ShiftFormer: Spatial-Temporal Shift Operation in Video Transformer

Beiying Yang[1,2], Guibo Zhu[1,2,3(✉)], Guojing Ge[2], Jinzhao Luo[1,2], Jinqiao Wang[1,2,3,4]
[1]*School of Artificial Intelligence, University of Chinese Academy of Sciences*
[2]*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences*
[3]*Wuhan AI Research*, [4]*Peng Cheng Laboratory*
{beiying.yang, gbzhu, guojing.ge, jqwang}@nlpr.ia.ac.cn, luojinzhao2020@ia.ac.cn

***Abstract*—Transformers have achieved great success in various tasks, especially that introducing pure Transformers into video understanding shows powerful performance. However, video Transformer suffers from the problem of memory explosion: it is difficult to be deployed on hardware due to the intensive computation. To address this issue, we propose ST-shift (spatial-temporal) operation with zero computation and zero parameter. We are only shifting a small portion of the channels along the temporal and spatial dimensions. Based on this operation, we build an attention-free ShiftFormer, where ST-shift blocks substitute the attention layers in video Transformer. ShiftFormer is accurate and efficient: it can reduce 56.34% of memory usage and achieve 3.41× faster training. When both using random initialization, our model performs even better than Video Swin Transformer for video recognition on Something-Something v2.**

***Index Terms*—Video Classification; Transformer; Shift Operation**

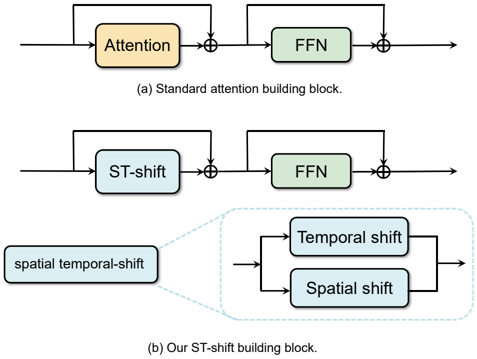


Fig. 1: An illustration of our ST-shift building block. We replace the attention layer with a spatial-temporal shift operation in video Transformer.

I. Introduction

Convolution-based backbone architectures have long dominated in computer vision. As CNNs became the backbone network for various vision tasks, these architectural advances led to performance improvements that it extensively changed and improved the entire domain. However, the current network backbone for image classification is shifting from CNNs to Transformers. This trend is originated with the proposal of the visual Transformer (ViT) [6]. The ViT and subsequent works [8], [14], [21] have achieved more and more competitive performance than CNNs. The great success of ViT on image classification is quickly expanded to many other computer vision tasks such as object detection, semantic segmentation and video recognition tasks.

Undeniably, the self-attention mechanism has enabled Transformer to significantly improve its ability to capture long-term dependencies [1], [2]. However, at the same time, this mechanism also needs to perform additional attention operations with temporal and spatial complexity. In particular, the enormous computational consumption brought to the field of video understanding is unbearable.

In addition, there is still no convincing research showing what makes ViT perform well in vision tasks. The attention mechanism can computes global similarity by modeling long-range dependencies, while convolution can only locally aggregates contexts in small neighborhoods, making it difficult to model global dependencies with restricted sensory fields. Due to these good properties, some work suggests that the attention mechanism facilitates the powerful expressive performance of ViTs.

However, ViT variants [18], [20] can still achieve competitive performance without attention operation in the image classification task. The paper [17] have verified that excellent results is achieved by building models on MLPs only, skipping connections, and not using self-attention layers. In another research direction, several ViT variants introduce local attention mechanisms. For example, Uniformer [11] integrates the advantages of CNN and Transformer models. Swin Transformer [14] restricts attention to a small local region while experimental results show that the local restriction does not degrade the network's performance.

Since the video Transformer generally requires more computation and more parameters for training. To build a memory-friendly model of efficient video understanding backbone and explore the necessity of attention mechanism in video understanding. We further reduce the attention layer to a straightforward case: an attention-free Transformer with zero computation and zero parameter. As shown in Fig. 1, only the attention layer is replaced by an effortless operation that improves the efficiency of the video model.

This work was supported by National Key R&D Program of China under Grant No. 2021YFE0205700, National Natural Science Foundation of China (No.62076235,61976210, 62002356, 6200623062002357), sponsored by Zhejiang Lab (No.2021KH0AB07).

In this work, we revisit the shift operation and extend it to 3D space. The standard Transformer block consists of an attention layer and a feedforward network. We remove the attention layer and replace it with the ST-shift block. To facilitate information exchanged among neighboring frames and spatial features, the operation is as follows: given an input feature, the proposed building block will shift a small portion of the channels along four spatial directions (left, right, top, and down) and two temporal directions (forward, backward). We follow the 3D window shift mechanism of the Video Swin Transformer while keeping the non-overlapping window-based shift calculation validly, which can increase the perceptual wildness of the spatial and temporal dimensions of the video. Based on this ST-shift block, we have built a Video Transformer-like backbone network: ShiftFormer.

For a fair comparison, we all use random initialization. Because it is not easy to pre-train with a large magnitude of datasets, it is expensive on storage hardware, computational resources, and experimental cycles. It performs even better than the powerful Video Swin Transformer for video recognition tasks on Something-Something v2 when both using random initialization. The proposed model can reduce 56.34% of memory usage and achieve 3.41× faster training compared with Video Swin Transformer. ShiftFormer is trained on small server GPUs while increasing the batch size, unlike the previous video Transformer, which only use large server GPUs (NVIDIA Tesla V100 and A100). Moreover, the training time is reduced by a factor of three, significantly saving training and inference time. Realizing efficient action recognition models with limited resources and data in different scenes, we have improved the capability and efficiency of video understanding. The contributions of this work can be concluded as follows:

- We proposed an attention-free video Transformer, which effectively reduces memory usage and training time.
- The proposed model achieves better performance than Video Swin Transformer when both utilizing random initialization.
- Experiments show that attention may not be necessary for the superiority of video Transformer in video understanding tasks.

II. Related Work

A. Attention Vision Transformers

As research advances, Transformer has progressed in visual tasks such as image classification [10], [19], object detection [16], and semantic segmentation [5]. It has also led to research on Transformer-based architectures for video recognition tasks. However, the current Transformer structure based on video tasks suffers from the problem of memory explosion. Most of the subsequent work of Transformer [7] only made some modifications to the attention mechanism. We aim to show whether attention mechanisms are necessary for video comprehension tasks. We are conducting memory-friendly model for video comprehension tasks.

B. MLP Variants

Recent work has tried to replace the self-attention layer in the Transformer with a linear layer to build models that contain only multilayer perceptrons(MLPs). For example, instead of using the self-attentive mechanism, MLP-Mixer [4] uses token-mixing MLP and channel-mixing MLP to capture the relationship between tokens and channels, respectively. CycleMLP [17] introduces a new tool called cycle fully connected layer, which replaces the token-mixing MLP in MLP-Mixer in token-mixing MLP. Our approach can also be attributed to the network of pure MLPs. Unlike previous methods, our approach can be used to process video data and is more straightforward.

Our model can handle with any input size. It can be used as a backbone network for video tasks.

C. Shift Operation

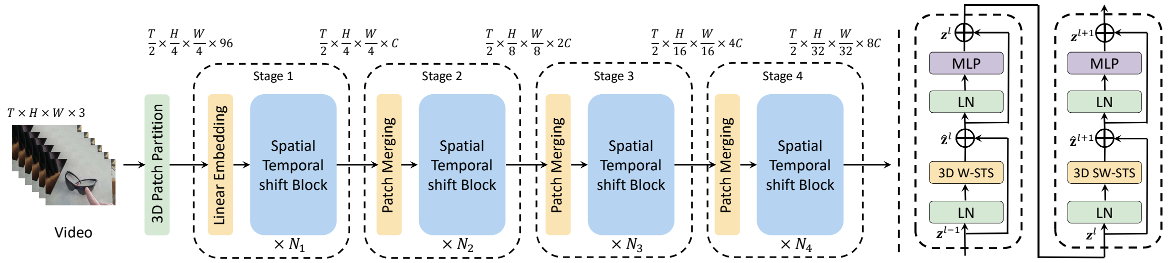
Shift operations are not new to computer vision. AS-MLP [12] shifted tokens along vertical and horizontal directions to get an axial receptive field. S2-MLP [22] also used the shift operation to achieve cross-patch communications. We believe that the capabilities of the MLP architecture have not been fully exploited. Our work is inspired by the partial shift operation in TSM [13] and ShiftViT [20]. To explore the field of video understanding for self-attention, we replaced it with a purely MLP-like architecture. The design details are different from previous works, which are more complex in building blocks and involve some auxiliary layers.

III. Methods

A. Overall Architecture

The overall architecture of the proposed ShiftFormer is shown in Fig. 2. For a fair comparison, we use the structure of the Video Swin Transformer. The input video is denoted as $T \times H \times W \times 3$ (T is the number of video frames). The video is first partitioned into non-overlapping 3D tokens with dimensions $2 \times 4 \times 4 \times 3$, and every two frames build a patch in the time dimension. Thus, the 3D patch partitioning layer's output size is $T/2 \times H/4 \times W/4$, and the feature dimension of each token is 96.

We simply replace the attention part of the Video Swin Transformer with our ST-shift block, following the original hierarchical structure. The model consists of four stages. In the first stage, a linear embedding layer is applied to this raw-valued feature to project it to an arbitrary dimension (denoted as C). In stages 2-4, the patch merging layer connects each set of 2×2 spatially adjacent patch features and performs 2-fold spatial downsampling (without downsampling in the time dimension). The output dimension is set to 2C. The main component of the model is the ST-shift block, replacing the attention mechanism of a standard video Transformer based on the spatiotemporal shift operation and keeping the other components unchanged. The number of ST-shift blocks used in each stage varies. Specifically, a ShiftFormer block consists of a 3D shifted window based ST-shift block followed by a feed-forward network. It contains two consecutive 2-layer MLP,


$\frac{T}{2}\times\frac{H}{4}\times\frac{W}{4}\times 96$
$\frac{T}{2}\times\frac{H}{4}\times\frac{W}{4}\times C$
$\frac{T}{2}\times\frac{H}{8}\times\frac{W}{8}\times 2C$
$\frac{T}{2}\times\frac{H}{16}\times\frac{W}{16}\times 4C$
$\frac{T}{2}\times\frac{H}{32}\times\frac{W}{32}\times 8C$
$T\times H\times W\times 3$
Video
3D Patch Partition
Stage 1
Linear Embedding
Spatial Temporal shift Block
$\times N_1$
Stage 2
Patch Merging
Spatial Temporal shift Block
$\times N_2$
Stage 3
Patch Merging
Spatial Temporal shift Block
$\times N_3$
Stage 4
Patch Merging
Spatial Temporal shift Block
$\times N_4$
z^l
MLP
LN
$\hat{z}^l$
3D W-STS
LN
z^{l-1}
z^{l+1}
MLP
LN
$\hat{z}^{l+1}$
3D SW-STS
LN
z^l

(a) Architecture

(b) Two Successive ST-shift Blocks

Fig. 2: Overall architecture of our proposed ShiftFormer. (a) ShiftFormer simply replaces the attention part of the Video Swin Transformer with our ST-shift block, following the original hierarchical structure. The model consists of four stages: In the first stage, a linear embedding layer is applied to this raw-valued feature to project it to an arbitrary dimension (denoted as C). In stages 2-4, the patch merging layer connects each set of 2 × 2 spatially adjacent patch features and performs 2-fold spatial downsampling (without downsampling in the time dimension). The output dimension is set to 2C. (b) ST-shift block is built by replacing the standard Transformer's multi-headed attention module based on the 3D-shift operation and keeping the other components unchanged.

Algorithm 1 pseudo code of ST-shift block (pytorch-like)

```
def ST-shift($Z$, n_div):
    # $Z$ is the input feature
    # 1 / n_div is the percentage of shifted channels in a single direction
    $B, T, H, W, C = Z.shape$
    $g = \text{int}(C/n\_\text{div})$
    $Z' = \text{zeros\_like}(Z)$
    # spatial shift
    $Z'[:, :, :, : -2, g*0 : g*1] = Z[:, :, :, 2 :, g*0 : g*1]$
    $Z'[:, :, :, 2 :, g*1 : g*2] = Z[:, :, :, : -2, g*1 : g*2]$
    $Z'[:, :, : -2, :, g*2 : g*3] = Z[:, :, 2 :, :, g*2 : g*3]$
    $Z'[:, :, 2 :, :, g*3 : g*4] = Z[:, :, : -2, :, g*3 : g*4]$
    # temporal shift
    $Z'[:, : -2, :, :, g*4 : g*5] = Z[:, 2 :, :, :, g*4 : g*5]$
    $Z'[:, 2 :, :, :, g*5 : g*6] = Z[:, : -2, :, :, g*5 : g*6]$
    # remaining channels
    $Z'[:, :, :, :, g*6 :] = Z[:, :, :, :, g*6 :]$
    return $Z'$
```

with a GELU non-linearity. The layer normalization operation is applied before the feed-forward network, and a residual connection is applied after each module. For each stage, the number of shift blocks can be various, which are denoted as $N1$, $N2$, $N3$, and $N4$, respectively. In our implementation, we choose the value of Ni so that the overall model shares a similar number of blocks with the baseline Video Swin Transformer (Swin-B) model.

B. Shift Block

The detailed architecture of our ST-shift block is depicted in Fig. 3. The module consists of three components: ST-shift block, layer normalization and MLP network. The shift operation has been widely used in computer vision. In the work of this paper, we improve on the work of the partial shift operation of TSM. Specifically, some channels of the input tensor shift along four spatial directions (up, down, left, and right) and shift the channels along the temporal dimensions (forward and backward), while the remaining channels remain unchanged. We perform partial shift operations in the temporal dimension, which allows contextual interactions to expand the temporal perception and achieve effective time fusion. After the shift operation, the redundant pixel features are truncated and the vacant pixel parts are zero-filled.

The input video feature Z is defined as $T \times H \times W \times C$, where T is the number of frames of the video, C is the number of channels, and H and W are the spatial height and width of the frames. The formula of the operation is given below, where the output feature Z' is the same size as input feature. is a ratio factor indicating the percentage of channels that will be shifted. In this case, γ is set to 1/24. The pseudo-code is presented in Algorithms 1. It is notable that TS-shift without incurring any extra computations and parameters.

$$
\begin{aligned}
&Z'[0:\mathrm{H}, 1:\mathrm{W}, 0:\mathrm{T}, 0:\gamma\mathrm{C}] \leftarrow \mathrm{Z}[0:\mathrm{H}, 0:\mathrm{W}-1, 0:\mathrm{T}, \mathrm{C}:\gamma\mathrm{C}]\\
&Z'[0:\mathrm{H}, 0:\mathrm{W}-1, 0:\mathrm{T}, \gamma\mathrm{C}:2\gamma\mathrm{C}] \leftarrow \mathrm{Z}[0:\mathrm{H}, 1:\mathrm{W}, 0:\mathrm{T}, \gamma\mathrm{C}:2\gamma\mathrm{C}]\\
&Z'[1:\mathrm{H}, 0:\mathrm{W}, 0:\mathrm{T}, 2\gamma\mathrm{C}:3\gamma\mathrm{C}] \leftarrow \mathrm{Z}[0:\mathrm{H}-1, 0:\mathrm{W}, 0:\mathrm{T}, 2\gamma\mathrm{C}:3\gamma\mathrm{C}]\\
&Z'[0:\mathrm{H}-1, 0:\mathrm{W}, 0:\mathrm{T}, 3\gamma\mathrm{C}:4\gamma\mathrm{C}] \leftarrow \mathrm{Z}[1:\mathrm{H}, 0:\mathrm{W}, 0:\mathrm{T}, 3\gamma\mathrm{C}:4\gamma\mathrm{C}]\\
&Z'[0:\mathrm{H}, 0:\mathrm{W}, 1:\mathrm{T}, 4\gamma\mathrm{C}:5\gamma\mathrm{C}] \leftarrow \mathrm{Z}[0:\mathrm{H}, 0:\mathrm{W}, 0:\mathrm{T}-1, 4\gamma\mathrm{C}:5\gamma\mathrm{C}]\\
&Z'[0:\mathrm{H}, 0:\mathrm{W}, 0:\mathrm{T}-1, 5\gamma\mathrm{C}:\mathrm{C}] \leftarrow \mathrm{Z}[0:\mathrm{H}, 0:\mathrm{W}, 1:\mathrm{T}, 5\gamma\mathrm{C}:\mathrm{C}]
\end{aligned}
$$

In this work, our ST-shift block directly moves the tensor to replace attention without any learned parameters. It only needs to implement memory replication, which is very efficient and greatly reduces the memory footprint, while essentially shortening the training time and enabling efficient video understanding. A ShiftFormer block consists of a 3D shifted window based ST-shift block followed by a feed-forward network, which contains two consecutive 2-layer MLP, with a GELU non-linearity. A layer normalization operation is applied before the feed-forward network, and a residual connection is used after each module.

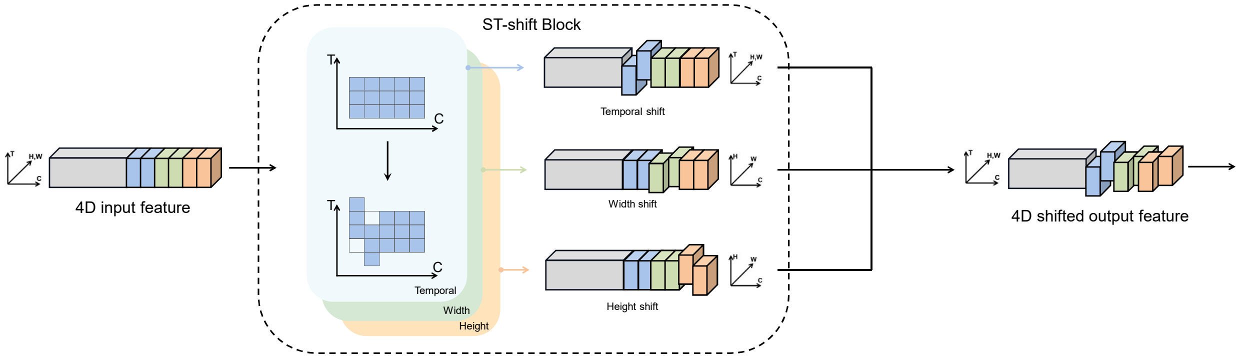


Fig. 3: Our proposed ST-shift block. Specifically, some channels of the input tensor shift along four spatial directions (up, down, left, and right) and two temporal directions (forward and backward), while the remaining channels remain unchanged. Those partial shift operations in three dimensions allow contextual interactions to expand the spatial-temporal perception and achieve effective spatial-temporal fusion.

C. 3D Shifted Window based ST-shift

ST-shift on non-overlapping 3D windows The multiple self-attention (W-MSA) mechanism on each non-overlapping 3D window is both practical and efficient for video recognition. To save computational and memory costs, we follow this idea. Unlike the attention operation in the Video Swin Transformer block, we use the ST-shift block directly. We perform the ST-shift on each non-overlapping 3D window.

3D Shifted Windows Since the ST-shift is performed on each non-overlapping 3D window, the lack of connectivity between different windows may limit the representability of the model. Therefore, to increase the introduction of cross-window connectivity, we follow the 3D window shift mechanism of the Video Swin Transformer while keeping the non-overlapping window-based shift calculation valid, which can increase the perceptual wildness of the spatial and temporal dimensions of the video.

IV. Experiment

A. Setup

Datasets For temporal modeling of human action recognition, we utilized the Something Something v2 (Sthv2) and Sthv2-top50 datasets. The Sthv2 dataset requires robust temporal modeling because most activities cannot be inferred based on spatial features alone (e.g., opening something, covering something with something). The dataset includes 168.9K training videos and 24.7K validation videos from 174 categories. To evaluate model performance in illegible challenging categories, we constructed the Sthv2-top50. The dataset is our ShiftFormer in the experimental results on the Sthv2 dataset, ranked by mean class accuracy, and the 50 classes with the lowest accuracy are selected to construct the Sthv2-top50 dataset. The dataset includes 38.3K training videos and 4.8K validation videos. For all methods, we followed the existing techniques and reported the accuracy for the top-1 and top-5.

Implementation Details For Sthv2 and Sthv2-top50, we employ AdamW optimizer to train 100 epochs with 2.5 epochs of linear warm-up. A batch size of 32 is used. The backbone and the head are randomly initialized without any pre-trained parameters. Although it is appealing to pretrain the whole model end-to-end with large-scale image datasets such as ImageNet dataset and video datasets such as Kinetics-400 [9] and Kinetics-600 [3], we are restricted by the enormous computation cost. Expressly, the initial learning rate of the spine and random initialization of the head are set to 3e-5 and 3e-4, respectively. We draw a 32-frame patch from each complete video, using a temporal stride of 2. The window size of the temporal dimension is 16. The patch size is 4×4 pixels. We result in input 3D tokens of $16 \times 56 \times 56$. We use a weight decay of 0.05. We employ augmentation, including label smoothing, random erasing, and RandAugment. We also employ stochastic depth with a ratio of 0.4. During inference, we follow by using 1×3 views, and for each clip, the shorter spatial side is scaled to 224 pixels, and we take three crops of size 224×224 that cover the longer spatial axis. The final score is computed as the average score of 1×3 views.

B. Comparison with other methods

Tab. I compares our approach with other strong video Transformer methods on Sthv2 and Sthv2-top50, because the datasets focus on temporal modeling. The Sthv2 accuracy reported in the Video Swin Transformer paper is a costly training process on a series of datasets: ImageNet (pretrain) → Kinetics-400 (pretrain) → Sthv2 (fine-tune). However, this paper does not focus on this step due to the restriction of enormous computation costs. For a fair comparison, all methods use random initialization. The results of Sthv2-top50 are shown on the right side of Tab. I. The previous best method achieves competitive results, where computation and parameters are too large for deployment. While our model is much more efficient: ShiftFormer achieves 50.25% Top-1 accuracy with 163.73 GFLOPs, which is 17.81% higher than Swin-B with 1.96× less computation and 1.5× fewer parameters. Therefore Our ShiftFormer can be trained on small server GPUs while increasing the batch size.

C. Visualization of activation graphs

We visualized the class activation map using GradCAM [15], and the results are shown in Fig.4. The baseline method

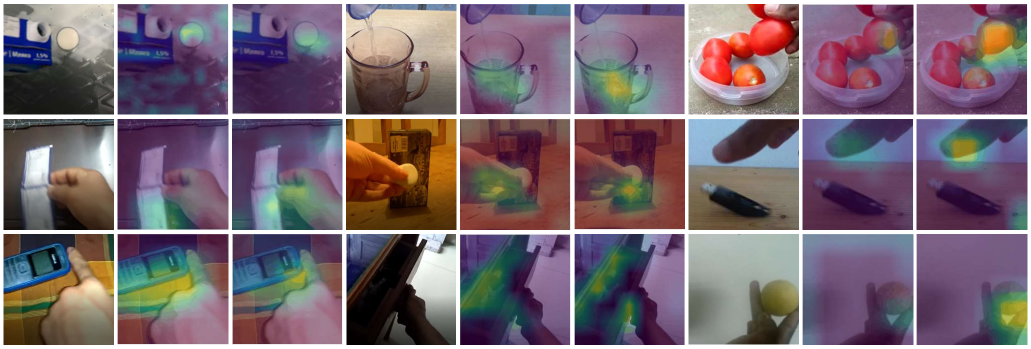


Fig. 4: Visualization of activation maps with CAM. Left: video, Middle: Baseline, Right: ShiftFormer. The activation map is on the center frame. These visualizations show that the baseline method cannot focus on motion-related regions, while our ShiftFormer can locate more motion-related parts benefit from our proposed ShiftFormer spatial-temporal modeling.

TABLE I: Comparison with other methods on Something-Something v2 and Sthv2-top50 datasets. For a fair comparison, all methods use random initialization.

Method	FLOPs (G)	Param (M)	Memory (MiB)	Train-time (h)	Sthv2		Sthv2-top50	
					Top-1 Acc.(%)	Top-5 Acc.(%)	Top-1 Acc.(%)	Top-5 Acc.(%)
Sparse Local Global	590	121.4	31034	114	23.52	53.61	15.97	44.38
Joint Space-Time	359.14	85.9	29630	104	26.18	57.39	18.89	47.52
Divided Space-Time	403.65	121.4	30320	112	27.29	60.84	20.29	49.72
Swin-B	321	88.83	16269	92	32.46	62.81	26.17	58.26
ShiftFormer (ours)	**163.73**	**58.92**	**7103**	**27**	**50.27**	**78.27**	**32.92**	**66.63**

TABLE II: Comparison with different shifted pixels.

D_{space}	D_{time}	Top-1 Acc.(%)	Top-5 Acc.(%)
1	2	26.07	58.87
2	1	27.72	61.41
1	1	30.65	64.01
2	2	**32.92**	**66.63**

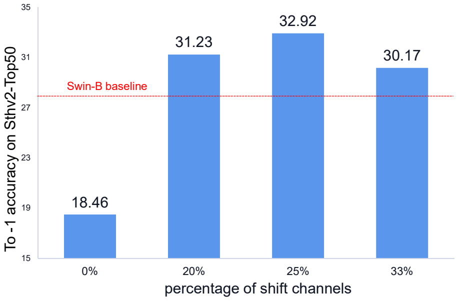


Fig. 5: Ablation analysis on the percentage of shifted channels. We plot the top-1 classification accuracy on Sthv2-top50. The red line indicates Video Swin Transformer (Swin-B) baseline.

TABLE III: Ablation on different positional shifting. ShiftFormer with space-time positional shifting achieved the best performance.

Positional Shifting	Top-1 Acc.(%)	Top-5 Acc.(%)
None	18.46	48.32
Time-only	24.83	51.29
Space-only	25.88	53.28
Space-Time	**32.92**	**66.63**

is Swin-B. In this visualization, we take 32 frames as input and draw the activation map on the keyframes. These visualizations show that the baseline method cannot focus on motion-related regions, while our ShiftFormer can locate more motion-related parts that benefit from our proposed ShiftFormer spatial-temporal modeling. For example, our ShiftFormer focuses more on hand motion with interactive objects, while temporal convolution may focus on the background.

D. Ablation Study

Percentage of shifted channels The scale of our shift channels is a hyperparameter of ShiftFormer, and the scale of the shift is set to 25% in this work. We set the scale of the shifting channels as 20%, 25%, and 30%. The results are shown in Fig. 5, which shows that the hyperparameters have a more obvious effect on the model's performance. When the percentage of shifted channels is set to 0%, the accuracy rate drops severely. Adjusting the ratio of shifted channels can improve the performance. All the settings achieve better accuracy than the Video Swin Transformer (Swin-Base) baseline.

Shifted pixels For the number of shifted pixels, we performed a thorough exploration. The number of pixels we shift in the spatial and temporal dimensions is denoted by D_{space} and D_{time}. The results are shown in Tab. II, which show that the model achieves the highest accuracy on Sthv2-top50 when shift two pixels in the shift operation. This is not difficult to understand because of the increased interaction in

TABLE IV: Comparison with different number of directions.

Shift directions	Convergence speed (epoch)	Top-1 Acc.(%)	Top-5 Acc.(%)
6	100	**32.92**	**66.63**
26	80	31.62	64.27

spatial-temporal location. Also, to explore the variability in the number of pixels shift in space and time, we set different numbers of pixels shift in spatial and temporal locations. For example, moving one pixel in the spatial position while moving two pixels in the temporal position. With this setting, the accuracy of the model in Sthv2-top50 is slightly worse than the default setting.

The importance of ST-shift To investigate the importance of our learned spatiotemporal shifting, we also conduct experiments with a few variants of ShiftFormer that use: (1) no shifting, (2) time-only shifting, (3) space-only shifting, and (4) space-time shifting. We report these results in Tab. III. We found that the variants of our model that use space-time position shifting yield the best accuracy on Sthv2-top50. Using space-only shifting leads to worse results on Sthv2-top50 because this dataset requires complex temporal reasoning.

The number of directions of movement The shift operation facilitates the communication of features between pixels by moving them in different directions. Based on this, we tried to perform shift operations in multiple directions. By default, channels are shifted in 6 directions: top, bottom, left, right, front, and back. We added 20 directions to the default shift directions, including upper left, upper left front, lower right, and lower right back. In total, 26 directions are shifted. The experimental results on Sthv2-top50 are shown in Tab. IV. It can be seen that after increasing the shift directions, the final performance is not very sensitive to this parameter, and the model convergence speed becomes significantly faster.

V. Conclusion

We propose an attention-free ShiftFormer to solve the problem of memory explosion in the video Transformer. It is a newly designed spatial-temporal shift operation that only shifts a small portion of the channels along the temporal and spatial dimensions. ShiftFormer is a cost-effective and efficient method for video understanding. It can save 56.34% of memory usage and achieve 3.41× faster training while performing even better than the powerful Video Swin Transformer for video recognition tasks on Something Something v2 when both using random initialization. In addition, experiments show that attention may not be necessary for the superiority of video Transformer in video understanding tasks.

References

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[2] A. Bulat, J. M. Perez Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos, "Space-time mixing attention for video transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 594–19 607, 2021.

[3] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

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

[5] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[7] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6824–6835.

[8] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.

[9] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[10] J. Li, Y. Yan, S. Liao, X. Yang, and L. Shao, "Local-to-global self-attention in vision transformers," *arXiv preprint arXiv:2107.04735*, 2021.

[11] K. Li, Y. Wang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unified transformer for efficient spatiotemporal representation learning," *arXiv preprint arXiv:2201.04676*, 2022.

[12] D. Lian, Z. Yu, X. Sun, and S. Gao, "As-mlp: An axial shifted mlp architecture for vision," *arXiv preprint arXiv:2107.08391*, 2021.

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

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[16] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 519–16 529.

[17] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.

[18] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek *et al.*, "Resmlp: Feedforward networks for image classification with data-efficient training," *arXiv preprint arXiv:2105.03404*, 2021.

[19] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[20] G. Wang, Y. Zhao, C. Tang, C. Luo, and W. Zeng, "When shift operation meets vision transformer: An extremely simple alternative to attention mechanism," *arXiv preprint arXiv:2201.10801*, 2022.

[21] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.

[22] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S2-mlp: Spatial-shift mlp architecture for vision," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 297–306.