ECCV 2022

MLCF

Spatiotemporal Self-attention Modeling with Temporal Patch Shift for Action Recognition

Wangmeng Xiang[1,2⋆], Chao Li[2], Biao Wang[2], Xihan Wei[2], Xian-Sheng Hua[2], and Lei Zhang[1⋆⋆]

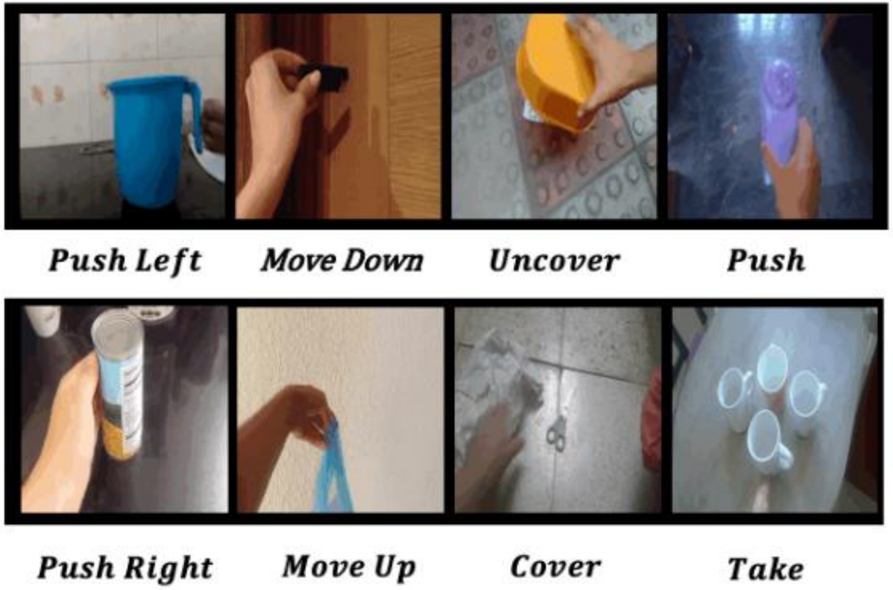
[1] The Hong Kong Polytechnic University, Hong Kong SAR, China
{cswxiang,cslzhang}@comp.polyu.edu.hk
[2] DAMO Academy, Alibaba, Hangzhou, China
{lllcho.lc,wb.wangbiao,xihan.wxh,xiansheng.hxs}@alibaba-inc.com

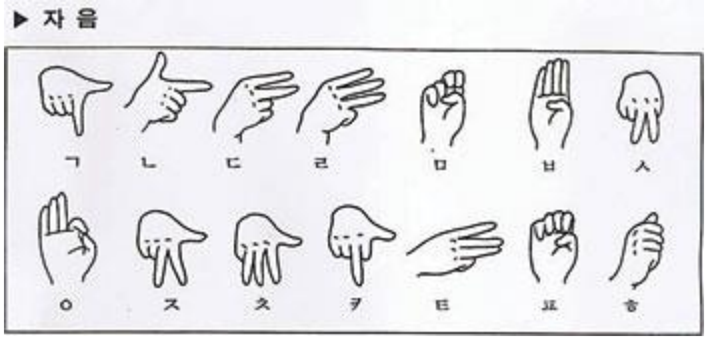
Reviewed by Susang Kim

Contents

1.Introduction : Action Recognition

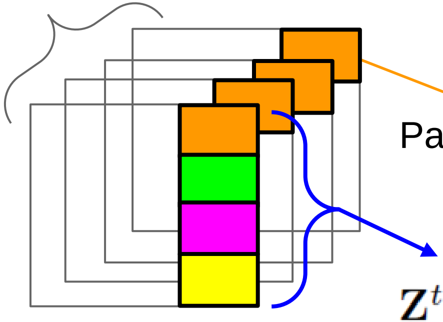
MLCF





https://www.yangsanilbo.com/news/articleView.html?idxno=29589

$\mathbf{Z} \in \mathbb{R}^{D \times T \times N}$ $X \in \mathbb{R}^{F \times H \times W \times C}$



$T = F/s$

$s \times k \times k$

Patch : p

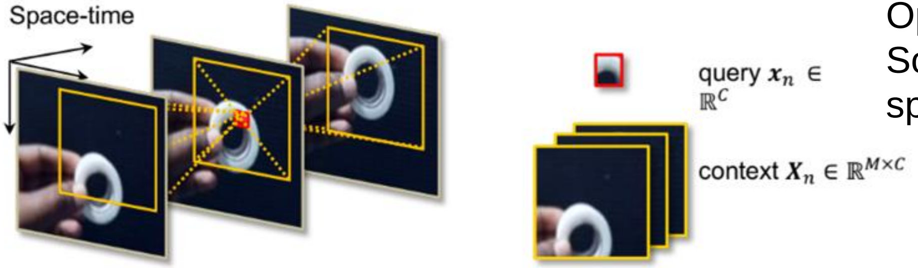
$\mathbf{Z}^t \in \mathbb{R}^{D \times N}$

Video based action recognition

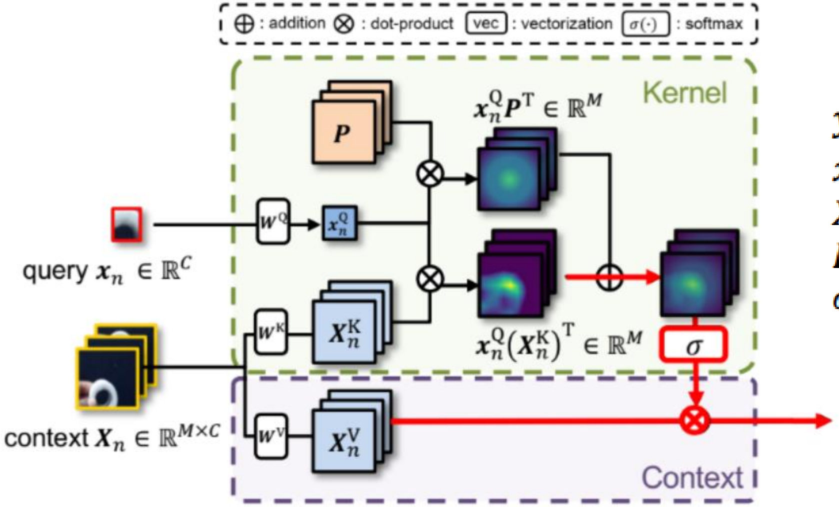
2D image-based => 3D(+temporal) video-based tasks

R. Goyal, et al. "The something something video database for learning and evaluating visual common sense," ICCV, 2017.

2.Preliminaries : Self-Attention in Space and Time



Operation is very computationally and memory expensive. So previous efforts to reduce the computational burden of spatiotemporal multihead Self-Attention.



$\boldsymbol{y}_n \in \mathbb{R}^C$
$\boldsymbol{x}_n^{\mathrm{Q}} \in \mathbb{R}^C$
$\boldsymbol{X}_n^{\mathrm{K}}, \boldsymbol{X}_n^{\mathrm{V}}, \boldsymbol{P} \in \mathbb{R}^{M\times C}$
$\boldsymbol{P} \in \mathbb{R}^{M\times C}$
$\sigma(\cdot)$: softmax

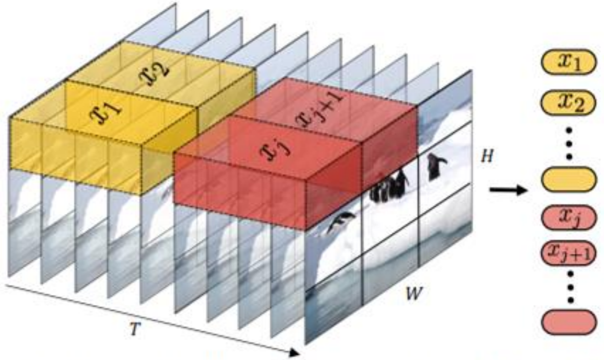


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

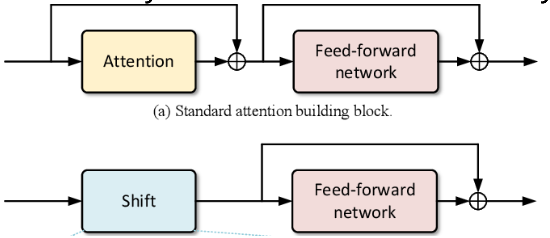
$$\boldsymbol{y}_n = \underbrace{\sigma\left(\boldsymbol{x}_n^{\mathrm{Q}}(\boldsymbol{X}_n^{\mathrm{K}})^{\mathrm{T}} + \boldsymbol{x}_n^{\mathrm{Q}}\boldsymbol{P}^{\mathrm{T}}\right)}_{\text{Kernel}}\underbrace{\boldsymbol{X}_n^{\mathrm{V}}}_{\text{Context}}$$

A. Arnab et al., “ViViT: A Video Vision Transformer,” ICCV, 2021.
M. Kim et al., “Relational Self-Attention: What’s Missing in Attention for Video Understanding,” NeurIPS, 2021.

2.Preliminaries : Shift Operation

Shift operation does not hold any parameter or arithmetic operation

1. Shift Operation Meets Vision Transformer : The attention layers in ViT are substituted by shift operation

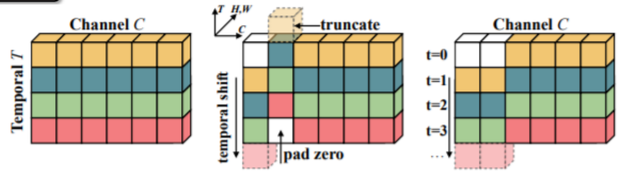


(a) Standard attention building block.

2. Temporal Shift Module (TSM) :

Shifts the channels along the temporal dimension,

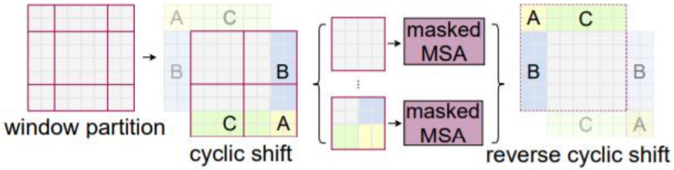
so computationally free and strong spatio-temporal modeling ability.



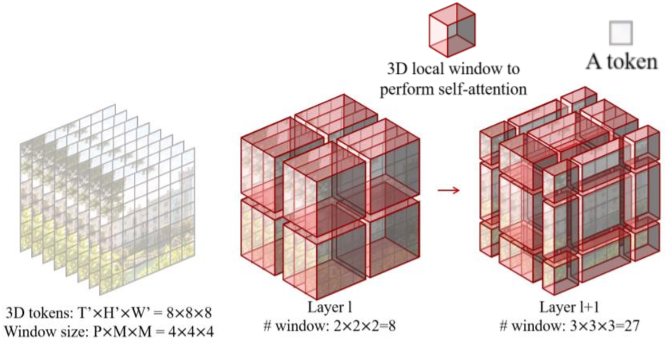
original bi-direction uni-direction

3. Video Swin Transformer : non-overlapping shifted window mechanism.

$$
\begin{aligned}
\hat{\mathbf{z}}^{l} &= \text{3DW-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1},\\
\mathbf{z}^{l} &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l}\right)\right) + \hat{\mathbf{z}}^{l},\\
\hat{\mathbf{z}}^{l+1} &= \text{3DSW-MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l},\\
\mathbf{z}^{l+1} &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},
\end{aligned}
$$



3D Local Window => 3D Shifted Window



Video Swin Transformer blocks(3D shifted window)

Liu, Ze, et al. "Video swin transformer." CVPR 2022.
Wang, Guangting, et al. "When shift operation meets vision transformer.", AAAI 2022
Lin, Ji et al, "Tsm: Temporal shift module for efficient video understanding." ICCV 2019.
A Vision Transformer without Attention : https://keras.io/examples/vision/shiftvit/

3.Method : Overview of Temporal Patch Shift (TPS)

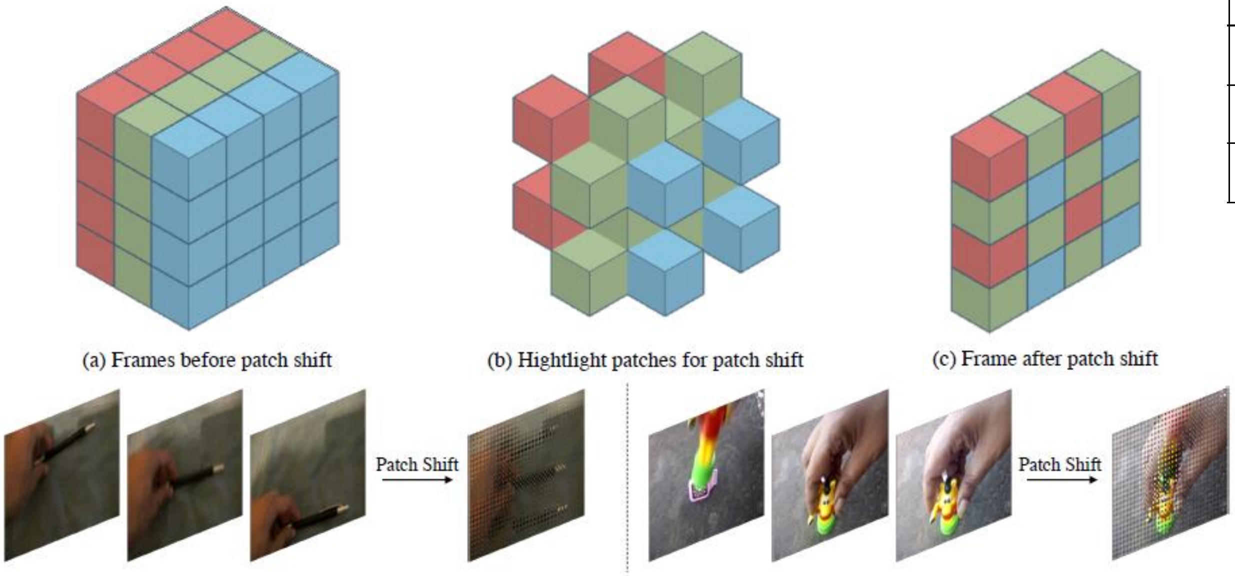


Fig. 1. An example of temporal patch shift for three adjacent frames.

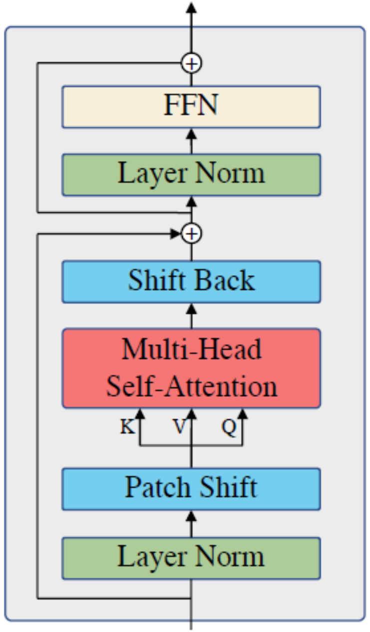
0	1	0	1
-1	2	-1	2
0	1	0	1
-1	2	-1	2

(b) Pattern B

Bayer filter

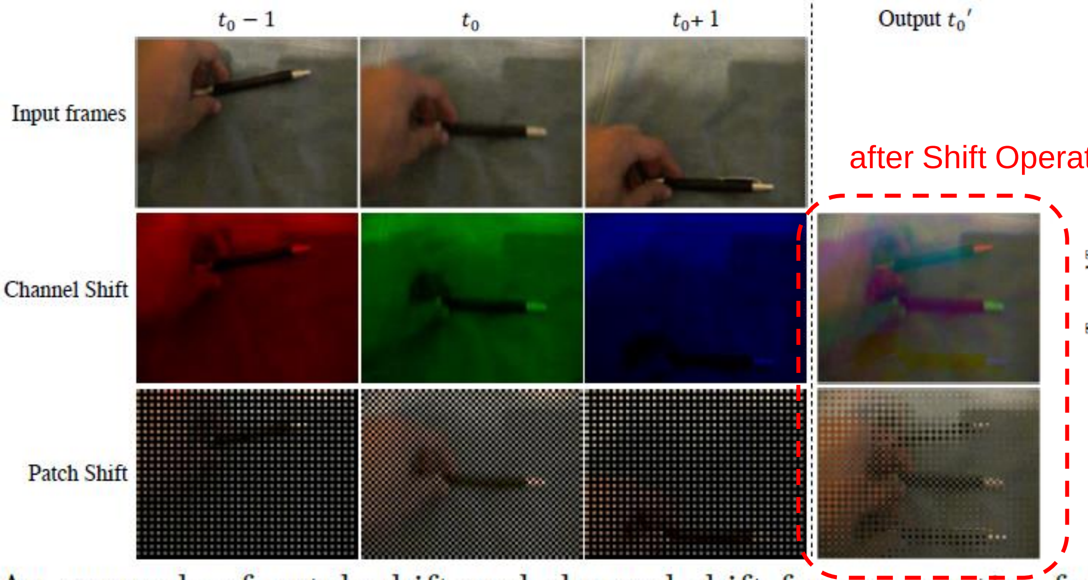
$$\{\mathbf{i}', \mathbf{Z}'_{l-1}\} = \mathrm{PatchShift}(\mathbf{p}, \mathbf{i}, \mathbf{Z}_{l-1}),$$

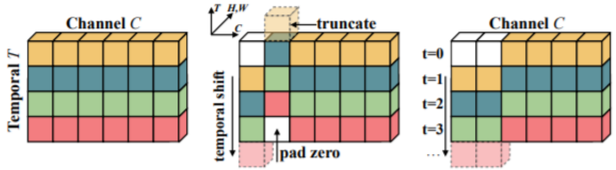
$$Q_l, K_l, V_l = W_l^Q \mathbf{Z}'_{l-1}, W_l^K \mathbf{Z}'_{l-1}, W_l^V \mathbf{Z}'_{l-1},$$



TPS Block

3.Method : Patch and Channel shifts





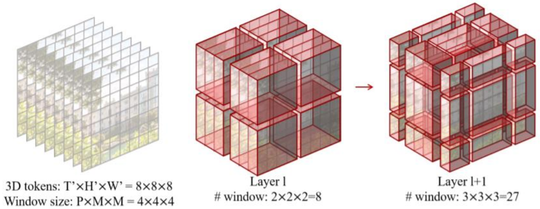
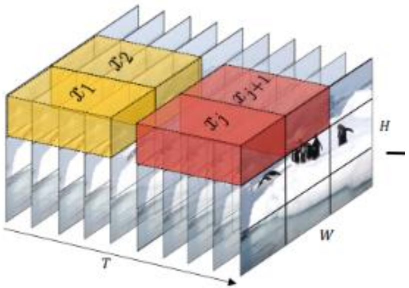


Fig. 3. An example of patch shift and channel shift for consecutive frames.

Patch Shift	Channel Shift
Space-wise sparse and Channel-wise dense	Space-wise dense and Channel-wise sparse.
the global channel information for each patch	partial channel information
both can capture the motion of action / zero parameter and low-cost temporal modeling methods	

3.Method : Notation of Temporal Patch Shift (TPS)



$$\mathbf{Z} \in \mathbb{R}^{D \times T \times N}$$

$$\mathbf{Z}^t = [\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_N], \qquad \mathbf{Z}^t, \mathbf{Z}^{t'} \in \mathbb{R}^{D \times N}$$

patch features for current frame t

$$\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N], \qquad \mathbf{a}_i \in \mathbb{R}^D$$

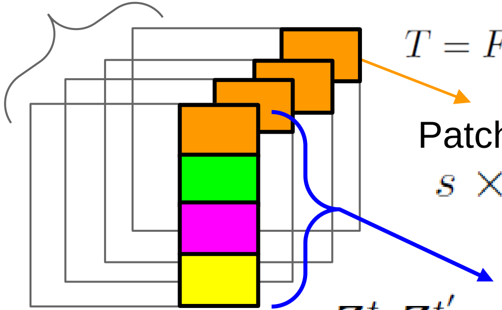
the vector of channel shifts for patch i

$$\hat{\mathbf{Z}}^t = \mathbf{A} \odot \mathbf{Z}^{t'} + (\mathbf{1} - \mathbf{A}) \odot \mathbf{Z}^t,$$

output image patches after shift operation

$$\mathbf{p} = \{0, 1\} \qquad \mathbf{A} = [0, 1, 0, 1, \ldots]$$

shifting one patch for every two patches



$$T = F/s$$

Patch(cube)

$$s \times k \times k$$

$$\mathbf{Z}^t, \mathbf{Z}^{t'} \in \mathbb{R}^{D \times N}$$

$$X \in \mathbb{R}^{F \times H \times W \times C}$$

$$\mathbf{x}^{(t,p)} \in \mathbb{R}^{3sk^2}$$

(a)

0	0	0
0	1	0
0	0	0

(b)

0	1	0
-1	0	-1
0	1	0

(c)

-4	1	2
-1	0	3
-2	-3	4

Fig. 2. Examples of patch shift patterns when patch number is 3×3.

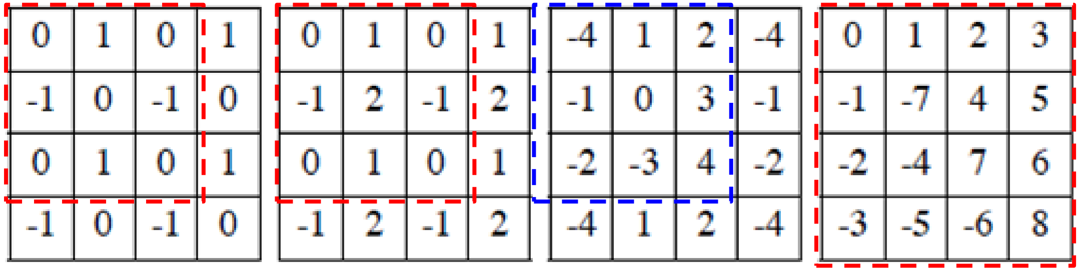
$$E \in \mathbb{R}^{D \times 3sk^2}$$

weight of a linear layer

$$\mathbf{z}_0^{(t,p)} = E\mathbf{x}^{(t,p)} + \mathbf{e}_{pos}^{(t,p)}$$

learnable spatiotemporal positional embedding

3.Method : Shift Patterns

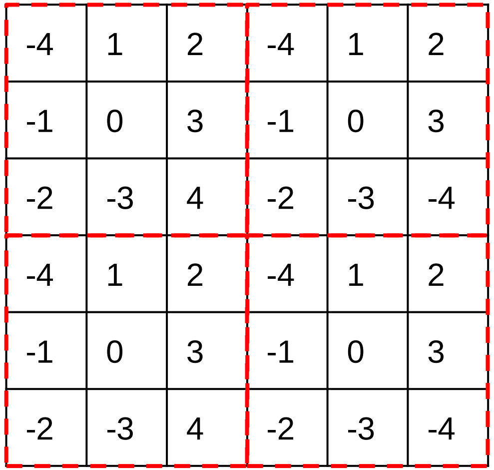


(a) Pattern A (b) Pattern B (c) Pattern C (d) Pattern D

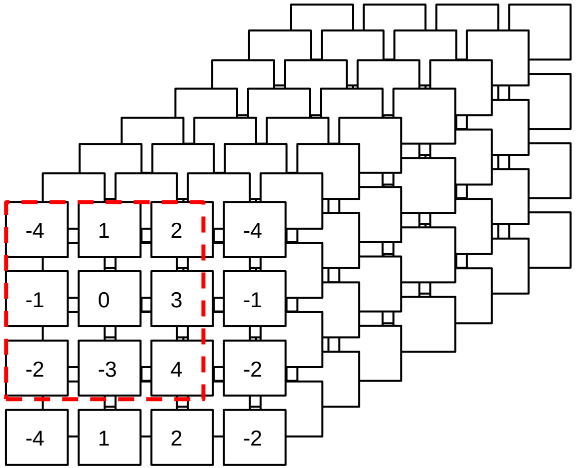
We use cyclic padding in for patches that exceed the temporal boundary

(b) Shift patterns

Pattern	Top-1	Top-5
A-3	48.6	77.8
B-4	50.7	79.3
C-9	**51.8**	**80.3**
D-16	50.0	79.5

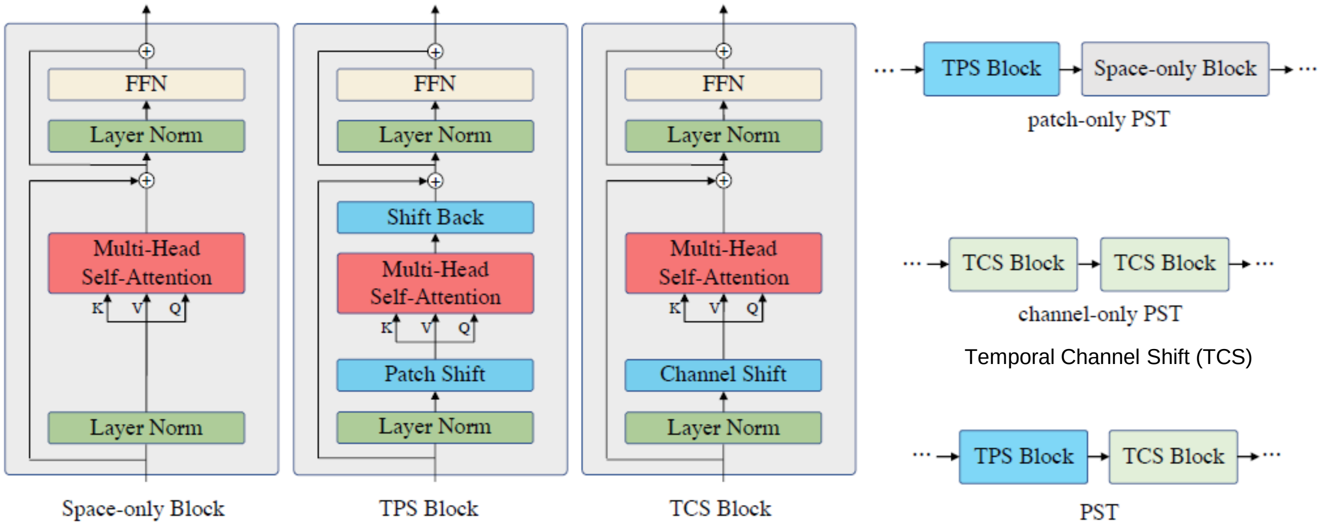


Pattern C



3.Method : Patch Shift Transformers(PST)

MLCF



Insert one TPS(Temporal Patch Shift) block for every two SA modules (to alleviate gathering information in a sparse manner and sacrifice self-attention within frames)

Temporal Channel Shift (TCS)

Fig. 4. An overview of building blocks and variants of PST.

$$\hat{\mathbf{Z}}_l = \mathrm{SA}(\mathrm{LN}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1},$$

$$\mathbf{Z}_l = \mathrm{FFN}(\mathrm{LN}(\hat{\mathbf{Z}}_l)) + \hat{\mathbf{Z}}_l,$$

$$Q_l, K_l, V_l = W_l^Q \mathbf{Z}_{l-1}, W_l^K \mathbf{Z}_{l-1}, W_l^V \mathbf{Z}_{l-1}$$

$$\hat{\mathbf{Z}}_l = \mathrm{SoftMax}(Q_l K_l^T / \sqrt{d}) V_l,$$

$$\{\mathbf{i}', \mathbf{Z}'_{l-1}\} = \boxed{\mathrm{PatchShift}}(\mathbf{p}, \mathbf{i}, \mathbf{Z}_{l-1}),$$

$$Q_l, K_l, V_l = W_l^Q \mathbf{Z}'_{l-1}, W_l^K \mathbf{Z}'_{l-1}, W_l^V \mathbf{Z}'_{l-1},$$

$$\hat{\mathbf{Z}} = \boxed{\mathrm{ShiftBack}}(\mathrm{SoftMax}(Q_l K_l^T / \sqrt{d} + B(\mathbf{i}')) V_l),$$

3.Method : the computation burdens

Attention	SA-Complexity
Joint	$\mathcal{O}(N^2T^2)$
Divide	$\mathcal{O}(N^2T + T^2N)$
Sparse/Local	$\mathcal{O}(\alpha N^2T^2)$
PatchShift	$\mathcal{O}(N^2T)$

Joint : Attention operation

Divide : Spatial Only + Temporal Only

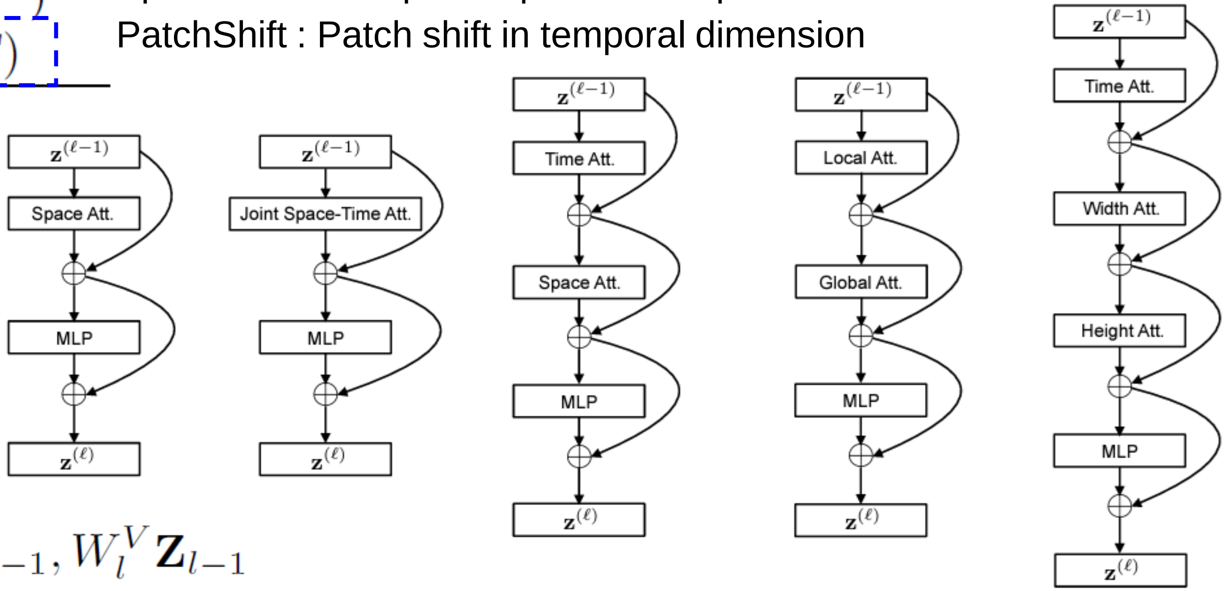
Sparse : Subsample in space or temporal dimension

PatchShift : Patch shift in temporal dimension

$$\hat{\mathbf{Z}}_l = \mathrm{SA}(\mathrm{LN}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1},$$

$$\mathbf{Z}_l = \mathrm{FFN}(\mathrm{LN}(\hat{\mathbf{Z}}_l)) + \hat{\mathbf{Z}}_l,$$

$$Q_l, K_l, V_l = W_l^Q \mathbf{Z}_{l-1}, W_l^K \mathbf{Z}_{l-1}, W_l^V \mathbf{Z}_{l-1}$$



Space Attention (S) | Joint Space-Time Attention (ST) | Divided Space-Time Attention (T+S) | Sparse Local Global Attention (L+G) | Axial Attention (T+W+H)

Bertasius, G., Wang, H., & Torresani, Is space-time attention all you need for video understanding? , ICML 2021

4.Experiments : Setup

MLCF

[Models]
Backbone : Swin Transformer with PST-T, PST-B an increase in model size
32 frames as input and the tubelet embedding strategy in ViViT with patch size 2×4×4 by default.
PST-T† and PST-B†, which doubles the temporal attention window to 2 with slightly increased computation

[Training]
Images to 256 and then apply center cropping of 224×224. random flip, AutoAugment for augmentation.
AdamW with the cosine learning rate schedule for network training

[Testing]
On Something-something V1&V2 and Diving-48 V2, uniform sampling and center crop
(or three-crop) testing are adopted. On Kinetics400, we adopt the dense sampling strategy as in with 4 view, three-crop testing.

Arnab, Anurag, et al. "Vivit: A video vision transformer." CVPR 2021.

4.Experiments : Datasets

Something-something v1 & v2 (SS-V1 & V2) are both large-scale action recognition benchmarks, including 108k and 220k action clips. Both are 174 classes.

- temporal related

Pouring [something]

Kinetics400 is a action recognition dataset, which contains 400 classes, with at least 400 video clips for each class. Each clip is trimmed to around 10s.

- less temporal related

riding a bike

Diving-48 V2 is fine-grained action benchmark that is heavily dependent on temporal modeling containing 18k videos with 48 diving classes

- temporal related

['Forward', '15som', 'NoTwis', 'PIKE']

Kay, Will, et al. "The kinetics human action video dataset." *arXiv preprint arXiv:1705.06950* (2017).

4.Experiments : Ablation study

All the experiments are conducted on SS V1 with Swin-Tiny as backbone (IN-1K pretrained).

(a) Patch distribution

	Distribution	Top-1	Top-5
	None	40.6	71.4
	Center-one	45.3	75.1
1/4	Uneven	45.3	75.5
	Even-2	46.2	76.1
	Even-3	**48.6**	**77.8**

(b) Shift patterns

Pattern	Top-1	Top-5
A-3	48.6	77.8
B-4	50.7	79.3
C-9	**51.8**	**80.3**
D-16	50.0	79.5

(c) Number of stages with TPS

Stage 1	Stage 2	Stage 3	Stage 4	Top-1	Top-5
✓				47.3	77.0
✓	✓			48.4	77.6
✓	✓	✓		50.4	79.1
✓	✓	✓	✓	**51.8**	**80.3**

increases when the temporal field grows.

the number of shifting of the total patches,
shifting 1/4 patches to previous and 1/4 to next (even-3).

(d) Shift back, Alternative shift and shift RPE

Shift back	Alternative	Shift RPE	Top-1	Top-5
	✓	✓	47.3	77.0
✓		✓	46.4	76.6
✓	✓		46.1	76.0
✓	✓	✓	**51.8**	**80.3**

(e) Comparison of spatiotemporal attentions

	FLOPs	Memory	Top-1	Top-5
Avgpool	72G	3.7G	40.6	71.4
Joint	106G	20.2G	51.5	80.0
Local	88G	11G	49.9	79.2
Sparse	72G	4.0G	42.7	74.0
Channel-only	72G	3.7G	51.2	79.7
Patch-only	72G	3.7G	51.8	80.3
PST	**72G**	**3.7G**	**52.2**	**80.3**

TPS block for every two SA modules (alternative shift in short)
Shift RPE represents whether relative positions are shifted alongside patches.

4.Experiments : Comparison with SOTA

MLCF

Table 3. Comparisons with the other methods on Something-something V1 & V2.

Model	Pretrain	Crops × Clips	FLOPs	Params	Sthv1 Top-1	Sthv1 Top-5	Sthv2 Top-1	Sthv2 Top-5
TSM [22]	K400	3 × 2	65G	24.3M	-	-	63.4	88.5
TEINet [26]	IN-1K	1 × 1	66G	30.4M	49.9	-	62.1	-
TEA [20]	IN-1K	1 × 1	70G	24.3M	51.9	80.3	-	-
TDN [38]	IN-1K	1 × 1	72G	24.8M	53.9	82.1	65.3	89.5
ACTION-Net [42]	IN-1K	1 × 1	70G	28.1M	-	-	64.0	89.3
SlowFast R101, 8x8 [13]	K400	3 × 1	106G	53.3M	-	-	63.1	87.6
MSNet [18]	IN-1K	1 × 1	101G	24.6M	52.1	82.3	64.7	89.4
blVNet [11]	IN-1K	1 × 1	129G	40.2M	-	-	65.2	90.3
Timesformer-HR [2]	IN-21K	3 × 1	1703G	121.4M	-	-	62.5	-
ViViT-L/16x2 [1]	IN-21K	3 × 1	903G	352.1M	-	-	65.9	89.9
MViT-B, 64×3 [9]	K400	3 × 1	455G	36.6M	-	-	67.7	90.9
Mformer-L [29]	K400	3 × 1	1185G	86M	-	-	68.1	91.2
X-ViT [3]	IN-21K	3 × 1	283G	92M	-	-	66.2	90.6
SIFAR-L [10]	K400	3 × 1	576G	196M	-	-	64.2	88.4
Video-Swin [25]	K400	3 × 1	321G	88.1M	-	-	69.6	92.7
PST-T	IN-1K	1 × 1	72G	28.5M	52.2	80.3	65.7	90.2
	IN-1K	3 × 1			52.8	80.5	66.4	90.2
	K400	1 × 1			53.2	82.2	66.7	90.6
	K400	3 × 1			53.6	82.2	67.3	90.5
PST-T†	K400	3 × 1	74G		54.0	82.3	67.9	90.8
PST-B	IN-21K	1 × 1	247G	88.8M	55.3	81.9	66.7	90.7
	IN-21K	3 × 1			55.6	82.2	67.4	90.9
	K400	1 × 1			57.4	83.2	68.7	91.3
	K400	3 × 1			57.7	83.4	69.2	91.9
PST-B†	K400	3 × 1	252G		**58.3**	**83.9**	**69.8**	**93.0**

† denotes doubles the temporal attention window

4.Experiments : Comparison with SOTA

MLCF

Table 4. Comparisons with the state-of-the-art methods on Kinetics400.

Model	Pretrain	Crops × Clips	FLOPs	Params	Top-1	Top-5
I3D [4]	IN-1K	1×1	108G	28.0M	72.1	90.3
NL-I3D [41]	IN-1K	6×10	32G	35.3M	77.7	93.3
CoST [19]	IN-1K	3×10	33G	35.3M	77.5	93.2
SlowFast-R50 [13]	IN-1K	3×10	36G	32.4M	75.6	92.1
X3D-XL [12]	-	3×10	48G	11.0M	79.1	93.9
TSM [22]	IN-1K	3×10	65G	24.3M	74.7	91.4
TEINet [26]	IN-1K	3×10	66G	30.4M	76.2	92.5
TEA [20]	IN-1K	3×10	70G	24.3M	76.1	92.5
TDN [38]	IN-1K	3×10	72G	24.8M	77.5	93.2
Timesformer-L [2]	IN-21K	3×1	2380G	121.4M	80.7	94.7
ViViT-L/16x2 [1]	IN-21K	3×1	3980G	310.8M	81.7	93.8
X-ViT [3]	IN-21K	3×1	283G	92M	80.2	94.7
MViT-B, 32×3 [9]	IN-21K	1×5	170G	36.6M	80.2	94.4
MViT-B, 64×3 [9]	IN-21K	3×3	455G	36.6M	81.2	95.1
Mformer-HR [29]	K400	3×1	959G	86M	81.1	95.2
TokenShift-HR [45]	IN-21K	3×10	2096G	303.4M	80.4	94.5
SIFAR-L [10]	IN-21K	3×1	576G	196M	**82.2**	**95.1**
Video-Swin [24]	IN-21K	3×4	282G	88.1M	**82.7**	**95.5**
PST-T	IN-1K	3×4	72G	28.5M	78.2	92.2
PST-T†	IN-1K	3×4	74G	28.5M	78.6	93.5
PST-B	IN-21K	3×4	247G	88.8M	81.8	95.4
PST-B†	IN-21K	3×4	252G	88.8M	**82.5**	**95.6**

PST-B† achieves 82.5% with less computation overheads

4.Experiments : Latency, throughput and memory

Table 6. Memory and latency comparison on Something-something V1&V2 (Measured on NVIDIA Tesla V100 GPU)

Methods	FLOPs	Param	Memory	Latency	Throughput	Sthv1		Sthv2	
						Top-1	Top-5	Top-1	Top-5
2D Swin-T	72G		1.7G	29ms	35.5 v/s	40.6	71.4	56.7	84.1
Video-Swin-T [24]	106G(↑ 34G)	28.5M	3.0G(↑ 1.3G)	62ms(↑ 33ms)	17.7 v/s	51.5	80.0	65.7	90.1
PST-T	72G		1.7G	31ms(↑ 2ms)	34.7 v/s	**52.2**	**80.3**	**65.7**	**90.2**
2D Swin-B	247G		2.2G	71ms	15.5 v/s	-	-	59.5	86.3
Video-Swin-B [24]	321G(↑ 74G)	88.8M	3.6G(↑ 1.4G)	147ms(↑ 76ms)	7.9 v/s	-	-	69.6	92.7
PST-B†	252G(↑ 5G)		2.4G(↑ 0.2G)	81ms(↑ 10ms)	13.8 v/s	-	-	**69.8**	**93.0**

Table 5. Comparisons with the other methods on Diving-48 V2.

Model	Pretrain	Crops × Clips	FLOPs	Params	Top-1	Top-5
SlowFast R101, 8x8 [13]	K400	3 × 1	106G	53.3M	77.6	-
Timesformer [2]	IN-21K	3 × 1	196G	121.4M	74.9	-
Timesformer-HR [2]	IN-21K	3 × 1	1703G	121.4M	78.0	-
Timesformer-L [2]	IN-21K	3 × 1	2380G	121.4M	81.0	-
PST-T	IN-1K	3 × 1	72G	28.5M	79.2	98.2
	K400	3 × 1	72G		81.2	98.7
PST-T†	K400	3 × 1	74G		82.1	98.6
PST-B	IN-21K	3 × 1	247G	88.1M	83.6	98.5
	K400	3 × 1	$247G$		85.0	98.6
PST-B†	K400	3 × 1	252G		**86.0**	**98.6**

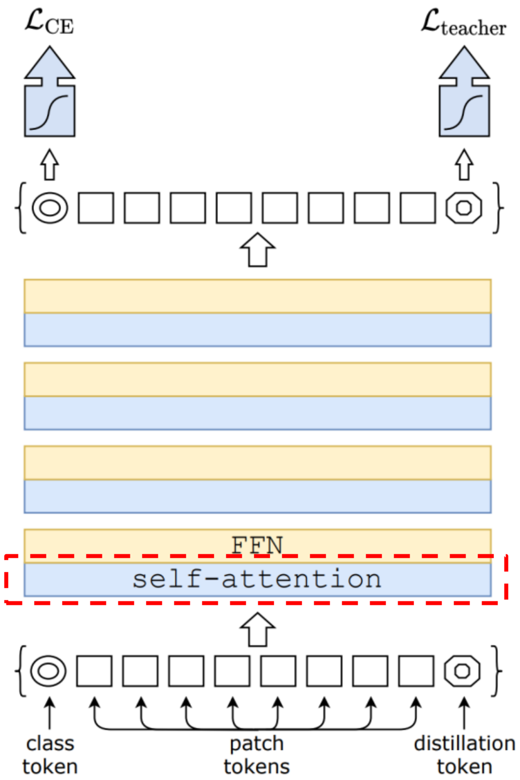
4.Experiments : Additional results on DeiT backbone

MLCF

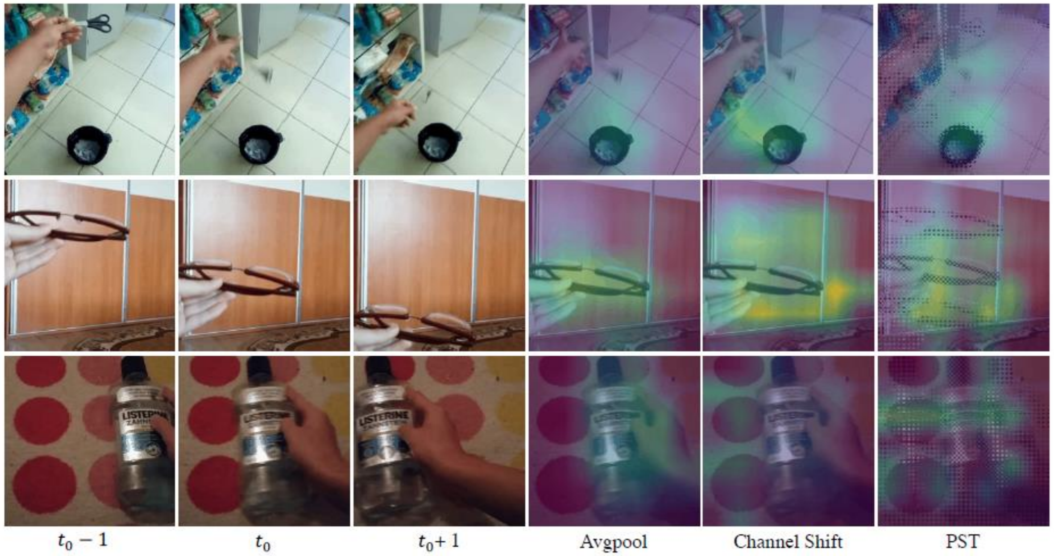
Table 7. More backbones experiments on Kinetics400.

Model	Pretrain	Crops × Clips	FLOPs	Params	Top-1	Top-5
DeiT-S-2D [36]	IN-1K	3×4	74G	22M	73.0	90.7
DeiT-S-TSM	IN-1K	3×4	74G	22M	74.8	91.6
DeiT-S-TPS	IN-1K	3×4	74G	22M	**75.3**	**91.8**

14 × 14 image patches(center cropping of 224×224-16x16 tokens) at every layer and an additional class token. We insert a TPS module in every two blocks of DeiT.



4.Experiments : Visualization results



PST can learn to focus on the motion of objects

5.Conclusions

Discovering the difference between **Patch Shift and Channel Shift.**

TPS is a plug-and-play module and can be easily embedded into many existing 2D transformers **without additional parameters and computation costs.**

The resulted PST is **highly cost-effective in both computation and memory.**
- PST achieved competitive performance comparing to previous methods on the datasets of Something-something V1&V2, Diving-48 and Kinetics400.

PST achieved a **good balance between accuracy and computational cost** for effective action recognition.

Table 6: Ablation study on the 3D shifted window approach with Swin-T on K400.

	Top-1	Top-5
w. 3D shifting	78.8	93.6
w/o temporal shifting	78.5	93.5
w/o 3D shifting	78.1	93.3

Liu, Ze, et al. "Video swin transformer." CVPR 2022.

Thanks
Any Questions?

You can send mail to
Susang Kim(healess1@gmail.com)