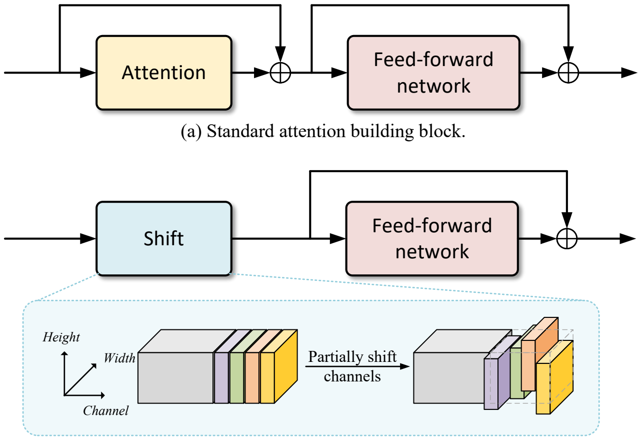
When Shift Operation Meets Vision Transformer: An Extremely Simple Alternative to Attention Mechanism

Guangting Wang[1]*, Yucheng Zhao[1]*, Chuanxin Tang[2]*, Chong Luo[2] , Wenjun Zeng[2]
[1] University of Science and Technology of China
[2] Microsoft Research Asia

Abstract

Attention mechanism has been widely believed as the key to success of vision transformers (ViTs), since it provides a flexible and powerful way to model spatial relationships. However, is the attention mechanism truly an indispensable part of ViT? Can it be replaced by some other alternatives? To demystify the role of attention mechanism, we simplify it into an extremely simple case: ZERO FLOP and ZERO parameter. Concretely, we revisit the shift operation. It does not contain any parameter or arithmetic calculation. The only operation is to exchange a small portion of the channels between neighboring features. Based on this simple operation, we construct a new backbone network, namely ShiftViT, where the attention layers in ViT are substituted by shift operations. Surprisingly, ShiftViT works quite well in several mainstream tasks, e.g., classification, detection, and segmentation. The performance is on par with or even better than the strong baseline Swin Transformer. These results suggest that the attention mechanism might not be the vital factor that makes ViT successful. It can be even replaced by a zero-parameter operation. We should pay more attentions to the remaining parts of ViT in the future work. Code is available at github.com/microsoft/SPACH.



(a) Standard attention building block.

(b) Our shift building block.

Figure 1: An illustration of our shift building block. We propose to replace the attention layer with a simple shift operation in vision transformers. It spatially shifts a small portion of the channels along four directions, and the rest of the channels remain unchanged.

Introduction

Designing backbone networks plays a fundamental role in computer vision. Since the revolutionary progress of AlexNet (Krizhevsky, Sutskever, and Hinton 2012), convolution neural networks (CNNs) have dominated this area for nearly 10 years. However, the recently developed Vision Transformers (ViTs) have shown potential to challenge this throne. The advantage of ViT was first demonstrated in image classification task (Dosovitskiy et al. 2020), where the ViT backbone outperforms its CNN counterparts by a remarkable margin. Thanks to the promising results, the flourish of ViT variants rapidly broadcasts to many other computer vision tasks, such as object detection, semantic segmentation, and action recognition.

Despite the impressive performances of recent ViT variants, it is still not yet clear what makes ViT good for visual recognition tasks. Some conventional wisdom leans to credit the success to the attention mechanism, since it provides a flexible and powerful way to model spatial relationships. Concretely, the attention mechanism leverages a self-attention matrix to aggregate features from arbitrary locations. Compared with the convolution operation in CNN, it has two significant strengths. First, this mechanism opens a possibility to simultaneously capture both short- and long-ranged dependencies, and get rid of the local restriction of the convolution. Second, the interaction between two spatial locations dynamically depends on their own features, rather than a fixed convolutional kernel. Due to such good properties, some pieces of work believe it is the attention mechanism that facilitates the powerful expressive ability of ViTs.

However, are these two advantages truly the key to success? The answer is probably NOT. Some existing work proves that, even without these properties, the ViT variants can still work well. For the first one, the fully-global dependencies may not be inevitable. More and more ViTs introduce a local attention mechanism to restrict their attention

[*] These authors contributed equally.

[†] This work was done during the internship of Guangting and Yucheng at MSRA

scope within a small local region, e.g., Swin Transformer (Liu et al. 2021b) and Local ViT (Li et al. 2021). The experiments show that the performance does not drop due to the local restriction. Besides, another line of research investigates the necessity of the dynamic aggregation. MLP-Mixer (Tolstikhin et al. 2021) proposes to substitute the attention layer with a linear projection layer, where the linear weights are not dynamically generated. In this case, it can still reach a leading performance on the ImageNet dataset.

Now that both global and dynamic properties might not be crucial for the ViT framework, what is the essential reason for the success of ViT? To figure it out, we further simplify the attention layer into an extremely simple case: *NO global scope, NO dynamics, and even NO parameter and NO arithmetic calculation*. We desire to know whether ViT can retain the good performance under this extreme case.

Conceptually, this zero-parameter alternative must rely on the handcrafted rule to model spatial relationships. In this work, we revisit the shift operation, which we believe is one of the simplest spatial modeling module. As depicted in Figure 1, the standard ViT building block consists of two parts: the attention layer and the feed-forward network (FFN). We replace the former attention layer with a shift operation, while keeping the latter FFN part untouched. Given an input feature, the proposed building block will first shift a small portion of the channels along four spatial directions, namely left, right, top, and down. As such, the information of neighboring features is explicitly mingled by the shifted channels. Then, the subsequent FFN performs channel-wise mixing to further fuse the information from neighbors.

Based on this shift building block, we construct a ViT-like backbone network, namely ShiftViT. Surprisingly, this backbone can also work well for the mainstream visual recognition tasks. The performance is on par with or even better than the strong Swin Transformer baseline. Concretely, within the same computational budgets as Swin-T model, our ShiftViT achieves a top-1 classification accuracy of 81.7% (against Swin-T's 81.3%) on ImageNet dataset. For the dense prediction task, it attains a mean average precision (mAP) score of 45.7% (against Swin-T's 43.7%) on COCO detection dataset, and a mean IoU (mIoU) score of 46.3% (against Swin-T's 44.5%) on ADE20k segmentation dataset.

Since the shift operation is already the simplest spatial modelling module, the excellent performance must come from the remaining components, e.g., the linear layers and the activation function in FFN. These components are less studied in existing work, because they look trivial. However, to further demystify the reasons why ViT works, we argue that we should pay more attentions to these components, instead of just focusing on the attention mechanism. We hope our work can shed a new light on the ViT research. As a summary, the contributions of this work are two folds:

- We present a ViT-like backbone, where the vanilla attention layer is replaced by an extremely simple shift operation. The proposed model can achieve an even better performance than Swin Transformer.
- We analyze the reasons behind the success of ViTs. It hints that the attention mechanism might not be the vital factor that makes ViT work. We should take the remaining components seriously in the future study of ViTs.

Related Work

Attention and Vision Transformers

Transformer architecture (Vaswani et al. 2017) is first introduced in the area of natural language processing (NLP). It solely adopts attention mechanism to build the connections between different language tokens. Thanks to the great performance, Transformers have rapidly dominated the NLP area and become the *de facto* standard.

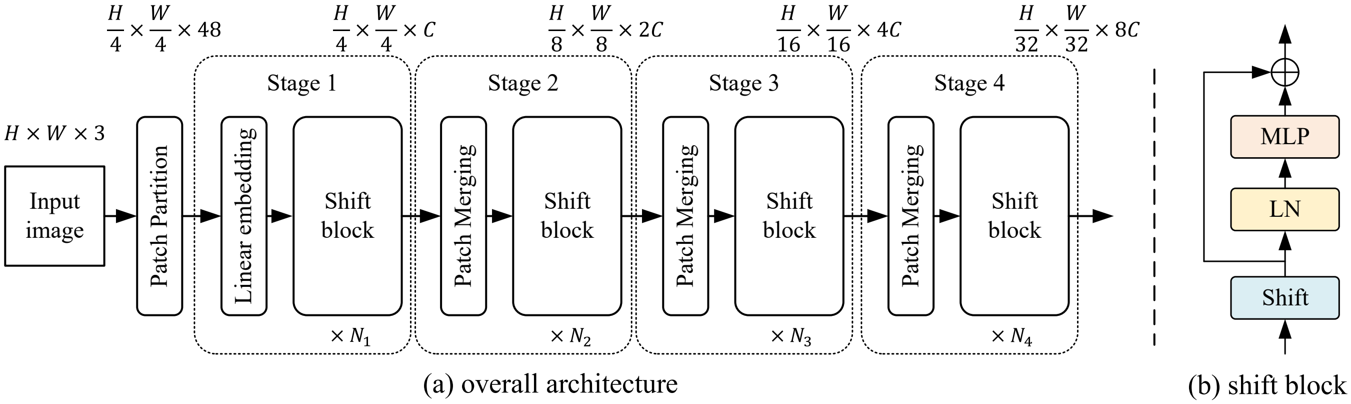
Inspired by the successful application in NLP, attention mechanism has also received increasing interests from the computer vision community. The early explorations can be roughly divided into two categories. On the one hand, some literature considers attention as a plug-and-play module, which can be seamlessly integrated into the existing CNN architectures. The representative work includes non-local network (Wang et al. 2018), relation network (Hu et al. 2018), and CCNet (Huang et al. 2019). On the other hand, some pieces of work aim to substitute all convolution operations with the attention mechanism, such as local relation network (Hu et al. 2019) and self-attention network (Zhao, Jia, and Koltun 2020).

Although these two kinds of work have shown promising results, they are still built on the CNN architecture. ViT (Dosovitskiy et al. 2020) is the pioneering work that leverages a pure transformer architecture for visual recognition tasks. Thanks to its impressive performance, the community recently bursts out a rising wave of research on vision transformers. Along this line of research, the main focus is to improve the attention mechanism, so that it can satisfy the intrinsic properties of visual signals. For example, MSViT (Fan et al. 2021) builds hierarchical attention layers to obtain multi-scale features. Swin Transformers (Liu et al. 2021b) introduces a locality constrain into its attention mechanism. The related efforts also include pyramid attention (Wang et al. 2021), local-global attention (Li et al. 2021), cross attention (Chen, Fan, and Panda 2021), to name a few.

Unlike the particular interests in attention mechanism, the remaining components of ViT are less studies. DeiT (Touvron et al. 2020) has setup a standard training pipeline for vision transformers. Most follow-up work inherits its setting, and only make some modifications on the attention mechanism. Our work also follows this paradigm. However, the goal of this work is not to complex the design of attention. On the contrary, we aim to show that the attention mechanism might not be the critical part of making ViTs work. It can be even replaced by an extremely simple shift operation. We hope these results can inspire researchers to rethink the role of attention mechanism.

MLP Variants

Our work is related to the recent multi-layer-perceptron (MLP) variants. Specifically, MLP variants propose to extract image features through a pure MLP-like architecture. They also jump out of the attention-based framework in



(a) overall architecture

(b) shift block

Figure 2: (a) The overall architecture of our ShiftViT. We follow Swin Transformer (Liu et al. 2021b) to build hierarchical representations. (b) The detail design of a shift block. We only use a simple shift operation to model spatial relationships.

ViT. For example, instead of using the self-attention matrix, MLP-Mixer (Tolstikhin et al. 2021) introduces a token-mixing MLP to directly connect all spatial locations. It eliminates the dynamic property of ViT, but without losing accuracy. The follow-up work investigates more MLP designs, like the spatial gating unit (Liu et al. 2021a) or cyclic connection (Chen et al. 2021).

Our ShiftViT can be also categorized into the pure MLP architecture, where the shift operation is viewed as a special token-mixing layer. Compared with the existing MLP work, our shift operation is even much simpler, since it contains no parameter and no FLOP. Moreover, the vanilla MLP variants fail to handle variable input size because of the fixed linear weights. Our shift operation overcomes this obstacle and therefore make the backbone feasible for more vision tasks like object detection and semantic segmentation.

Shift Operation

Shift operation is not new in computer vision. As early as in 2017, it was proposed to be an efficient alternative to the spatial convolution operation (Wu et al. 2018). Concretely, it uses a sandwich-like architecture, two 1×1 convolutions and a shift operation, to approximate a $K \times K$ convolution. In the follow-up work, the shift operation is further extended into different variants, such as active shift (Jeon and Kim 2018), sparse shift (Chen et al. 2019) and partial shift (Lin, Gan, and Han 2019).

In this work, we adopt the partial shift operation (Lin, Gan, and Han 2019). It is notable that the goal of this work is not to present a novel operation. Instead of that, we integrate the existing shift operation with the popular ViT to verify the effectiveness of attention mechanism. The similar vision are shared with the concurrent work ShiftMLP (Yu et al. 2021) and AS-MLP (Lian et al. 2021), but the design details are quite different. Their building blocks are more complex, which involve some auxiliary layers like pre-transformation and post-transformation.

Shift Operation Meets Vision Transformer

Architecture Overview

For a fair comparison, we follow the architecture of Swin Transformer (Liu et al. 2021b). The architecture overview is illustrated in Figure 2 (a). Specifically, given an input image of shape $H \times W \times 3$, it first splits the images into non-overlapping patches. The patch size is 4×4 pixels. Therefore, the output of patch partition is is $\frac{H}{4} \times \frac{W}{4}$ tokens, where each token has a channel size of 48.

The modules followed by can be divided into 4 stages. Each stage contains two parts: embedding generation and stacked shift blocks. For the embedding generation of the first stage, a linear projection layer is used to map each token into an embedding of channel size C. For the rest stages, we merge neighbouring patches through the convolution with a kernel size of 2×2. After patch merging, the spatial size of the output is half down-sampled, while channel size is twice the input, i.e., from C to $2C$.

The stacked shift block is built by some repeated basic units. The detail design of each shift block is shown in Figure 2 (b). It composes of a shift operation, a layer normalization and a MLP network. This design is almost the same as the standard transformer block. The only difference is that we use a shift operation rather than a attention layer. For each stage, the number of shift blocks can be various, which is denoted as N_1, N_2, N_3, N_4 respectively. In out implementation, we carefully choose the value of N_i so that the overall model share a similar number of parameters with the baseline Swin Transformer model.

Shift Block

The detail architecture of our shift block is depicted in Figure 2 (b). Specifically, this block consists of three sequentially-stacked components: shift operation, layer normalization and MLP network.

Shift operation has been well studied in CNNs. It can have many design choices, such as active shift (Jeon and Kim 2018) and sparse shift (Chen et al. 2019). In this work, we follow the partial shift operation in TSM (Lin, Gan, and Han 2019). The illustration is presented in Figure 1 (b). Given an input tensor, a small portion of channels will be shifted along 4 spatial directions, namely left, right, top, and down, while the remaining channels keep unchanged. After shifting, the out-of-scope pixels are simply dropped and the vacant pixels are zero padded. In this work, the shift step is set to 1 pixel.

Formally, we assume that the input feature $\mathbf{z}$ is of shape $H \times W \times C$, where C is the number of channels, H and W are spatial height and width, respectively. The output feature $\hat{\mathbf{z}}$ has the same shape as input. It can be written as:

$$\begin{aligned}
\hat{\mathbf{z}}[0:H, 1:W, 0:\gamma C] &\leftarrow \mathbf{z}[0:H, 0:W-1, 0:\gamma C] \\
\hat{\mathbf{z}}[0:H, 0:W-1, \gamma C:2\gamma C] &\leftarrow \mathbf{z}[0:H, 1:W, \gamma C:2\gamma C] \\
\hat{\mathbf{z}}[0:H-1, 0:W, 2\gamma C:3\gamma C] &\leftarrow \mathbf{z}[1:H, 0:W, 2\gamma C:3\gamma C] \\
\hat{\mathbf{z}}[1:H, 0:W, 3\gamma C:4\gamma C] &\leftarrow \mathbf{z}[0:H-1, 0:W, 3\gamma C:4\gamma C] \\
\hat{\mathbf{z}}[0:H, 0:W, 4\gamma C:C] &\leftarrow \mathbf{z}[0:H, 0:W, 4\gamma C:C]
\end{aligned}$$

where γ is a ratio factor to control how many percentages of channels will be shifted. In most experiments, the value of γ is set to $\frac{1}{12}$.

It is notable that shift operation does not hold any parameter or arithmetic calculation. The only implementation is memory copying. Therefore, shift operation is highly efficient and it is very easy to implement. The pseudo code is presented in Algorithms 1. Compared with the self-attention mechanism, shift operation is clean, neat, and more friendly to deep learning inference library like TensorRT.

The rest of the shift block is the same as the standard building block of ViT. The MLP network has two linear layers. The first one increases the channel of the input feature to a higher dimension, e.g., from C to τC. Then the second linear layer projects the high-dimensional feature into the original channel size of C. Between these two layers, we adopt GELU as the non-linear activation function.

Architecture Variants

For a fair comparison with the baseline Swin Transformer, we also build multiple models with various number of parameters and computational complexity. Specifically, we introduce Shift-T(iny), Shift-S(mall), Shift-B(ase) variants [1], which is corresponded to Swin-T, Swin-S and Swin-B, respectively. Shift-T is the smallest one, which shares a similar size with Swin-T and ResNet-50. Another two variants, Shift-S and Shift-B, are roughly 2× and 4× more complex than ShiftViT-T. The detail configurations of basic embedding channels C and number of blocks $\{N_i\}$ are presented as following:

- Shift-T: $C = 96$, $\{N_i\} = \{6, 8, 18, 6\}$, $\gamma = 1/12$
- Shift-S: $C = 96$, $\{N_i\} = \{10, 18, 36, 10\}$, $\gamma = 1/12$
- Shift-B: $C = 128$, $\{N_i\} = \{10, 18, 36, 10\}$, $\gamma = 1/16$

Beside the model size, we also have a closer look at the model depth. In our proposed model, nearly all parameters are concentrated in the MLP part. Therefore, we can control the expand ratio of MLP τ to obtain a deeper network depth. If not specified, the expand ratio τ is set to 2. We have an ablation analysis to show that the deeper model achieve a better performance.

Algorithm 1: Pytorch-like pseudo code of shift

```
def shift(feat, gamma=1/12):
    # feat is a tensor with a shape of
    # [Batch, Channel, Height, Width]
    B, C, H, W = feat.shape
    g = int(gamma * C)
    out = zeros_like(feat)
    # spatially shift
    out[:, 0*g:1*g, :, :-1] = x[:, 0*g:1*g, :, 1:]
    out[:, 1*g:2*g, :, 1:] = x[:, 1*g:2*g, :, :-1]
    out[:, 2*g:3*g, :-1, :] = x[:, 2*g:3*g, 1:, :]
    out[:, 3*g:4*g, 1:, :] = x[:, 3*g:4*g, :-1, :]
    # remaining channels
    out[:, 4*g:, :, :] = x[:, 4*g:, :, :]
    return out
```

Experiments

Implementation Details

We conduct experiments on three mainstream visual recognition benchmarks: image classification on ImageNet-1k dataset (Deng et al. 2009), object detection on COCO dataset (Lin et al. 2014) and semantic segmentation on ADE20k dataset (Zhou et al. 2019).

For image classification task, we exactly follow the protocol as in Swin Transformer (Liu et al. 2021b). An average pooling layer and a linear classification layer are appended after the backbone network. All the parameters are randomly initialized and trained for 300 epochs with an AdamW optimizer. The learning rate starts from 0.001 and gradually decay to 0 with a cosine schedule. We include all data augmentations and regularization tricks as in Swin Transformer (Liu et al. 2021b). The batch size is set to 1024.

For object detection task, there exists many off-the-shelf detection frameworks, such as Faster R-CNN, Mask R-CNN and RetinaNet. For a fair comparison with other methods, we follow the common practice of using Mask R-CNN and Cascade Mask R-CNN. In such detection frameworks, the backbone is our proposed Shift network, while the rest of components like FPN and detection head remain the same. We initialize the backbone with pretrained weights of the ImageNet-1k classifier. The training duration lasts for 12 epochs (denoted as 1× schedule) or 36 epochs (denoted as 3× schedule). The optimizer is AdamW, with an initial learning rate 0.0001. The batch size is 16. During training period, we utilize the multi-scale training trick, i.e., the shorter side of the input image is resized into a range from 480 pixels to 800 pixels. We report the mean average precision (mAP) metrics on the validation set of COCO dataset.

For semantic segmentation task, we evaluate our method on ADE20K dataset, which contains 20K images for training and 2K images for validation. In these experiments, the base segmentation framework is UperNet. The model is trained on the training set of ADE20K and the evaluation metric is the mean IoU (mIoU) score on the validation set. Similar to the setting of object detection, our Shift backbones are also pretrained on ImageNet-1k. The rest of settings are same as Swin-Transformer. The training batch size is 16 and we train the model for 160k iterations. For the comparison with the state-of-the-arts, we adopt the multi-scale testing strategy.

[1] For simplification, we ignore the suffix of "ViT" and use Shift-T to denote ShiftViT-T in this work.

Table 1: Comparison with the baseline Swin Transformer on three mainstream tasks: image classification, object detection and semantic segmentation. The suffix `/light` denotes the lightweight version of our ShiftViT, where we only replace attention layers with the shift operation and keep remaining parts unchanged. The throughput speed is evaluated on a single NVidia GTX1080-Ti GPU.The gree and gray colors indicate the gain and loss, respectively.

Model	Param (M)	ImageNet			COCO				ADE20k
		FLOPs (G)	Speed (FPS)	Top-1 Acc.(%)	Mask R-CNN 1×		Mask R-CNN 3×		UpperNet
					AP^b	AP^m	AP^b	AP^m	mIoU
ResNet-50	26	4.1	676	76.1	38.0	34.4	41.0	37.1	-
Swin-T	29	4.5	356	81.3	43.7	39.5	46.0	41.6	44.5
Shift-T/light	20	3.0	790	79.4	41.3	38.0	43.2	39.2	42.6
Shift-T	29	4.5	396	81.7 (+0.4)	45.4 (+1.7)	40.9 (+1.4)	47.1 (+1.1)	42.3 (+0.7)	46.3 (+1.8)
Swin-S	50	8.7	217	83.0	46.4	41.7	48.5	43.3	47.6
Shift-S/light	34	5.7	457	81.6	44.8	40.4	46.0	41.1	45.4
Shift-S	50	8.8	215	82.8 (-0.2)	47.2 (+0.8)	42.2 (+0.5)	48.6 (+0.1)	43.4 (+0.1)	47.8 (+0.2)
Swin-B	88	15.4	158	83.5	46.9	42.1	48.7	43.4	48.1
Shift-B/light	60	10.2	312	82.3	45.7	41.0	46.0	41.2	45.8
Shift-B	89	15.6	154	83.3 (-0.2)	47.7 (+0.8)	42.7 (+0.6)	48.0 (-0.7)	42.8 (-0.6)	47.9 (-0.2)

Table 2: Comparison with state-of-the-art methods on the ImageNet-1k classification task.

Model	Input resolution	# Params (M)	FLOPs (B)	Top-1 Acc. (%)
CNN-based				
RegNetY-4G	224^2	21	4.0	80.0
RegNetY-8G	224^2	39	8.0	81.7
RegNetY-16G	224^2	84	16.0	82.9
EfficientNet-B4	380^2	19	4.2	82.9
EfficientNet-B5	456^2	30	9.9	83.6
EfficientNet-B6	528^2	43	19.0	84.0
ViT-based and MLP-based				
DeiT-S	224^2	22	4.6	79.8
DeiT-B	224^2	86	17.5	81.8
PVT-S	224^2	25	3.8	79.8
PVT-L	224^2	61	9.8	81.7
Swin-T	224^2	29	4.5	81.3
Swin-S	224^2	50	8.7	83.0
Swin-B	224^2	88	15.4	83.5
MLP-Mixer-B/16	224^2	79	-	76.4
gMLP-S	224^2	20	4.5	79.4
gMLP-B	224^2	73	15.8	81.6
S^2-MLP-D	224^2	71	14.0	80.0
S^2-MLP-W	224^2	51	10.5	80.7
AS-MLP-T	224^2	28	4.4	81.3
AS-MLP-S	224^2	50	8.5	83.1
AS-MLP-B	224^2	88	15.2	83.3
Ours				
Shift-T	224^2	28	4.4	81.7
Sfhit-S	224^2	50	8.5	82.8
Sfhit-B	224^2	88	15.2	83.3

Comparison with Baseline

The goal of this work is to demystify the role of attention mechanism and explore whether it can be replaced by an extremely simple shift operation. Concretely, our proposed backbones are based on the architecture of Swin Transformer, which is one of the most representative ViT variants. We therefore consider Swin Transformer as the baseline model, and compare our ShiftViT to it.

For an apple-to-apple comparison, we first build a lightweight version of ShiftViT. It is nearly the same as the Swin Transformer counterpart, except that the attention layers are substituted by the shift operations. We denote this backbone with a suffix `/light`, because replacing attention with shift will lead to a reduction in parameters and FLOPs. The experimental results are presented in Table 1. We exhaustively compare all variants in three different sizes. The results show that the shift operation is weaker than the attention mechanism, because it does not contain any learnable parameter or arithmetic calculation. For example, the Shift-T/light model has only 20M parameters and 3.0 FLOPs, which are nearly 33% less than the Swin-T model. Therefore, there is no wonder that its performance is marginally worse than the baseline. Despite the relative gap to the baseline, it is worth noting that the absolute accuracy of the lightweight ShiftViT is not bad. Compared with the typical ResNet-50 backbone, Shift-T/light is more powerful and more efficient.

To remedy the complexity gap between shift operation and attention mechanism, we can adopt more building blocks in ShiftViT to make sure it has a similar number of parameters with the Swin baseline. In such fair comparisons, our models achieve even better results than Swin-Transformer. For the small-size models, our Shift-T backbone attains an mAP score of 45.4% on COCO and an mIoU score of 46.3% on ADE20k, which outperform the Swin-T backbone by a remarkable margin. For the large-size models, ShiftViT seems to be saturated. But the performance is still on par with the Swin baseline.

Although the shift operation is weaker than the attention mechanism in spatial modelling, its simple architecture allows the network to grow deeper. As such, the weakness of the shift operation is greatly alleviated. Within the same computational budget, the overall performance of ShiftViT is comparable to the attention-based Swin Transformer. These experiments prove that the attention mechanism might not be necessary for ViTs. Even an extremely simple operation can achieve the similar results.

Comparison with State-of-the-Art

To further demonstrate the effectiveness, we compare ShiftViT backbones with existing state-of-the-art methods. For image classification task on ImageNet-1k, our proposed models are compared to three different types of models, namely CNN, ViT and MLP. The results are detailed in Table 2. Overall, our method can achieve a comparable performance with the state-of-the-arts. For ViT-based and MLP-based methods, the best performances are around 83.5% top-1 accuracy, while our model achieves an accuracy of 83.3%. For CNN-based methods, our model is slightly worse than EfficientNet series, but the comparison is not fully fair because EfficientNet takes a larger input size.

Another interesting thing is the comparison with two concurrent work S^2-MLP (Yu et al. 2021) and AS-MLP (Lian et al. 2021). These two pieces of work share the similar idea on shift operation , but they introduce some auxiliary modules into the building block, e.g., the pre- and post-projection layers. In Table 2, our performances are slightly better than these two work. It justifies our design choice that building backbone solely with a simple shift operation is good enough.

Beside the classification task, the similar performance tread can be also observed in the object detection task and semantic segmentation task. It is notable that some ViT-based and MLP-based methods cannot be easily extended to such dense prediction tasks, because the high-resolution inputs yield unaffordable computational burdens. Our method does not suffer from this obstacle thanks to the high efficiency of shift operation. As shown in Table 3 and Table 4, the advantages of our ShiftViT backbones are clear. Shift-T attains an mAP score of 47.1 on object detection and an mIoU score of 47.8 on semantic segmentation, which outperform other methods by a considerable margin.

Ablation Analysis

In this section, we aim to explore what factors contribute to the good performance of ShiftViT. We first analyze the impact of two hyper-parameters in ShiftViT. Then, we dive into the training scheme of ViT series.

Expand ratio of MLP The previous experiments have justified our design principle, i.e., a great model depth can remedy the weakness of each building block. Generally, there exists a trade-off between the model depth and the complexity of building blocks. With a fixed computational

Table 3: Comparison with state-of-the-art methods on the COCO object detection task. Following the common practice, we couple the backbones with two detection frameworks, namely Mask R-CNN and Cascade Mask R-CNN.

Backbone	Params (M)	FLOPs (G)	AP^b	AP^m
Mask R-CNN 3×				
Res-50	44	260	41.0	37.1
PVT-S	44	245	43.0	39.9
AS-MLP-T	48	260	46.0	41.5
Swin-T	48	264	46.0	41.6
Shift-T	48	265	47.1	42.3
Res-101	63	336	42.8	38.5
PVT-M	64	302	44.2	40.5
AS-MLP-S	69	346	47.8	42.9
Swin-S	69	354	48.5	43.3
Shift-S	70	350	48.6	43.4
Cascade Mask R-CNN 3×				
Res-50	82	739	46.3	40.1
AS-MLP-T	86	745	50.1	43.5
Swin-T	86	739	50.4	43.7
Shift-T	86	743	50.3	43.4
ResX-101	101	819	48.1	41.6
AS-MLP-S	107	824	51.1	44.2
Swin-S	107	838	51.8	44.7
Shift-S	107	827	50.9	44.0

Table 4: Comparison with state-of-the-art methods on the ADE20k semantic segmentation task. We report the mIoU metrics on the validation set.

Method	Backbone	Params (M)	FLOPs (G)	val mIoU
DANet	ResNet-101	69	1119	45.2
DNL	ResNet-101	69	1249	46.0
DeepLabV3	ResNet-101	63	1021	44.1
OCRNet	ResNet-101	89	1381	44.9
DeepLabV3	ResNeSt-101	66	1051	46.9
DeepLabV3	ResNeSt-200	88	1381	48.4
OCRNet	HRNet-w64	71	664	45.7
UperNet	ResNet-101	89	1029	44.9
UperNet	Swin-T	60	945	45.8
UperNet	AS-MLP-T	60	937	46.5
UperNet	Shift-T	60	942	47.8
UperNet	Swin-S	81	1038	49.5
UperNet	AS-MLP-S	81	1024	49.2
UperNet	Shift-S	81	1029	49.6
UperNet	Swin-B	121	1188	49.7
UperNet	AS-MLP-B	121	1166	49.5
UperNet	Shift-B	121	1174	49.2

Table 5: Ablation analysis on the expand ratio of MLP. The first row shows the Swin-T baseline. The row with blue background denotes the default setting in our experiments. All entries share the same number of parameters and FLOPs.

Expand	Depth	ImgNet	COCO		ADE20k
Ratio		Acc. (%)	AP^b	AP^m	mIoU
Swin	48	81.3	43.7	39.5	44.5
4	57	81.3	44.0	39.8	44.4
3	75	81.5	44.4	40.2	45.5
2	114	81.7	45.4	40.9	46.3
1	225	81.8	45.2	40.6	47.3

budget, a lightweight building block can enjoy a deeper network architecture.

To further investigate this trade-off, we present some ShiftViT models with different depths. For ShiftViT, most parameters exist in the MLP part. We can change the expand ratio of MLP τ to control the model depth. As shown in Table 5, we choose Shift-T as our baseline model. We explore the expand ratio τ within a range from 1 to 4. It is worth noting that the parameters and FLOPs for different entries are almost the same. From Table 5, we can observe a trend that a deeper model results in a better performance. When the depth of ShiftViT increases to 225, it outperforms the 57-layer counterpart by 0.5%, 1.2% and 2.9% absolute gains on classification, detection and segmentation, respectively. This trend supports our conjecture that a powerful-and-heavy module, like attention, may not be the optimal choice for backbone. We hope it can help the future work to rethink such trade-off when designing backbones.

Percentage of shifted channels The shift operation has only one hyper-parameter, namely the percentage of shifted channels. By default, it is set to 33%. In this section, we explore some other settings. Specifically, we set the percentage of shifted channels to 20%, 25%, 33% and 50%, respectively. The results are presented in Figure 3. It shows that the final performance is not very sensitive to this hyper-parameter. Shifting 25% of channels only results in 0.3% absolute loss compared to the best setting. Within the reasonable range (from 25% to 50%), all the settings achieve a better accuracy than the Swin-T baseline.

ViT-style training scheme Shift operation has been well studied in CNNs. However, the previous work does not show the impressive performance as ours. Shift-ResNet-50 (Wu et al. 2018) only achieve an accuracy of 75.6% on ImageNet, which is far behind our 81.7% accuracy. This gap raise a natural concern about what makes good for our ShiftViT.

We suspect the reason might lie in the ViT-style training scheme. Specifically, most existing ViT variants follow the setting as in DeiT (Touvron et al. 2020), which is quite different from the standard pipeline of training CNNs. For example, ViT-style scheme adopts AdamW optimizer and the training duration lasts for 300 epochs on ImageNet. As a comparison, CNN-style scheme prefers SGD optimizer and the training schedule is usually 90 epochs only. Since our model inherit the ViT-style training scheme, it is interesting to see how such differences affect the performance.

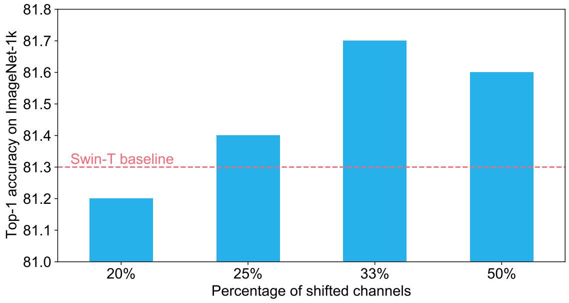


Figure 3: Ablation analysis on the percentage of shifted channels. We plot the top-1 classification accuracy on ImageNet-1k. The red line indicates Swin-T baseline.

Table 6: Ablation analysis on the typical configurations of CNNs and ViTs. We gradually transfer the training configuration from the CNN's setting to the ViT's setting, and investigate how these factors influence the model performances.

SGD ↓ AdamW	ReLU ↓ GELU	BN ↓ LN	90ep ↓ 300ep	ImageNet Top-1 Acc. (%)
				76.4
✓				77.9
✓	✓			78.5
✓	✓	✓		78.4
✓	✓	✓	✓	81.7

Due to the resource limitation, we cannot fully align all settings between ViT-style and CNN-style. Therefore, we pick four important factors that we believe can bring some insights, i.e. optimizer, activation function, normalization layer and training schedule. From Table 6, we can observe that such factors can significantly influence the accuracy, especially the training schedule. These results shows that the good performance of ShiftViT is partly brought by the ViT-style training scheme. Similarly, the success of ViT may be also related to its special training scheme. We should take it seriously in the future study of ViTs.

Conclusion

In this work, we move a small step toward demystifying the essential reason why ViT works. The experiments show that the attention mechanism might not be the vital factor for the success of ViT. We can even use an extremely simple shift operation to replace the attention layer. The proposed backbone, namely ShiftViT, can work as well as the Swin Transformer baseline. Since the shift operation is already the simplest spatial modelling module, we argue that the good performance must come from the remaining components of ViT, e.g., the FFN and the training scheme. In future work, we plan to have more analysis on such factors and investigate more ViT variants.

References

Chen, C.-F.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*.

Chen, S.; Xie, E.; Ge, C.; Liang, D.; and Luo, P. 2021. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*.

Chen, W.; Xie, D.; Zhang, Y.; and Pu, S. 2019. All you need is a few shifts: Designing efficient convolutional neural networks for image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7241–7250.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3588–3597.

Hu, H.; Zhang, Z.; Xie, Z.; and Lin, S. 2019. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3464–3473.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 603–612.

Jeon, Y.; and Kim, J. 2018. Constructing Fast Network through Deconstruction of Convolution. *Advances in Neural Information Processing Systems*, 31: 5951–5961.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *NIPS*, 25: 1097–1105.

Li, J.; Yan, Y.; Liao, S.; Yang, X.; and Shao, L. 2021. Local-to-Global Self-Attention in Vision Transformers. *arXiv preprint arXiv:2107.04735*.

Lian, D.; Yu, Z.; Sun, X.; and Gao, S. 2021. AS-MLP: An Axial Shifted MLP Architecture for Vision. *arXiv preprint arXiv:2107.08391*.

Lin, J.; Gan, C.; and Han, S. 2019. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7083–7093.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, H.; Dai, Z.; So, D. R.; and Le, Q. V. 2021a. Pay Attention to MLPs. *arXiv preprint arXiv:2105.08050*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv*.

Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Keysers, D.; Uszkoreit, J.; Lucic, M.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2020. Training data-efficient image transformers and distillation through attention. *arXiv preprint arXiv:2012.12877*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Wu, B.; Wan, A.; Yue, X.; Jin, P.; Zhao, S.; Golmant, N.; Gholaminejad, A.; Gonzalez, J.; and Keutzer, K. 2018. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9127–9135.

Yu, T.; Li, X.; Cai, Y.; Sun, M.; and Li, P. 2021. S ^{2}-MLP: Spatial-Shift MLP Architecture for Vision. *arXiv preprint arXiv:2106.07477*.

Zhao, H.; Jia, J.; and Koltun, V. 2020. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10076–10085.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3): 302–321.