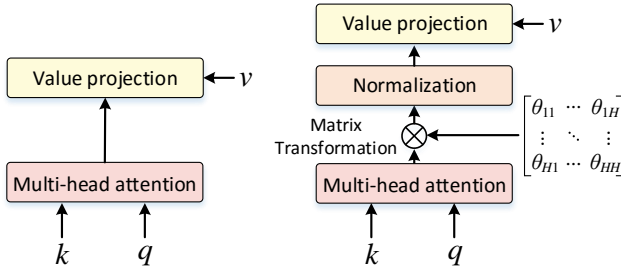

Figure 7: (**Left**): The original self-attention mechanism; (**Right**): Our proposed re-attention mechanism. As shown, the original attention map is mixed via a learnable matrix Θ before multiplied with values.

ilar blocks in the model. These experiments demonstrate that the proposed Re-attention can indeed solve the attention collapse problem and thus enables training a very deep vision transformer without extra datasets or augmentation policies.

Table 4: ImageNet Top-1 accuracy of deep ViT (DeepViT) models with Re-attention and different number of transformer blocks.

Model	#Similar Blocks	Param. (M)	Top-1 Acc. (%)
ViT-16B [7]	5	24.5	78.9
DeepViT-16B	0	24.5	79.1 (+0.2)
ViT-24B [7]	11	36.3	79.4
DeepViT-24B	0	36.3	80.1 (+0.7)
ViT-32B [7]	16	48.1	79.3
DeepViT-32B	0	48.1	80.9 (+1.6)

Comparison to adding temperature in self-attention: The most intuitive way to mitigate the over-smoothing phenomenon is to sharpen the distribution of the elements in the attention map of MHSA. We could achieve this by as-

signing a temperature τ to the Softmax layer of MHSA:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\tau\sqrt{d}}\right)V. \quad (4)$$

As the attention collapse is observed to be severe on deep layers (as shown in Fig. 3), we design two sets of experiments on a ViT model with 32 transformer blocks: (a) linearly decaying the temperature τ in each block such that the attention map distribution is sharpened and (b) making the temperature learnable and optimized together with the model training. We first check the impact of the Soft-Max temperature on reducing the attention map similarity. As shown in Fig. 8(a), the number of similar blocks are still large. Correspondingly, the feature similarity among blocks are also large as shown in Fig. 8(b). Thus, adding a temperature to the SoftMax only reduces the attention map similarity by a small margin. The classification results on ImageNet are shown in Tab. 5. As shown, using a learnable temperature could improve the performance but the improvement is marginal.

Table 5: ImageNet Top-1 accuracy of the ViT models with Soft-Max temperature τ and the drop attention. The embedding dimension of all the models is set as 384.

# Blocks	# Similar Blocks	Model	Acc. (%)
32	16	Vanilla	79.3
32	13	Linearly decayed τ	79.0
32	10	Learnable τ	79.5
32	8	drop attention	79.5
32	0	Re-attention	80.9

Comparison to dropping attentions: Another baseline we have attempted to differentiate the self-attention maps across layers is to use random dropout on the attention maps **A**. As the dropout will mask out different positions on
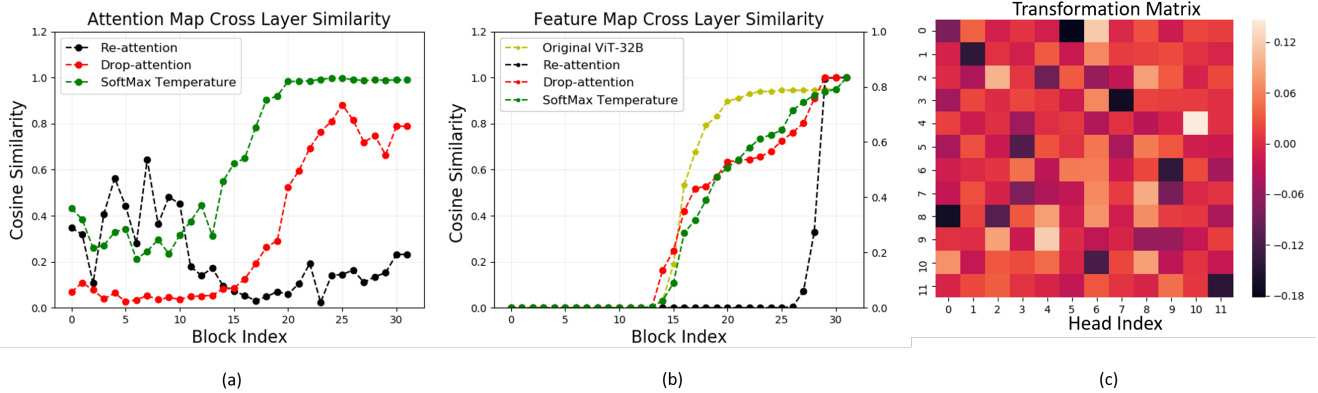
Figure 8: (a) Adjacent block attention map similarity with different methods. As can be seen, our proposed Re-attention achieves low cross layer attention map similarity. (b) Cosine similarity between the feature map of the last block and each of the previous block. (c) Visualization of transformation matrix of the last block.

the attention maps for different blocks, the similarity between attention maps could be reduced. The impacts on the attention maps and the output features of each block are shown in Fig. 8(a-b). It is observed that dropping attention does reduce the cross layer similarity of the attention maps. However, the similarity among features are not reduced by much. This is because the difference between attention maps comes from the zero positions in the generated mask. Those zero values do reduce the similarity between attention maps but not contribute to the features. Thus, the improvement is still not significant as shown in Tab. 5.

Advantages of Re-attention: Our proposed Re-attention brings more significant improvements over the temperature-tuning and the attention-dropping methods. This is because both adding temperature and dropping attention are regularizing the distribution of the originally over-smoothed self-attention maps, without explicitly encouraging them to be diverse. However, our proposed Re-attention mechanism uses different heads (whose attention maps are dissimilar) as basis and re-generate the attention maps via the transformation matrix Θ. This process incorporates the inter-head information communication and the generated attention maps can encode richer information. It is worth noting that the original MHSA design can be thought as a special case of Re-attention with an identity transformation matrix. By making Θ learnable for each block, an optimized pattern could be learned end to end. As shown in Fig. 8(c), the learned transformation matrix assigns a diverse set of weights for each newly generated head. It clearly shows that the combination for each new heads takes different weights from the original heads in the re-attention process and thus reduces the similarity between their attention maps. As shown in Fig. 8(a), our proposed Re-attention achieves the lowest cross layer attention map similarity. Consequently, it also reduces the feature map similarity across layers as

shown in Fig. 8(b).

5.4. Comparison with other SOTA models

With Re-attention, we design two ViT variants, *i.e.*, DeepViT-S and DeepViT-L, based on the ViT with 16 and 32 transformer blocks respectively. For both models, we use Re-attention to replace the self-attention. To have a similar number of parameters with other ViT models, we adjust the embedding dimension accordingly. The hidden dimensions of DeepViT-S and DeepViT-L models are set as 396 and 408 respectively. More details on the model configuration are given in the supplementary material. Besides, motivated by [42], we add three CNN layers for extracting the token embeddings, using the same configurations as [42]. It is worth noting that we do not use the optimized training recipes and the repeated augmentation as [37] for training our models. The results are shown in Tab. 6. Clearly, our DeepViT model achieves higher accuracy with less parameters than the recent CNN and ViT based models. Notably, without any complicated architecture change as made by T2T-ViT [42] (adopting a deep-narrow architecture) or DeiT [37] (introducing token distillation), simply using the Re-attention makes our DeepViT-L outperforms them by 0.4 points with even smaller model size (55M vs. 64M & 86 M).

6. Conclusion

In this work, we found the attention collapse problem of vision transformers as they go deeper and propose a novel Re-attention mechanism to solve it with minimum amount of computation and memory overhead. With our proposed Re-attention, we are able to maintain an increasing performance when increasing the depth of ViTs. We hope our observations and methods could facilitate the development of vision transformers in future.

Table 6: Top-1 accuracy comparison with other SOTA models on ImageNet. * denotes our reproduced results. ⋆ denotes our model trained with training recipes used in DeiT [37].

Model	Params. (M)	MAdds (G)	Acc. (%)
ResNet50 [9]	25	4.0	76.2
ResNet50*	25	4.0	79.0
RegNetY-8GF [27]	40	8.0	79.3
Vit-B/16 [7]	86	17.7	77.9
Vit-B/16*	86	17.7	79.3
T2T-ViT-16 [42]	21	4.8	80.6
DeiT-S [37]	22	–	79.8
DeepVit-S (Ours)	27	6.2	81.4
DeepVit-S⋆ (Ours)	27	6.2	82.3
ResNet152 [9]	60	11.6	78.3
ResNet152*	60	11.6	80.6
RegNetY-16GF [27]	54	15.9	80.0
Vit-L/16 [7]	307	–	76.5
T2T-ViT-24 [42]	64	12.6	81.8
DeiT-B [37]	86	–	81.8
DeiT-B*	86	17.7	81.5
DeepVit-L (Ours)	55	12.5	82.2
DeepVit-L⋆ (Ours)	58	12.8	83.1
DeepVit-L⋆ ↑ 384 (Ours)	58	12.8	84.3

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[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] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] 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.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[9] 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.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[11] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[12] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. *arXiv preprint arXiv:2103.02907*, 2021.

[13] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[14] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3464–3473, 2019.

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

[16] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *arXiv preprint arXiv:1811.06965*, 2018.

[17] Zihang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Convbert: Improving bert with span-based dynamic convolution. *arXiv preprint arXiv:2008.02496*, 2020.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[20] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Changhu Wang, and Jiashi Feng. Improving convolutional networks with self-calibrated convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10096–10105, 2020.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[27] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[28] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[29] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.

[32] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[34] 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.

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

[36] Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*, 2019.

[37] 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.

[38] 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.

[39] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Standalone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2020.

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[42] 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.

[43] 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.

[44] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[45] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.

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

[47] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[48] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020.

[49] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020.

[50] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[51] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

[52] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. *ECCV, August*, 2, 2020.

A. Experiment Implementation Details

Attention reuse: As shown in Fig. 3(b) and Tab. 3 in the main paper, the vision transformers with 24 blocks and 32 blocks have 11 and 15 blocks with similar attention maps, respectively. To verify the effectiveness of the attention maps from those blocks, we directly force those blocks to share the same attention map as the last 'unique' block as defined in Sec. 5.2. Specifically, we take the attention map of the last 'unique' block and denote it as $\mathbf{A}_{unique}$. For all the following blocks, the attention output is calculated by:

$$\text{Attention}(Q, K, V) = \text{Norm}(\Theta^{\top}\mathbf{A}_{unique})V, \quad (5)$$

where Θ is used to simulate the small variance between attention maps across layers since they are not identical. Norm is batch normalization used to adjust the variance across layers. As shown in Tab. 3, for a ViT with 32 blocks, forcing the top 15 blocks to share the same attention map causes negligible degradation on the classification accuracy on ImageNet. This proves that adding those blocks do not contribute to the accuracy improvement.

Training loss: We use the cross-entropy (CE) loss as the training loss for all experiments. To minimize the similarity of the attention maps across layers, we add the cosine similarity between layers into the loss function when training the model.

$$\text{Loss}_{train} = \text{Loss}_{CE} + \lambda \sum_{l=0}^{B} \text{Similarity}(\mathbf{A}^l, \mathbf{A}^{l+1}) \quad (6)$$

where $\text{Similarity}(\mathbf{A}^l, \mathbf{A}^{l+1})$ denotes the cosine similarity between layer l and $l+1$ and $\mathbf{A}^l$ denotes the attention map of layer l. B denotes the number of bottom blocks used for regularization and is a hyper-parameter. We set B to 4, 8 and 12 for training ViT models with 16, 24 and 32 blocks respectively.

Table 7: Structural hyper-parameter of DeepViT-S and DeepViT-L. Note that the embedding dimension is slightly larger than the baseline models. This is to adjust the size of the model to have a comparable size with other methods for a fair comparison.

Model	#Blocks	#Embedding	MLP size	Split ratio
DeepViT-S	16	396	1188	11-5
DeepViT-L	32	420	1260	20-12

B. DeepViT architecture design

As observed in Fig. 3(a), the attention maps of the transformer blocks become similar only at the top blocks. Thus,
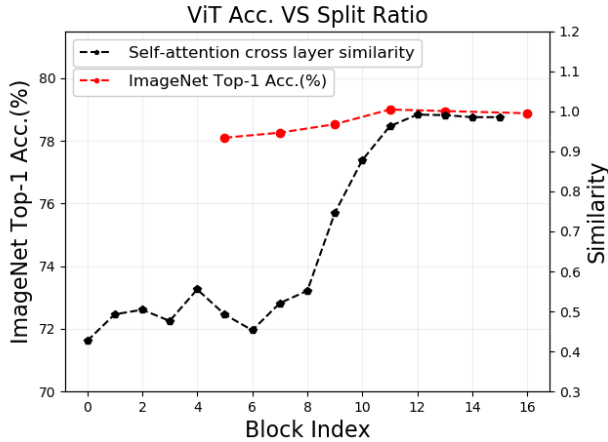
Figure 9: ViT classification accuracy with Re-attention applied on different number of blocks. The black dotted line denotes the cosine similarity ratio between adjacent blocks of the original ViT model with 16 blocks. The red dotted line denotes the top-1 classification accuracy on ImageNet. The accuracy of the model with blocks index k denotes that the Re-attention is applied on top $(16 - k)$ blocks.

it is not necessary to apply re-attention to all blocks. To study the optimal number of blocks with re-attention, we conduct a set of experiments on a ViT model with 16 transformer blocks. For each experiment, we only apply re-attention on the top K blocks where K ranges from 5 to 15. The rest of the blocks are using the original transformer block structure. We train each model on ImageNet with the same set of training hyper-parameters as those for baseline models as detailed in Sec. 5 in the main paper. The results are shown in Fig. 9.

It is observed that, as the number of re-attention blocks varies, the top-1 classification accuracy changes correspondingly. The highest accuracy appears at the position where the number of re-attention blocks is the same as the number of similar attention map blocks. Based on this observation, we define the architecture of DeepViT-S and DeepViT-L with 5 and 12 re-attention blocks respectively. Detailed configurations are shown in Tab. 7. Note that we adjust the embedding dimension to have a comparable size with other methods.

C. Impacts of hyper-parameters

In the main paper, all experiments are run with the same set of training hyper-parameters as the one used for reproducing ViT models. However, as shown in [37], an improved training recipe could improve the performance of ViT models significantly. In Tab. 8, we present the performance of DeepViT-S and DeepViT-L with the same set

of training recipes as DeiT except that we do not use repeated augmentation. In Tab. 8, it is clearly shown that the performance of DeepViT could be further improved with optimized training hyper-parameters.

Table 8: DeepViT model with different training recipes. * denotes the model trained with DeiT [37] training recipes.

Model	Params. (M)	MAdds (G)	Acc. (%)
DeiT-S [37]	22	–	79.8
DeiT-S (KD) [37]	22	–	81.2
DeepVit-S (Ours)	27	6.2	81.4
DeepVit-S* (Ours)	27	6.2	82.3
DeiT-B [37]	86	17.7	81.8
DeiT-B (KD) [37]	86	17.7	83.4
DeepViT-L (Ours)	55	12.5	82.2
DeepViT-L* (Ours)	58	12.8	83.1

D. Similarity calculation

Cosine similarity between layers To measure the similarity between the attention maps, we define the similarity $S^{p,q}$ between the attention maps of two layers, p and q, as the ratio of the number of similar vector pairs to the total number of pairs between two attention maps:

$$ S(p,q) = \frac{\sum I}{|M^{p,q}|}, \quad I_{i,j} = \begin{cases} 1, & \text{if } M_{i,j}^{p,q} > \tau \\ 0, & \text{otherwise} \end{cases} \quad (7) $$

where τ is a hyper-parameter and used as a threshold for deciding similar vectors[2].

Definition of similar blocks A block is counted as a similar block if the similarity between it's attention map and the adjacent block's attention map is larger than 80%. To measure the block similarity for a ViT model with B blocks, we take the ratio of the number of similar blocks to the total number of blocks as a measurement.

---

[2]0.5 is selected as a threshold for visualization purpose in this paper