Vision Transformer with High Spatial Structure Sensitivity

Zhiwei Xu xuz131@mcmaster.ca
McMaster University

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

Self-attention operation, the core operation of the vision transformer (VT), is position-independent. Therefore, VT uses positional embedding to encode spatial information. However, we found that the role of positional encoding is very limited, and VT is insensitive to spatial structure. We demonstrated a significant sensitivity gap to random block shuffling and masking between VT and convolutional neural network (CNN), which indicates that VT does not learn the spatial structure of the target well and focuses too much on small-scale detail features.

We argue that self-attention should use position-dependent operations to encode spatial information instead of relying on positional embedding. We replace the linear projection of self-attention with convolution operation and use regular receptive field for each feature point, which significantly increases VT's sensitivity to spatial structure without sacrificing performance.

Keywords: Vision transformer, self-attention, spatial structure, sensitivity

1. Introduction

The convolutional neural network (CNN) and vision transformer (VT) are the two most mainstream architectures in computer vision. CNN matches image-specific inductive bias (e.g., locality, two-dimensional neighborhood structure, and translation equivariance). The convolution operation aggregates the features within the receptive field in a specific way. The output of the convolution operation is dependent on the spatial structure of the input feature map. On the contrary, VT has much less image-specific inductive bias. The self-attention operation, the core operation of the VT, is position-independent. It generates a key, query, and value for each point on the input feature maps through linear projection. The key and query are used to generate a weight matrix. Then, self-attention performs feature aggregation based on the weight matrix. During this process, the spatial structure information of feature points does not participate in the calculation. Randomly shuffling the position of feature points within the receptive field does not alter the self-attention output. To solve this problem, VTs use positional embedding (e.g., absolute positional embedding Dosovitskiy et al. (2020)) to encode spatial information. However, we observe that positional embedding does not make VTs have a sufficient positional dependency and learn the spatial structure of the target object. If we perform random block shuffling on images, which destroys the structure of objects, VTs still predict the existence of the target object with high confidence.

Figure 1 shows the vulnerability of VTs. We compare the Swin-T Liu et al. (2021) model with ResNet50 He et al. (2016) (both are pretrained on ImageNet-1K Deng et al. (2009)). They both output the correct category with high confidence (Swin-T: 87%, ResNet50: 99%)

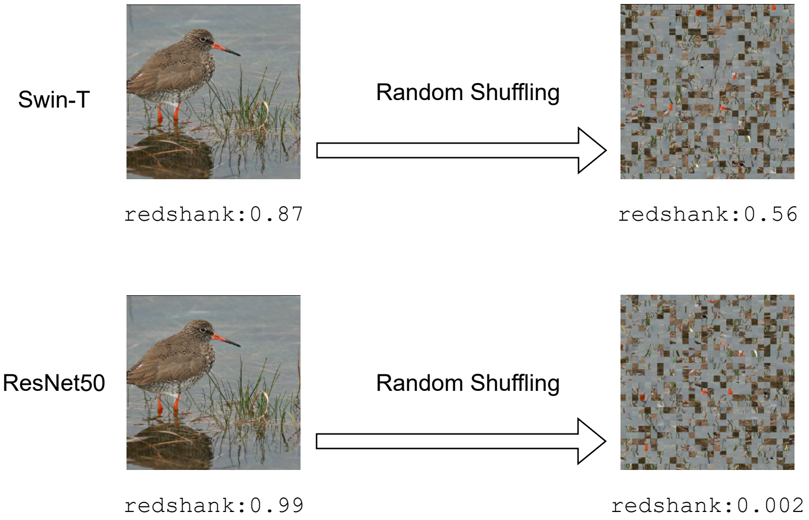

Figure 1: The sensitivity to random block shuffling of ResNet50 and Swin-T

when fed with a clear image (from the validation set of ImageNet-1K) containing a redshank. Then, we divide the image into blocks of size 8 × 8 and randomly shuffle the positions of all image blocks. ResNet50 classifies the shuffled image as redshank with a confidence of only 0.16%. However, Swin-T still classifies the shuffled image as redshank with a high confidence (56%). This indicates that VTs pay little attention to the position of features. In other words, VTs are insensitive to spatial structure.

Moreover, the insensitivity of VTs is severe. Figure 2 shows this phenomenon. We perform random block (16 × 16) shuffling on the redshank image, then paste a normal mountain bike image into the center. We feed the merged image to Swin-T; the output is redshank (85%) and mountain bike (2.2%). The disorganized redshank feature wins over the well-structured bicycle feature.

VTs learn a certain degree of spatial structure through positional embedding Jelassi et al. (2022); Raghu et al. (2021), as the prediction confidence decreases after block permutation

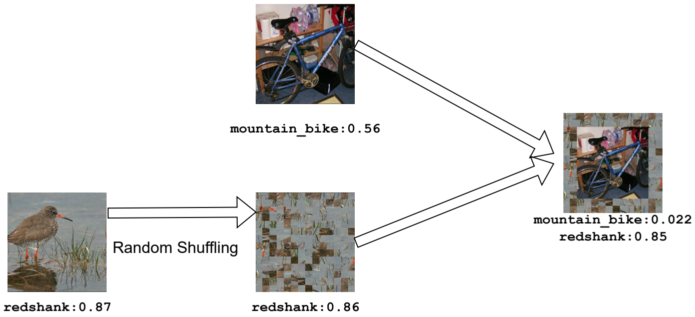

Figure 2: VT is insensitive to spatial structure. The disorganized feature wins over the well-structured features.

(see figure 1). However, a significant spatial structure sensitivity gap exists between VTs and pure CNNs (more quantitative analysis in Section IV). In fact, VT treats images as a set of patches. It focuses on learning which patches are included in the image, not the spatial structure of the image patches.

Why spatial structure sensitivity is important? We believe that vision models need to have high spatial structure sensitivity. First, low sensitivity represents abnormal generalization. After random permutation, images do not belong to their original category. VTs' high-confidence prediction on permutated images is wrong. Generally speaking, vision models should perform well on images from a similar distribution of the training dataset and make random predictions on images from a distribution different from the training set. Second, low-sensitivity models may pose security risks in real scenarios. For instance, attackers can induce the model to make the wrong prediction by adding noise blocks to the background (as shown in figure 2). For autonomous driving or facial recognition systems, this will have catastrophic consequences.

We argue that the spatial insensitivity of VTs mainly comes from the following two reasons. First, self-attention uses linear projection for query, key, and value generation. Linear projection only considers the features of the current feature point and does not encode the spatial structure of the feature map. Second, self-attention uses irregular receptive fields. Each feature point is at a different relative location inside its receptive field. Some feature points are located in the center of the receptive field, while others are at the edges. This results in different spatial extension topologies for feature aggregation of different feature points, which prevents the model from achieving high spatial structure sensitivity. Current VTs use positional embedding to encode spatial structure information of images. However, we have shown that the function of positional embedding is very limited.

Based on the above arguments, we made the following two modifications to the self-attention operation. First, we use convolution operations with a small kernel to replace the linear projection of key generation. This encodes the local structural information around feature points into the key tensor. Second, we use regular and consistent receptive fields. All feature points are at the center of its receptive field. Our modifications do not affect the performance of VTs since the power of self-attention comes from two aspects. (1) It can model long-dependency relationships. Patches that are far apart in the image can interact with each other in the shallow layers of the model. (2) It has a more complicated feature extraction ability than convolution operation. It learns a weight matrix for feature aggregation. In fact, self-attention does not limit how query, key, and value are generated or the shape of the receptive field.

We evaluate our idea on the ImageNet-1K dataset, and the experimental results show that our solution significantly improves the spatial structure sensitivity of VTs without sacrificing their performance.

2. Related work

2.1. Convolutional neural network

CNN has a long history in computer vision. Convolution operation highly matches the inductive bias of images. With the emergence of GPUs and large vision datasets, CNN has dominated the field of computer vision and achieved astonishing results in various vision tasks (e.g., classification and object detection). The success of Alexnet Krizhevsky et al. (2017) on ImageNet is seen as the starting point for CNN's prosperity. Since then, the model has become increasingly large. Researchers improve the power of CNNs by adding more layers and increasing the number of features (e.g., VGG Simonyan and Zisserman (2014)). The emergence of ResNet He et al. (2016) brings revolutionary changes to CNN. The residual structure allows gradients to propagate very efficiently to the shallow layers of the model, making it possible to train very deep networks. Nowadays, network structures containing hundreds of convolutional layers are very common and lots of different architectures have been proposed in recent years (e.g., ResNext Xie et al. (2017), Res2Net Gao et al. (2019), ConvNext Liu et al. (2022), GoogleNet Szegedy et al. (2015), DenseNet Huang et al. (2017), EfficientNet Tan and Le (2019)). No matter how the model architecture changes, convolution has always been CNN's core operation. Since the convolution utilizes the spatial structure information of input feature maps, CNN is spatial structure sensitive.

2.2. Vision transformers

Transformer Vaswani et al. (2017) rises in the natural language processing (NLP) field. Google introduced transformer architecture into computer vision and proposed the first vision transformer named ViT Dosovitskiy et al. (2020), which treats images as patch sequences and needs to be pretrained on a vast dataset to achieve performance similar to CNN. Since then, many new vision transformers (e.g., Swin Transformer Liu et al. (2021), DeiT Touvron et al. (2021), DaViT Ding et al. (2022), Cswin Dong et al. (2022), XCiT Ali et al. (2021)) and training methods Touvron et al. (2021); Steiner et al. (2021) have been proposed. Swin Transformer is an excellent vision transformer. It performs self-attention in 7×7 windows and models the far relation of image patches through shifted window partitioning. In addition, it generates multi-scale feature maps through patch merging, which makes it suitable for fine-grained vision tasks (e.g., object detection and semantic segmentation). The vision transformer has achieved similar or even better performance than CNN on various tasks and does not need to be pre-trained on a vast dataset.

Self-attention is position-independent; the random shuffling of inputs does not change the output. Vision transformers try to solve this problem by using positional embedding. For instance, ViT merges the positional embedding into patch embedding. Swin transformer merges positional embedding into the self-attention process, and different self-attention layers learn different positional embeddings. Some work Jelassi et al. (2022); Cordonnier et al. (2019); Raghu et al. (2021) have investigated the working mechanism of positional encoding and shown that VTs learn spatial structure to some extent. They also show that VTs are less sensitive to spatial structure change than pure CNNs. For instance, Jelassi et al. (2022) shows that ViT trained on block permutated training set can still perform well on normal validation set. On the contrary, the performance of CNN decreases significantly if trained on block permutated training set.

2.3. Hybrid model

There exist hybrid models, which include both convolutional and self-attention. For example, Steiner et al. (2021) proposes stacking attention layers after convolutional layers. The shallow layers of the model are convolutional layers, and the deeper layers are attention layers. Pan et al. (2022) performs convolution and self-attention operations simultaneously inside each layer. We will show that hybrid models have a spatial structure sensitivity between pure CNN and pure VT.

3. Preliminaries

Convolution and self-attention are the core operations of CNN and VT, respectively, and result in different behaviors of the two architectures. This section reviews the working mechanism and characteristics of these two operations.

3.1. Convolution

We consider the standard convolution with kernel size k. Then the kernel $K \in R^{k\times k\times C_{in}\times C_{out}}$, where C_{in} and C_{out} are the input and output channel size. Given input feature map $F \in R^{H\times W\times C_{in}}$, the output feature map $G \in R^{H\times W\times C_{out}}$, where H, W denote the height and width (we suppose strides = 1). We denote $f_{i,j} \in R^{C_{in}}, g_{i,j} \in R^{C_{out}}$ as the feature tensors of location (i, j) on F and G respectively, where $0 \leq i < H$ and $0 \leq j < W$. Then, the convolution process can be formulated as follows:

$$g_{i,j} = \sum_{p,q} K_{p,q} f_{i+p-\lfloor k/2\rfloor, j+q-\lfloor k/2\rfloor}$$

where $K_{p,q} = K[p, q, ...] \in R^{C_{in}\times C_{out}}$ represents kernel weights with regard to the indices of position (p,q). $p, q \in \{0, 1, \ldots, k-1\}$. The convolution process can be summarized into two stages. The first stage performs linear projection on the input feature, corresponding to the formula's matrix multiplication. The second stage performs feature aggregation, corresponding to the formula's summation part. Convolution operations have the following characteristics. First, the linear projection in the first stage is position-dependent. Since $K_{p,q}$ is different for different p, q. The pixels on the input feature map undergo different linear projections. The output of convolution encodes the spatial information of the input feature map, which is why CNN is sensitive to spatial structure. Second, the feature aggregation in the second stage is primitive. Features at different locations have the same significance.

3.2. Self-attention

We consider the standard one-head self-attention operation. Similarly, we denote $F \in R^{H\times W\times C_{in}}$ and $G \in R^{H\times W\times C_{out}}$ as input and output feature map. Let $f_{i,j} \in R^{C_{in}}, g_{i,j} \in R^{C_{out}}$ denote the corresponding feature tensor of location (i, j) on F and G. Then, the standard self-attention can be summarized as following equations.

$$query_{i,j} = W_q f_{i,j}$$

$$key_{i,j} = W_k f_{i,j}$$

$$value_{i,j} = W_v f_{i,j}$$

$$g_{i,j} = softmax(f(query_{i,j}, key))value$$

where f represents the function for weight matrix generation, and the dot product is widely used as f. W_q, W_k, W_v are the projection matrices for query, key, and value generation. Self-attention can also be summarized into two stages. The first stage performs linear projection with parameter W_v on F to generate *value*. The second stage performs feature aggregation based on the weight matrix. Compared to convolution, the second stage of self-attention is more complicated. It filters features through the weight matrix.

As shown in the equation, the generation of $query, key, value$ is position-independent. For each (i, j), $query_{i,j}, key_{i,j}, value_{i,j}$ is only related to $f_{i,j}$ and is independent of pixels in other positions. This causes self-attention to be insensitive to spatial structure. Some vision transformers like the swin transformer add positional embedding B into self-attention and change the equation to:

$$g_{i,j} = softmax(f(query_{i,j}, key) + B)value$$

3.3. Receptive field

Convolution and self-attention have different receptive field designs. For convolution, the receptive field is always regular. Each feature is at the center of its receptive field, and the receptive field size is always small (e.g., 3×3). For self-attention, the receptive field is always irregular. For instance, the receptive field of ViT Dosovitskiy et al. (2020) is the whole feature map; different feature points have different locations inside the feature map. Swin Liu et al. (2021) uses 7×7 window as the receptive field. However, different feature points are at different positions within the window. The irregular receptive field of VTs prevents the model from learning the spatial structure of the target in images.

4. Spatial structure sensitivity analysis

In this section, We quantitatively analyze the sensitivity of various models to spatial structure. We use the accuracy on the block permutated test set (the spatial structure of the image has been disrupted) to measure model sensitivity. The higher the accuracy, the lower the sensitivity. All experiments are conducted on the ImageNet-1K validation dataset, and we use public pretrained models (pretrained on the ImageNet-1K training set) for analysis.

4.1. Random block shuffling

Figure 3 shows the visual effect of the random block (8×8) shuffling under different configurations. We divide each image into blocks and choose different proportions (α) of blocks for shuffling. For instance, $\alpha = 0.6$ means randomly choosing 60% blocks for shuffling, and other 40% blocks remain unchanged. The image is visually unrecognizable when $\alpha > 0.6$ since the object's structure has been destroyed.

We compare the following four models: ResNet50 He et al. (2016), Swin-T Liu et al. (2021), ViT-B/8 Dosovitskiy et al. (2020) and ViT-R26-S/32. ResNet50 is a pure CNN, Swin-T (uses positional embedding multiple times) and ViT-B/8 (uses positional embedding once) are pure vision transformers. ViT-R26-S/32 is a hybrid model. The shallow 26 layers are convolutional layers and the deeper layers are self-attention layers.

Figure 4 shows the result. We observed the following characteristics. (1)ResNet50 is very sensitive to random block shuffling. When using 8 as the block size, the accuracy of

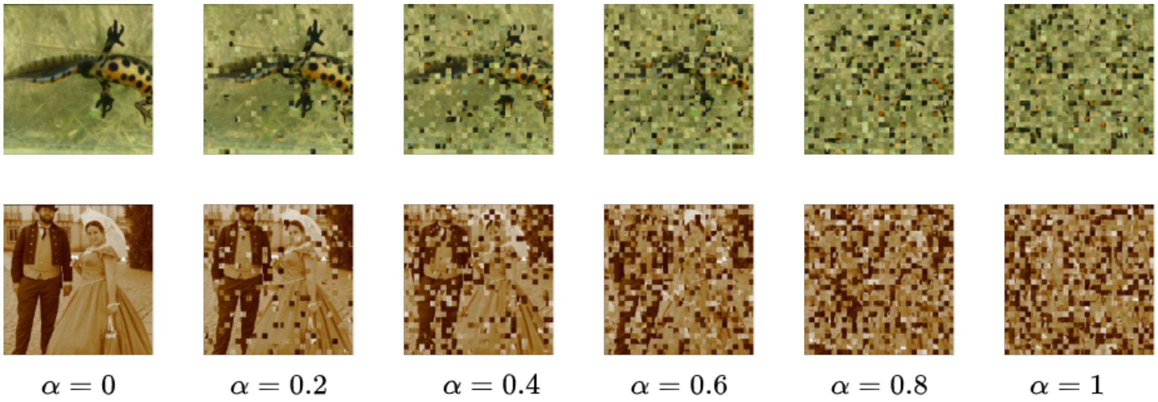

Figure 3: The visual effect of random block shuffling, α represents the ratio of blocks engaged in shuffling.

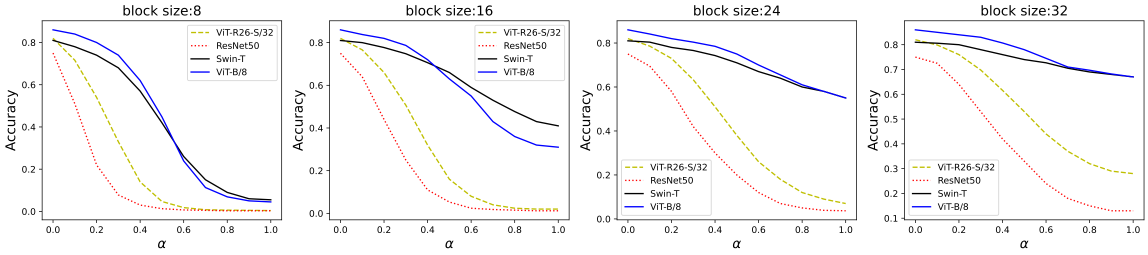

Figure 4: The sensitivity to the random block shuffling of different models under different configurations.

ResNet50 rapidly decreases as the ratio α increases. When α is about 0.4, the accuracy is close to 0. (2)Vision transformers are much less sensitive to random block shuffling than CNN. (3)There is no significant difference between ViT-B/8 and Swin-T. This indicates that using positional embedding multiple times does not increase the sensitivity. (4)The sensitivity of the hybrid model ViT-R26-S/32 is between pure CNN and pure VT. This indicates that the impact of self-attention operations on the spatial sensitivity of the model is essential. Even though ViT-R26-S/32 contains 26 convolutional layers, subsequent self-attention operations still significantly reduced the spatial structure sensitivity of the model.

Table 1: The AC of VTs on shuffled dataset (α=1.0).

Model	Block Size			
	8×8	16×16	24×24	32×32
Swin-T Liu et al. (2021)	30.2%	52.6%	60.2%	69.3%
ViT-B/8 Dosovitskiy et al. (2020)	50.5%	65.5%	76.2%	81.8%

Table 1 shows the AC (Average Confidence) of VTs on the shuffled validation set. AC measures the confidence that the model predicts correctly on the shuffled images and is calculated as:

$$AC = \frac{1}{N}\sum_{i=1}^{N} p(x_i, y_i)$$

where N represents the number of correctly classfied images, $p(x_i, y_i)$ represents the confidence that the model predict the image x_i as its true label y_i. Swin-T and ViT-B/8 both predict correct labels on shuffled images with high confidence. In addition, ViT-B/8 is more spatial structure insensitive than Swin-T as it achieves higher AC. This may be because ViT uses a larger receptive field (whole feature map), and the repeated use of positional encoding (Swin) has also played a role.

4.2. Random block masking

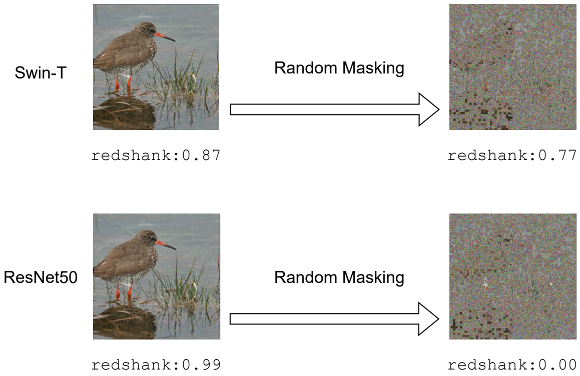

Figure 5: The sensitivity to random masking of ResNet50 and Swin-T

Another manifestation of the model's insensitivity to spatial structure is its excessive focus on detailed features. VTs can make the right prediction even when the vast majority of information in the image is masked. Figure 5 shows this phenomenon. We randomly choose 20% blocks and keep them unchanged, and the other 80% blocks are replaced with Gaussian noise (the noise has the same mean and standard deviation as the original image). We feed the image after random masking to ResNet50 and Swin-T. The two models output completely different results. ResNet50 predicts the image as redshank with confidence 0.04%. On the contrary, Swin-T predicts the redshank category with high confidence (77%).

Similarly, we quantitatively analyze the sensitivity of CNN and VT to random block masking in this subsection. Figure 6 shows the visual effect of random masking under different configurations. k represents the block size, and α represents the proportion of blocks that are masked. Figure 7 shows the results. We obtained experimental results similar to those in the previous subsection. ResNet50 is very sensitive to random masking; the classification accuracy decreases to less than 10% if more than 30% percent blocks are masked. Swin-T and ViT-B/8 are much less sensitive to random masking. When nearly 80% of the image blocks are masked, the accuracy is still high. This indicates that the VTs pay great attention to detail features.

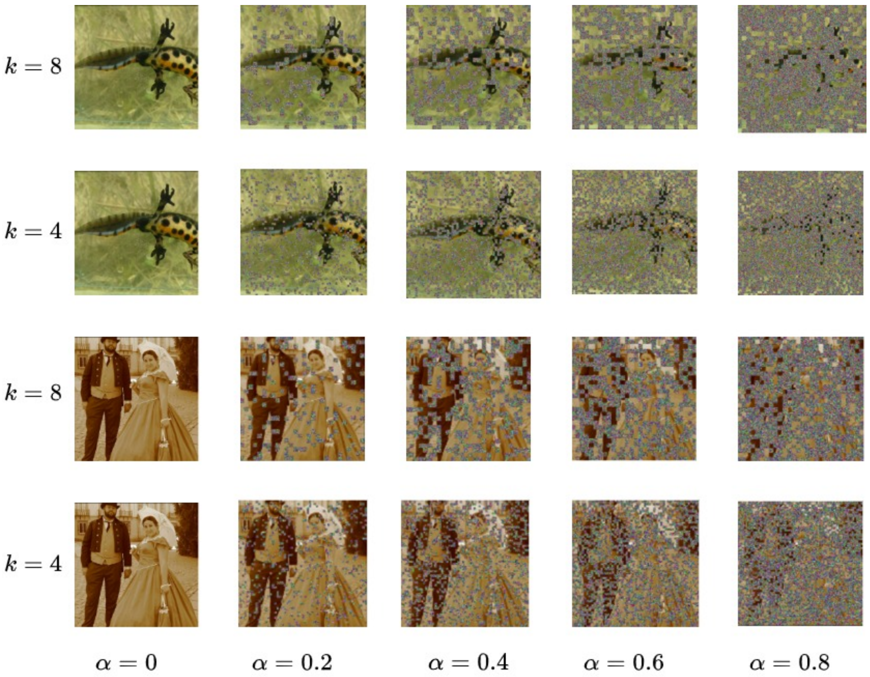

Figure 6: The visual effect of random block masking, α represents the ratio of blocks engaged in masking and k represents block size.

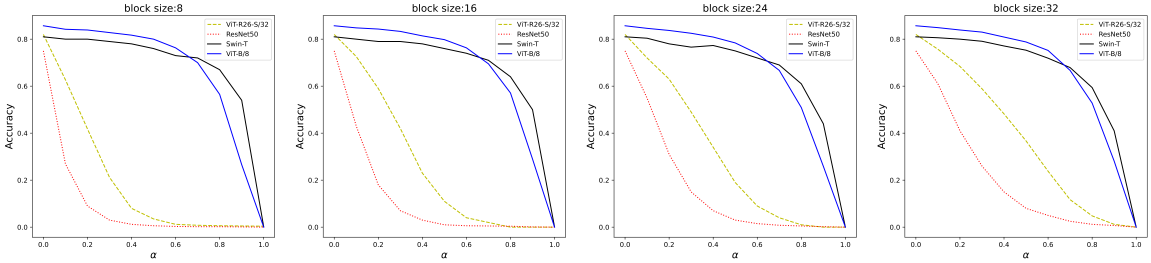

Figure 7: The sensitivity to the random block masking of different models under different configurations.

Table 2 shows the AC of Swin-T and ViT-B/8 on the masked validation set. Similarly, the two models are confident in classifying masked images, and ViT-B/8 achieves a slightly higher AC than Swin-T.

Table 2: The AC of VTs on masked validation set(α=0.8).

Model	Block Size			
	8 × 8	16 × 16	24 × 24	32 × 32
Swin-T	66.4%	67.4%	72.7%	66.1%
ViT-B/8	77.7%	78.1%	79.3%	77.5%

4.3. Dicussion

Vision Transformers regard images as a bunch of patches, and image patch is the basic unit of information. If we perform random shuffling or masking with a block size smaller than the patch size, which destroys features inside each patch, vision transformers should behave similarly to CNN.

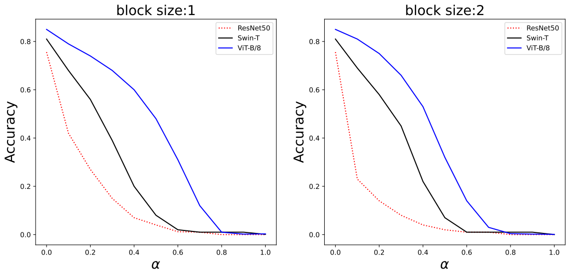

Figure 8: Random block shuffling

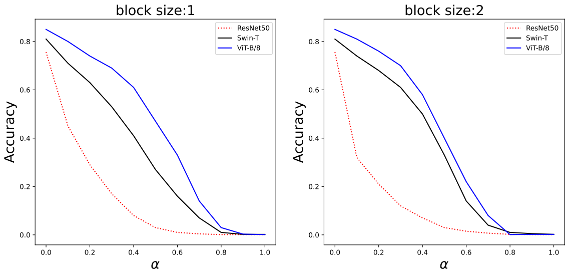

Figure 9: Random block masking

Figure 8 shows the results of random shuffling. We perform random shuffling on the validation set with block size set to 1 (pixel level shuffling) and 2. When α is small (<0.5), the accuracy of Swin-T and ViT-B/8 is still significantly higher than ResNet50 since most patches are not severely damaged. When α is large (>0.8), almost all patches have been severely damaged, the accuracy decreases to 0 like ResNet50. Figure 9 shows the result of random masking. When α is large than 0.8, the accuracy of Swin-T and ViT-B/8 decreases to 0, like ResNet50.

5. Analysis

We argue that the following two modifications can increase the spatial structure sensitivity of vision transformers. First, use convolution operation to replace the liner projection of self-attention. Second, use a regular receptive field. Each feature point needs to be at the center of its receptive field.

In this section, we evaluate the effectiveness of the proposed two modifications. All experiments are conducted on the ImageNet-1H dataset (a randomly selected subset of ImageNet-1K), which contains 100 categories. All models are trained from scratch with the same configuration. We use SGD optimizer and train all models for 60 epochs. The initial learning rate is 0.01, and we decay the learning rate with a factor of 10 at the 40th and 50th epochs. We only use random flip and random crop as data augmentation and abandon other complicated methods (e.g., Cutmix Yun et al. (2019), MixUp Zhang et al. (2017)) used by standard vision transformers training.

5.1. Replace linear projection

Since convolution operation is an excellent position-dependent operation, we try to replace the linear projection of self-attention with convolution and investigate how this changes the sensitivity of VTs to spatial structure. We train the following five models: ResNet50, Swin-T, Swin-T$_{key}$ (use 3×3 convolution for key generation), Swin-T$_{query}$ (use 3×3 convolution for query generation) and Swin-T$_{value}$ (use 3 × 3 convolution for value generation). For Swin-T$_{query}$, Swin-T$_{key}$ and Swin-T$_{value}$, we remove positional embedding.

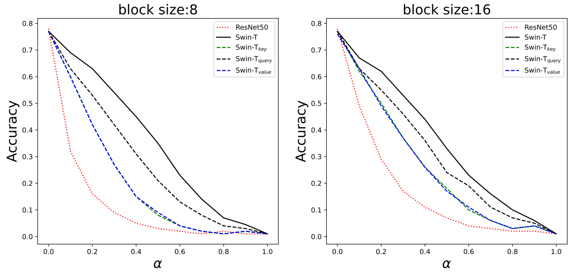

Figure 10: Random block shuffling

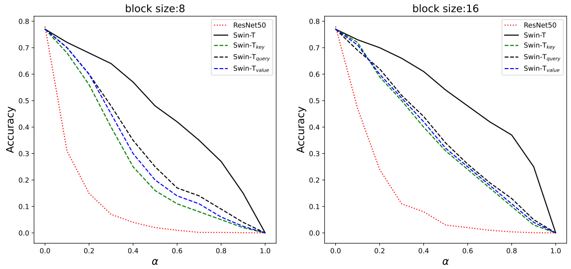

Figure 11: Random block masking

Figure 10 shows the sensitivity to random shuffling of different models. We observed the following characteristics: (1) Replacing the linear projection with 3×3 convolution does not reduce model performance. Swin-T achieves 77% classification accuracy on the validation set. The accuray of Swin-T$_{query}$, Swin-T$_{key}$ and Swin-T$_{value}$ are all about 77%. (2) Swin-T$_{query}$, Swin-T$_{key}$ and Swin-T$_{value}$ are significantly more sensitive to random shuffling than Swin-T. This proves that using position-dependent operations like convolution is better than positional embedding. In addition, key, value, and query have different degrees of influence on the model's sensitivity. The influence of the key is the strongest, while the influence of the query is the weakest. (3) There is still a sensitivity gap between ResNet50 and Swin-T$_{query}$,Swin-T$_{key}$,Swin-T$_{value}$. This may be because Swin-T uses an irregular receptive field (each feature point is not located at the center of the receptive field), which prevents the model from learning spatial structure. Figure 11 shows the sensitivity to random masking of different models under different configurations (block size). We observed similar results. Replacing linear projection with convolution significantly increases the sensitivity. Since all models are trained under the same configuration. The sensitivity differences result from the difference in model architecture. Compared to Swin-T, Swin-T$_{query}$, Swin-T$_{key}$, Swin-T$_{value}$ are less focused on the local features of the object and more dependent on the spatial structure.

5.2. Regular receptive field

In this subsection, we create a model named Swin-T_r. For each self-attention layer of Swin-T_r, we use a regular receptive field. The size of the receptive field is 7 × 7 (the same as the standard Swin-T model), and each feature point is located at the center of its receptive field. Similarly, Swin-T_r does not use positional embedding.

Figure 12 and 13 show the sensitivity of different models to random block shuffling and random block masking. We can observe that using a regular receptive field can significantly increase model sensitivity. When α is larger than 0.5, Swin-T_r has almost the same sensitivity as the pure CNN model ResNet50.

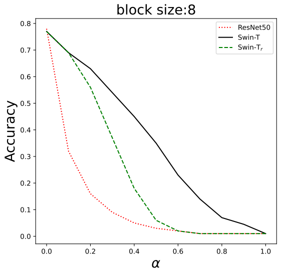

Figure 12: Random block shuffling

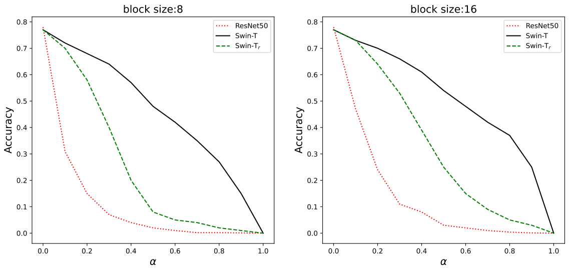

Figure 13: Random block masking

6. Experiment

In this section, we evaluate our methods on the standard ImageNet-1K dataset. We use the standard Swin-T as the baseline. Swin-T_m represents the modified model. It uses a regular receptive field (7×7) and uses 3×3 convolution for key generation. We use the same training configuration as the official Swin-T (e.g., training 300 epochs, using 1024 as batch size, using AdamW Loshchilov and Hutter (2017) as the optimizer).

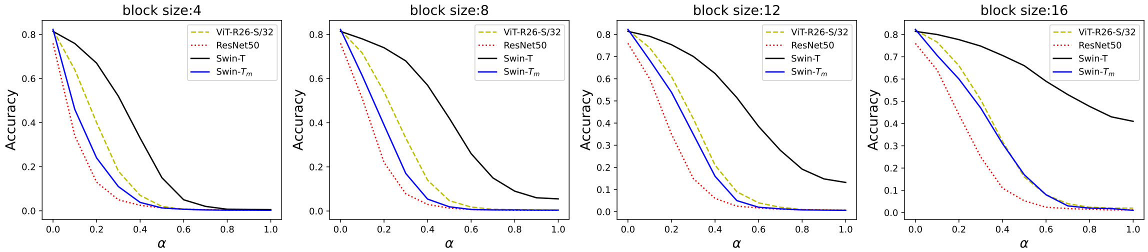

Figure 14: The sensitivity to the random block shuffling of different models under different configurations.

Spatial structure sensitivity. Figure 14 shows the sensitivity of different models. We can see that Swin-T_m has high sensitivity (close to pure CNN ResNet50, even better than hybrid model ViT-R26-S/32), much better than standard Swin-T. This validates the effectiveness of our solution.

Performance. Another interesting descovery is that Swin-T_m achieves higher performance. Its accuracy on the ImageNet validation set is 82.1%, better than standard Swin-T (81.2%). This indicates that our solution also enhances the generalization of the model.

7. Conclusion

In this paper, we point out that vision transformers are insensitive to spatial structure. The positional embedding cannot encode spatial information very well. Through experiments, we demonstrated a significant sensitivity gap to random block shuffling and masking between vision transformers and pure CNN. Vision transformers are more focused on small-scale features and do not learn the spatial structure of the object sufficiently.

We argue that self-attention should use position-dependent operations to encode spatial information and use a regular receptive field. We replaced the linear projection of self-attention with 3×3 convolution and placed each feature point at the center of its receptive field, significantly increasing the sensitivity of vision transformers to spatial structure without sacrificing performance.

References

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.

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

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European Conference on Computer Vision*, pages 74–92. Springer, 2022.

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.

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.

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

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.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

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

Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–825, 2022.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

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

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

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.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

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.

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