Accumulated Trivial Attention Matters in Vision Transformers on Small Datasets

Xiangyu Chen[1], Qinghao Hu[2], Kaidong Li[1], Cuncong Zhong[1], Guanghui Wang[3*]
[1]Department of EECS, University of Kansas, KS, USA
[2]Institute of Automation, Chinese Academy of Sciences, China
[3]Department of CS, Toronto Metropolitan University, Toronto, ON, Canada
xychen@ku.edu, wangcs@ryerson.ca (* corresponding author)

Abstract

Vision Transformers has demonstrated competitive performance on computer vision tasks benefiting from their ability to capture long-range dependencies with multi-head self-attention modules and multi-layer perceptron. However, calculating global attention brings another disadvantage compared with convolutional neural networks, i.e. *requiring much more data and computations to converge, which makes it difficult to generalize well on small datasets, which is common in practical applications. Previous works are either focusing on transferring knowledge from large datasets or adjusting the structure for small datasets. After carefully examining the self-attention modules, we discover that the number of trivial attention weights is far greater than the important ones and the accumulated trivial weights are dominating the attention in Vision Transformers due to their large quantity, which is not handled by the attention itself. This will cover useful non-trivial attention and harm the performance when trivial attention includes more noise,* e.g. *in shallow layers for some backbones. To solve this issue, we proposed to divide attention weights into trivial and non-trivial ones by thresholds, then Suppressing Accumulated Trivial Attention (SATA) weights by proposed Trivial WeIghts Suppression Transformation (TWIST) to reduce attention noise. Extensive experiments on CIFAR-100 and Tiny-ImageNet datasets show that our suppressing method boosts the accuracy of Vision Transformers by up to* 2.3%. *Code is available at https://github.com/xiangyu8/SATA.*

1. Introduction

Convolutional Neural Networks (CNN) have dominated computer vision tasks for the past decade, especially with the emergence of ResNet [16]. Convolution operation, the core technology in CNN, takes all the pixels from its receptive field as input and outputs one value. When the layers go deep, the stacked locality becomes non-local as the receptive field of each layer is built on the convolution results of the previous layer. The advantage of convolution is its power to extract local features, making it converge fast and a good fit, especially for data-efficient tasks. Different from CNN, Vision Transformer (ViT) [11] and its variants [6, 10, 12, 27, 30, 33] consider the similarities between each image patch embedding and all other patch embeddings. This global attention boosts its potential for feature extraction, however, requiring a large amount of data to feed the model and limiting its application to small datasets.

On the one hand, CNNs have demonstrated superior performance to ViT regarding the accuracy, computation and convergence speed on data-efficient tasks, like ResNet-50 for image classification [2, 3, 25, 23], object detection [43] and ResNet-12 for few-shot learning [5]. However, to improve the performance is to find more inductive bias to include, which is tedious. The local attention also sets a lower performance ceiling by eliminating much necessary non-local attention, which is in contrast to Vision Transformers. On the other hand, the stronger feature extraction ability of Vision Transformers can perfectly make up for the lack of data on small datasets. As a result, Vision Transformers show promising direction for those tasks.

To adapt Vision Transformers to data-efficient tasks, some researchers focus on transfer learning [20, 35, 39], semi-supervised learning and unsupervised learning to leverage large datasets. Others dedicate to self-supervised learning or other modalities to dig the inherent structure information of images themselves [4]. For supervised learning, one path is to integrate convolution operations in Vision Transformers to increase their locality. Another approach is to increase efficiency by revising the structure of Vision Transformers themselves [31]. The proposed method belongs to the second category.

The main transformer blocks include a multi-head self-attention (MHSA) module and a multi-layer perceptron (MLP) layer, along with some layer normalization, where the MHSA module is key to enriching each sequence by including long-range dependencies with all other sequences, *i.e.* attention. Intuitively, this attention module is expected

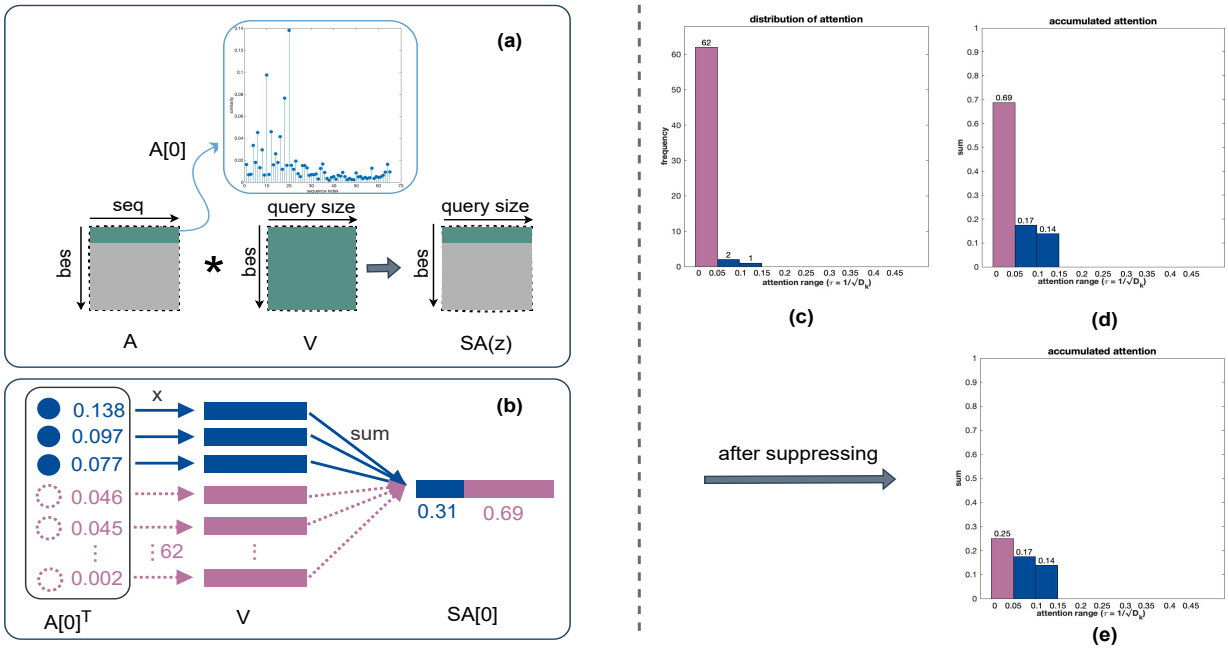

Figure 1: Our proposed SATA strategy. (a) The multi-head self-attention module in Vision Transformers. Each row in A represents attention weights corresponding to all sequences in $\mathbf{v}$. (b) A closer look at how to get the first sequence $SA[0]$ after applying attention. We set the threshold to 0.05. The blue part denotes larger attention weights and purple is for trivial ones. We get up to 62 trivial attention weights and sum up to 0.69 in the entire attention in SA[0] compared with 0.31 from similar sequences. (c) The distribution of attention weights. (d) Accumulated attention within each bin. (e) The result of suppressing trivial weights by our approach. Even if single attention trivial weight contributes little, the accumulated trivial attention is still dominating, which is harmful when the attention contains much noise as in shallow layers of some backbones.

to have larger coefficients for those sequences with higher similarity while smaller values for those less similar as the example $A[0]$ in Figure 1(a). In this way, all sequences can be enhanced by other similar sequences. However, this only considers single similarities themselves, but not their accumulation. Taking a closer look at the dot product operation on how each weighted sequence obtained in Figure 1(a), we can find it is from weighting each sequence with attention coefficients and then summing up into one sequence as shown in Figure 1(b). This is problematic when the sequence length is large and those less similar sequences are noise. When the similarities are added from all less similar sequences, the accumulated sum can be even greater than the largest similarity as in Figure 1(d) caused by the small-value but large-amount trivial attention coefficients. This means the accumulated trivial attention dominates the attention, which brings much noise to the convergence of the Transformer. As a result, the trivial attention would hinder the training of the Transformer on small datasets. To solve this rooted problem in Vision Transformers and make it better deploy on small datasets, we proposed to suppress all trivial attention and hence the accumulated trivial attention to make sequences with higher similarity dominant again.

The contributions of this paper are summarized below.

- We found the accumulated trivial attention inherently dominates the MHSA module in vision Transformers and brings many noises on shallow layers. To cure this problem, we propose Suppress Accumulated Trivial Attention (SATA) to separate out trivial attention first and then decrease the selected attention.
- We propose a *trivial weights suppression transformation (TWIST)* to control accumulated trivial attention. The proposed transformation is proved to suppress the trivial weights to a portion of maximum attention.
- Extensive experiments on CIFAR-100 and Tiny-ImageNet demonstrated up to 2.3% gain in accuracy by using the proposed method.

2. Related Work

Vision Transformer has become a powerful counterpart of CNN in computer vision tasks since its introduction in 2020 [11], benefiting from its power to capture long-term dependencies. This ability is brought by their inherent structures in ViT, including the MHSA attention module

which enhances each sequence with all other sequences, and MLP layers to model the relationships across all sequences. Including global attention also has weaknesses, like requiring large datasets to train, unlike the local attention in CNN. However, such large datasets are not easily accessible in many cases considering both time and effort cost in labeling and maintaining, *e.g*. rare diseases in the medical field. One direct solution is to search for more data, either borrowing data or knowledge from available large datasets and applying it to small datasets like transfer learning [20] and distillation [35, 39] or digging other information like self-supervised learning [24, 1, 15] and other modalities to exploit available labels [32, 33]. Another folder is to adjust the structure of transformers. For instance, integrate convolutional layers to transformers to mitigate its rely on the amount of data like CvT [38], LeViT [13], CMT [14] and CeiT [40], design efficient attention modules to replace the quadratic computation complexity MHSA as Reformer [19], Swin [27], Swin-v2 [26], Twins [6], HaloNet [36] and Cswin [10], or remove MLP layers [9].

Regarding the MHSA module in vision Transformers, previous works can be divided into two paths according to the components in the attention function, input and the function itself. For the input of MHSA module, *i.e*. $\mathbf{qk^T}$, Swin [27] and Swin-v2 [26] calculate attention within windows instead of full sequences, CvT [38] uses convolutional layers to replace the linear layers to get $\mathbf{kqv}$. And there are also some works argue that the Softmax function in original vision transformers can be revised (*e.g*. adding learnable temperature [23]) or even replaced with other functions (*e.g*. l_1 norm in SimA [20] and Gaussian kernel in SOFT [29]). In this work, we post-process the results after the Softmax function, which can be categorized into the attention function folder.

The attention module is designed to focus on more alike sequences and less on different ones. This is based on the premise that all sequences are clear and include little noise. In this way, attention can enhance each sequence with alike sequences and useful signals get emphasized. However, this does not hold true when the features contain much noise. As shown in Figure 1, the example attention for one sequence $A[0]$ looks reasonable, with less similar attention weights but many trivial weights. However, when we check the sum of all trivial weights, it is far greater than the maximum attention. Just imagine all these sequences assigned trivial attention are harmful sequences, the accumulated attention from trivial weights is dominating the whole attention and even covers it. This happens when attention contains too much noise, like some shallow layers as mentioned in [37]. For shallow layer features, the image is a natural signal and has low information density. In other words, the image itself contains much noise, like the background of an object. This noise can even extend to several shallow layers due to the limitation of current feature extraction models, making shallow layers contain much more noise than deeper ones. However, determining the boundary for "shallow" is difficult since it is dependent on the depth of the model, feature extraction of the model, and the noisy degree of datasets. Thus, we designed a learnable suppressing scale s for all layers, avoiding finding this boundary by adding little computation.

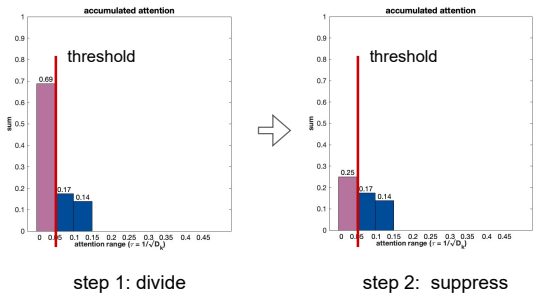

Figure 2: Our proposed suppressing method SATA. Step 1: divide trivial and non-trivial weights (0.05 in this example). Step 2: apply TWIST transformation on trivial weights.

3. Methodology

This section first introduces the MHSA module in Vision Transformers and its limitations, followed by the proposed suppressing steps, setting a threshold to separate out trivial attention coefficients and then decreasing their sum as shown in Figure 2.

3.1. Revisit MHSA

MHSA modules [11] in vision Transformers is the key to enriching sequence embedding in capturing long-range dependencies. To get this, input $\mathbf{z} \in \mathbb{R}^{\mathbf{N}\times\mathbf{D}}$, where N is the length of sequence and D is the dimension of sequence, is first passed through linear layers to get $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$. Then calculate the attention A to weight each sequence.

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{\mathbf{qkv}} \tag{1}$$

$$A = softmax(\mathbf{qk^T}/\sqrt{\mathbf{D_h}}) \tag{2}$$

Finally, compute a weighted sum across all sequences to get the final enhanced sequences, $SA(z)$ in Figure 1(a).

$$SA(\mathbf{z}) = \mathbf{A}\mathbf{v} \tag{3}$$

Taking a closer look at the detailed operations in Equation (3) as in Figure 1(b), to get the first row $SA[0]$, V is first weighted by all elements of the first row of A, and the weighted sequences are summed up to one sequence. One

example of the attention weights $A[0]$ on the CIFAR-100 dataset can be found in Figure 1(a), where several weighs are larger, indicating the corresponding sequence is more similar to the current query sequence and hence with higher attention, while most of them are small indicating less similarity and importance.

We further explore the statistics of $A[0]$, the distribution of attention as in Figure 1 (c), where each bar denotes the number of attention weights in each similarity range. As in the example, up to 62 attention weights are below 0.05 indicating less similar sequences, while only 3 weights are greater than it. Surprisingly, after calculating the sum of attention weights in each bar and getting the graph in Figure 1 (d), the sum of trivial weights (below 0.05) is far larger than the sum of important weights, *i.e*. SA[0] is composed of more information from trivial sequences than non-trivial sequences. In other words, trivial attention is dominating the MHSA altogether. This also means the MHSA modules bring more noise than information, making it converge slowly, especially on small datasets. To handle this, we need to first set a threshold to separate trivial/non-trivial attention weights and then suppress trivial ones.

3.2. Divide

The first step is to get all trivial attention weights before suppressing them. Here we use a suppression threshold to divide the attention weights into trivial and non-trivial ones. Those attention weights below the suppression threshold are regarded as trivial weights, otherwise as non-trivial weights. However, there are also some choices to set the threshold.

Relative or absolute? A relative threshold is a portion t of a value, *e.g*. the maximum attention weight x_m in a row. *i.e*.

$$threshold = t * x_m \tag{4}$$

Compared to this, an absolute threshold is a given value t. *i.e*.

$$threshold = t \tag{5}$$

Notice that the absolute attention weights depend on the length of the sequence, the function to get it (*e.g*. temperature in Softmax function), datasets and so on. We set it a relative threshold as in Equation (4), where t is the relative scale. This dividing process can be achieved by multiplying the original attention with a mask M with '1' for trivial positions and '0' for nontrivial ones as below:

$$M = \begin{cases} 1, & attention \leq t * x_m \\ 0, & elsewhere \end{cases} \tag{6}$$

Then the final attention after normalization can be obtained by multiplying the original attention with the mask M as below.

$$A' = M \odot A \tag{7}$$

where $\odot$ denotes element-wise product.

3.3. Suppress

To suppress the sum of trivial attention, we transform each trivial attention weight from x_j to x'_j and call this transformation as TWIST.

Lemma 1. *Given n positive attention weights $x_1 + x_2 ... + x_k + x_{k+1} + ... + x_{n-1} + x_m = 1$, where $x_1, ..., x_k$ are trivial weights, less than a threshold T, x_m is the maximum weight and $x_{k+1}, ..., x_{n-1}$ are the rest of weights. If*

$$x_j{}' = \frac{x_j^2}{\sum_{i=1}^{k} x_i}, j = 1...k \tag{8}$$

Then,

$$\sum_{i=1}^{k} x_i{}' \leq x_m \tag{9}$$

and

$$x'_j \leq x_j, j = 1..k \tag{10}$$

Proof. Since for all x_j, $0 < x_j \leq x_m \leq 1$ where $j = 1...k$, we have $x_j^2 \leq x_j * x_m$. Then

$$\sum_{j=1}^{k} x_j^2 \leq (x_1 + x_2 + ... + x_j) * x_m \tag{11}$$

Dividing both sides by the sum yields

$$\sum_{j=1}^{k} x_j^2 / (x_1 + x_2 + ... + x_j) \leq x_m \tag{12}$$

or

$$\frac{\sum_{j=1}^{k} x_j^2}{\sum_{i=1}^{k} x_i} \leq x_m \tag{13}$$

Rewrite the numerator,

$$\frac{x_1^2 + x_2^2 + ... + x_k^2}{\sum_{i=1}^{k} x_i} \leq x_m \tag{14}$$

which is exactly Equation (9) after substituting Equation (8).

To prove Equation (10), we only need to prove:

$$\frac{x_j^2}{\sum_{i=1}^{k} x_i} \leq x_j, j = 1...k \tag{15}$$

Dividing both sides by positive value x_j yields

$$\frac{x_j}{\sum_{i=1}^{k} x_i} \leq 1, j = 1...k \tag{16}$$

As x_j is one of the items in the denominator, Equation (16) holds true. □

	lr1 (model)	lr2 (CIFAR-100)	lr2 (T-ImageNet)
ViT	0.003	$7e^{-5}$	0.001
PiT	0.001	0.001	$3e^{-4}$

Table 1: Learning rates for model (lr1) and suppressing scale s (lr2) on CIFAR-100 and Tiny-ImageNet.

Lemma 1 means if we want to make the sum of trivial weights less than the maximum weight, we can simply transform x_j to ${x_j}'$ based on Equation (8), and attention weights after transformation are always no more than original weights. We can also add a scale s on both side of Equation (14) to make the sum smaller than a portion s ($s \geq 0$) of the maximum. As a result, our final transformation on $x_1, ...x_k$ to suppress the accumulated trivial attention weights is

$$ {x_j}' = s * \frac{x_j^2}{\sum_{i=1}^{k} x_i}, j = 1...k \tag{17} $$

where the suppressing scale s is learnable. We name this transformation in Equation 17 as *trivial weights transformation (TWIST)*. This transformation can guarantee

$$ \sum_{i=1}^{k} x_i' \leq s * x_m \tag{18} $$

4. Experiments

This section presents experiment settings, results and discussions after implementing the suppressing of Vision Transformers.

4.1. Settings

We perform image classification on small datasets, including CIFAR-100 [21] and Tiny-ImageNet [22]. CIFAR-100 includes 60,000 images with size 32×32, 50,000 for train split and 10,000 for validation split. Tiny-ImageNet has 100,000 and 10,000 64×64 images for train and validation split respectively. Following settings in [23], we perform data augmentations including CutMix [41], Mixup [42], Auto Augment [7], Repeated Augment [8], regularization including random erasing [44], label smoothing [34] and stochastic depth [18]. Optimizer is also AdamW [28]. The batch size is 128 and all models are trained for 100 epochs on one A100 GPU. The learning rate of model is set to 0.003 for ViT [11] and 0.001 for PiT [17]. The suppressing scale s is initialized to 0.5 and its learning rate can be found in Table 1 *lr2*. The threshold coefficient t is fixed to 0.05 for PiT on CIFAR-100 and 0.1 for all other experiments.

Model	Param (M)	CIFAR-100	T-ImageNet
ResNet56*	0.9	76.36	58.77
ResNet110*	1.7	79.86	62.96
EfficientNet B0*	3.7	76.04	66.79
ViT	2.8	73.70	56.45
SATA-ViT (ours)	2.8	**74.93**(*+1.23*)	**58.77**(*+2.32*)
PiT	7.1	72.31	57.87
SATA-PiT (ours)	7.1	**73.52**(*+1.21*)	**58.15**(*+0.28*)

Table 2: Classification results on CIFAR-100 and Tiny-ImageNet dataset. Top 1 accuracy (%) is reported.

4.2. Integrating with Vision Transformers

To evaluate the effect of our proposed suppressing method, we integrate it with both the original ViT [11] and PiT [17] following the scale for small datasets in [23], where the patch size is set to 4 for CIFAR-100 and 8 for Tiny-ImageNet, resulting in the same number of tokens 64 and 1 class token. From the results in Table 2 we see that the accuracy for ViT is increased by 1.23% on CIFAR-100 and up to 2.32% on Tiny-ImageNet by integrating our trivial attention suppressing module. It also boosts PiT on both datasets with up to 1.21% on CIFAR-100. These improvements demonstrate that taking care of trivial attention weights explicitly is necessary and suppressing them can improve performance. Besides this, examining the effect of our method on a different scale of tokens may be interesting future work.

4.3. Ablation study

To verify how each module works, we decompose each component in the proposed suppressing module using ViT on Tiny-ImageNet. Specifically, to understand the necessity of suppressing, we let s be a hyperparameter as t and perform a grid search on both hyperparameters. The best accuracy and its search result are listed in Table 3. Comparing row 2 with row 0 in Table 3, we observe that suppressing brings 1.24% more accuracy to ViT after grid search. In addition, we also set suppressing scale s to 0 and find that accuracies for most threshold t are near 56.45% when no suppressing exists as shown in Table 4. This indicates that those trivial attention weights are still helpful.

Grid Search. For grid search, we select s from $[1, 0.75, 0.5, 0.25, 0.1, 0]$ and t from $[0.1, 0.05, 0.025, 0.01, 0]$. The learning rate is 0.003, the same as ViT on CIFAR-100 and Tiny-ImageNet when no suppressing exists. In Table 4, we can find the best result is from $s = 0.75$ and $t = 0.1$ on Tiny-ImageNet, while it is $s = 0$ and $t = 0.075$ for ViT on CIFAR-100 according to 5, which means we get a better performance when removing those attention directly. We also select the

index	suppress	threshold	Top 1
0	-	-	56.45
1	0	0.075	57.47
2	0.75	0.1	57.69
3	learnable	0.1	**58.77**

Table 3: Ablation study. In this table, we compare suppressing with grid search s, suppressing to 0, suppressing with learnable s and no suppressing.

t \ s	1	0.75	0.5	0.25	0.1	0
0.1	56.43	**57.69**	56.22	57.30	56.46	56.51
0.075	56.79	57.06	56.74	56.29	56.99	57.47
0.05	56.11	56.72	56.47	56.87	57.24	56.47
0.025	56.06	56.85	56.71	56.65	56.95	56.31
0.01	57.23	56.31	56.63	56.45	56.44	56.55
0	56.45					

Table 4: Grid search on s and t for ViT on Tiny-ImageNet dataset. The baseline without suppressing is the last row when $t = 0$ and the last column denotes removing the attention directly.

t \ s	1	0.75	0.5	0.25	0.1	0
0.1	74.61	74.51	74.46	74.07	74.42	74.50
0.075	74.81	74.01	73.89	74.65	73.97	**74.75**
0.05	74.71	74.18	74.82	74.46	74.32	74.96
0.025	74.34	73.39	73.93	74.34	74.21	73.86
0.01	74.15	74.75	74.60	74.15	74.49	73.71
0	73.70					

Table 5: Grid search on s and t for ViT on CIFAR-100 dataset. The baseline without suppressing is the last row when $t = 0$ and the last column denotes removing the attention directly.

initialization values for learnable s from their average performance. From both Table 4 and Table 5, we can see that the accuracies are increased in most cases with suppressing compared with the baseline when no suppressing happens, *i.e.* when the last row $t = 0$ in both tables. The last column in Table 4 shows that deleting the attention will hurt the performance most of the time on Tiny-ImageNet while it helps all the time on CIFAR-100. We also observe that the relationship between s and t is complicated, neither linear nor inverse. This is reasonable since both parameters are highly correlated.

Learnable s. Learned s of ViT on both CIFAR-100 and Tiny-ImageNet can be found in Figure 3. Note that s denotes the suppressing scale to the maximum attention for each sequence. In Table 3 we can see, for the first 2 layers on Tiny-ImageNet and the first 4 layers on CIFAR-100 dataset, the sum of trivial weights is ensured to be below the maximum. While for deep layers, the sum of trivial attention can be scaled up to several times the maximum. This is reasonable since for deeper layers, features contain less noise, and hence suppressing trivial attention weights is no longer necessary. Instead, another function of our SATA works is to adjust the distribution of attention by increasing trivial attention weights to be more comparable with the maximum and hence influence the distribution.

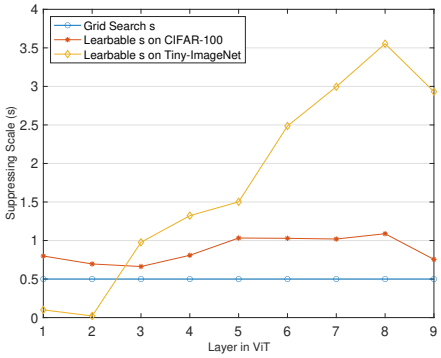

Figure 3: Learnable s vs. fixed s from grid search for ViT.

4.4. Comparison with different normalization

Softmax with temperature. The original Softmax can be denoted by

$$A = softmax(\tau \mathbf{q}\mathbf{k}^{\mathbf{T}}) \tag{19}$$

where τ is the temperature to control the scale of the Softmax function. In the original ViT [11], the normalization of attention modules is

$$A = softmax(\frac{\mathbf{q}\mathbf{k}^{\mathbf{T}}}{\sqrt{D_h}}) \tag{20}$$

where they use $1/\sqrt{D_h}$, the dimension of the head, as the temperature τ of the Softmax function to adjust the distribution of attention weights. The higher the temperature τ, the sharper the Softmax function as shown in Figure 4. Figure 4 shows that small values become even smaller and large ones are even larger when increasing the temperature of Softmax function from $\tau = 1$ to $\tau = 2$. This can also enlarge the ratio of larger values to smaller values, which is similar to the effect of our proposed suppressing trivial weights. However, increasing the temperature of Softmax cannot solve this. More specifically, it also brings side effects along with the suppression of trivial attention weights.

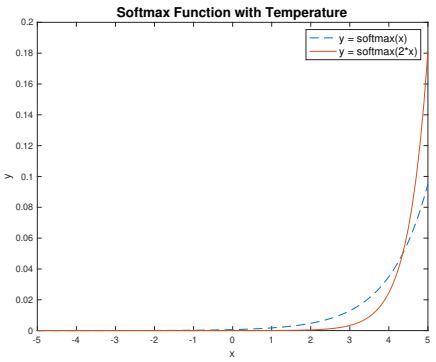

Figure 4: Softmax function with temperature. With the increase of temperature, the difference between large values and smaller values gets enlarged by decreasing smaller values and increasing greater ones.

To check this, we first conduct experiments by setting different temperatures. The results can be found in Table 6. Table 6 shows that, compared with the Softmax with default temperature $\tau = 1/\sqrt{D_h}$ in original ViT, increasing the temperature to $2\times$, $3\times$ and $4\times$ all improves the performance on Tiny-ImageNet dataset. And the best result is from $\tau = 4/\sqrt{D_h}$ Tiny-ImageNet.

In summary, Softmax with higher temperature can mitigate the dominating accumulated trivial attention effect partly at the cost of changing the distribution of sensitive larger attention weights. While our proposed suppressing method decouples trivial and non-trivial attention weights to solve the dominating effect, making us available to take advantage of both adjusting trivial attention weights according to noise level and adjusting the distribution of non-trivial attention weights with higher temperature freely. The results to build our SATA on softmax with temperature are shown in Table 6. According to Table 6, the performance for Softmax with temperature is improved with the increase of τ, while it decreases after applying our proposed module. In most cases *e.g.* $\tau = 1\times, 2\times, 3\times$ of default temperature, SATA module further boosts the accuracy. This indicates that the performance increase of Softmax with temperature is from the enlargement of the gap between larger and smaller values, which is good for shallow layers while harming deeper layers. Our module can adjust trivial attention weights in both situations, noisy or noiseless, and especially when both happen in the same model while requiring different handling.

Diagonal suppressing. This module LSA is proposed in [23] considering that the attention in diagonal is from self-attention in the MHSA module for Vision Transformers, which is not necessary since the skip connection in the MHSA module will add the attention itself with a larger ratio compared with the self-attention in the attention branch. However, this self-attention usually is larger than other attention, leaving less room for other attention to get large values. To this end, they propose to manually set the diagonal to an extremely small value, making the attention after Softmax small. As the goals of this module and our SATA module are different, we can combine both methods together to yield better attention. The results are shown in Table 7. According to this table, the performance is increased on both CIFAR-100 and Tiny-ImageNet after implementing the LSA module and our module further adds up to 0.46% on Tiny-ImageNet.

model	$\tau' = 1$	$\tau' = 2$	$\tau' = 3$	$\tau' = 4$
ViT + τ	56.45	57.20	57.21	**57.81**
ViT + τ + SATA	**58.77**	58.12	57.72	57.58

Table 6: Combining Softmax with different $\tau = \tau' \times \frac{1}{\sqrt{D_h}}$ on Tiny-ImageNet. We adjust learning rate for s after adding τ. The best results are reported.

model	CIFAR-100	Tiny-ImageNet
ViT	73.70	56.45
ViT + LSA	75.40	57.82
ViT + LSA + SATA	75.47	58.28

Table 7: Integrating with LSA module.

5. Conclusion

In this paper, we have examined the MHSA modules in Vision Transformers and discovered that attention from the trivial sequence is dominating the final attention after accumulation, affecting its performance by including more noise than information on shallow layers. This issue is not handled by the attention function *e.g.* Softmax. To solve this challenge, we propose to handle trivial weights explicitly by first separating out trivial attention weights with a relative threshold to the maximum attention and then adjusting them to a portion of the maximum attention weight. Experiments show up to 2.3% increase in accuracy, indicating this process is necessary to make the attention function work.

Acknowledgement

This work was partly supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant nos. RGPIN-2021-04244 and ALLRP 576612-22, and the United States Department of Agriculture (USDA) under grant no. 2019-67021-28996. This work was also supported in part by the National Natural Science Foundation of China (No. 62106267).

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[2] Feng Cen, Xiaoyu Zhao, Wuzhuang Li, and Guanghui Wang. Deep feature augmentation for occluded image classification. *Pattern Recognition*, 111:107737, 2021.

[3] Xiangyu Chen, Ying Qin, Wenju Xu, Andrés M Bur, Concong Zhong, and Guanghui Wang. Increasing input information density for vision transformers on small datasets. *Available at SSRN 4179882*.

[4] Xiangyu Chen and Guanghui Wang. Few-shot learning by integrating spatial and frequency representation. In *2021 18th Conference on Robots and Vision (CRV)*, pages 49–56. IEEE, 2021.

[5] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071, 2021.

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[8] 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.

[9] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. *arXiv preprint arXiv:2204.03645*, 2022.

[10] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[11] 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.

[12] Kamala Gajurel, Cuncong Zhong, and Guanghui Wang. A fine-grained visual attention approach for fingerspelling recognition in the wild. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3266–3271. IEEE, 2021.

[13] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021.

[14] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[16] 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.

[17] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021.

[18] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[19] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. 2020.

[20] Soroush Abbasi Koohpayegani and Hamed Pirsiavash. Sima: Simple softmax-free attention for vision transformers, 2022.

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

[22] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[23] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[24] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. 2022.

[25] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34:23818–23830, 2021.

[26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019.

[29] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.

[30] Wenchi Ma, Tianxiao Zhang, and Guanghui Wang. Miti-detr: Object detection based on transformers with mitigatory self-attention convergence. *arXiv preprint arXiv:2112.13310*, 2021.

[31] Krushi Patel, Andres M Bur, Fengjun Li, and Guanghui Wang. Aggregating global features into local vision transformer. *arXiv preprint arXiv:2201.12903*, 2022.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[33] Usman Sajid, Xiangyu Chen, Hasan Sajid, Taejoon Kim, and Guanghui Wang. Audio-visual transformer based crowd counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2249–2259, 2021.

[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] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[36] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021.

[37] Jun Wei, Qin Wang, Zhen Li, Sheng Wang, S Kevin Zhou, and Shuguang Cui. Shallow feature matters for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5993–6001, 2021.

[38] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[39] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. 2022.

[40] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021.

[41] 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.

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[43] Tianxiao Zhang, Bo Luo, Ajay Sharda, and Guanghui Wang. Dynamic label assignment for object detection by combining predicted ious and anchor ious. *Journal of Imaging*, 8(7):193, 2022.

[44] 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.