Video Swin Transformer

arXiv Jun, 2021

Outline

- Introduction
- Related Works
- Architecture
  - 3D Shifted Window-MSA Module
  - Pre-Training
  - Variants
- Experiments
- Conclusion

Introduction

- Video Swin
  - Pure transformer backbone architecture for video
- Adapts Swin architecture to videos
  - Takes advantage of spatio-temporal locality in videos
  - Swin, extended to the temporal domain

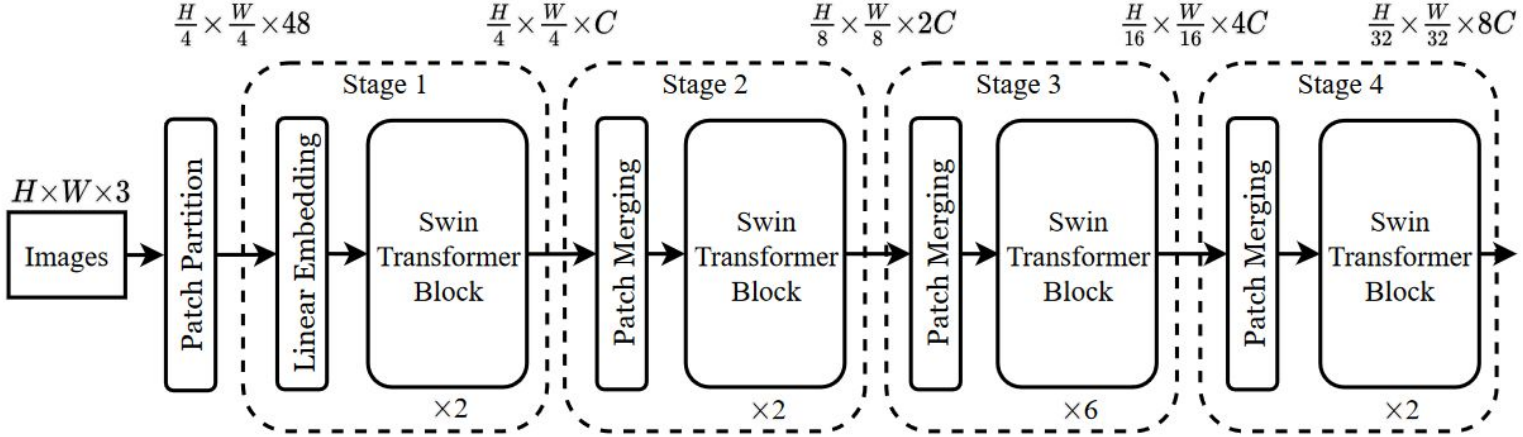
Related Works

- CNNs
  - C3D, I3D - 3D Convolutions
  - P3D - Disentangle spatial and temporal convolutions
- Transformers to Compliment CNNs
  - NLNet
  - GCNet
  - DNL
- Vision Transformers
  - Swin Transformer: Uses locality in images, general purpose backbone for image recognition.
  - TimeSformer: Examines variants of space-time attention; proposes factorized attention
  - ViViT: Examines four factorized spatial and temporal attentions
  - MViT: Reduces computation by pooling attention

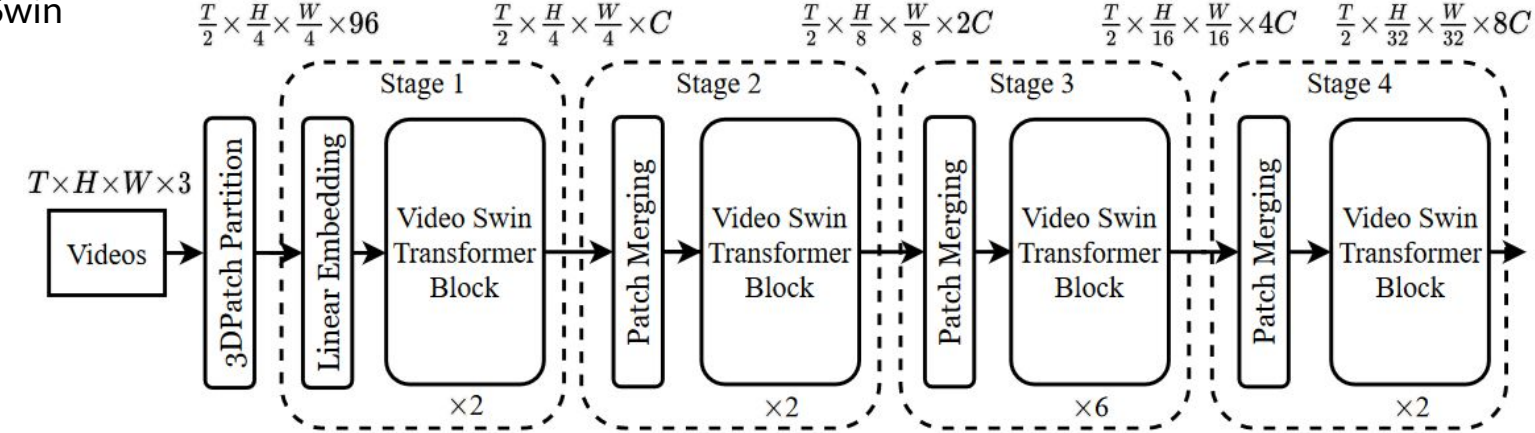
Changed From Swin

- Input includes a time dimension: number of Frames
- Patches are 3D
  - Size 2 x 4 x 4 x 3 (Frames x Height x Width x Channels)
- Clips are broken up into (T/2)*(H/4)*(W/4) 3D tokens
- Each patch token consists of a 96-dimensional features, up from 48
- No downsampling along the temporal dimension
  - Spatial dimension still downsampled the same
- Replaces MSA module with a 3D shifted window based MSA module

Swin

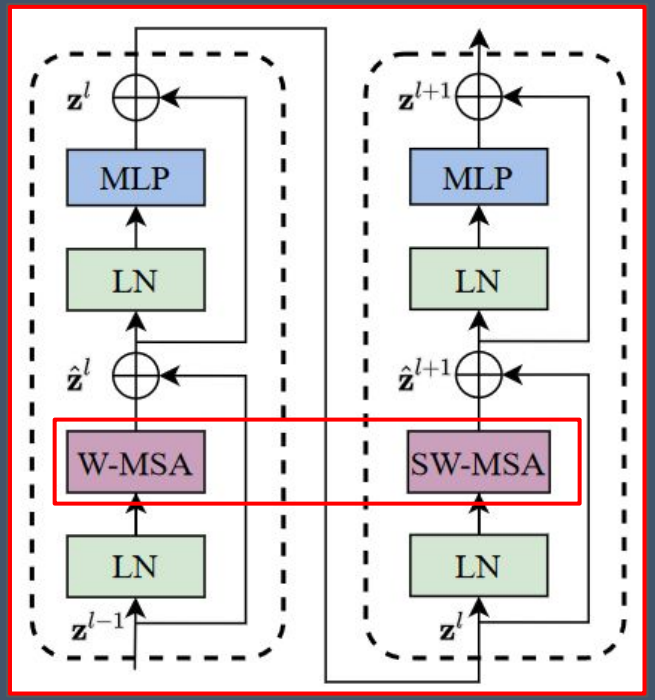


Video Swin

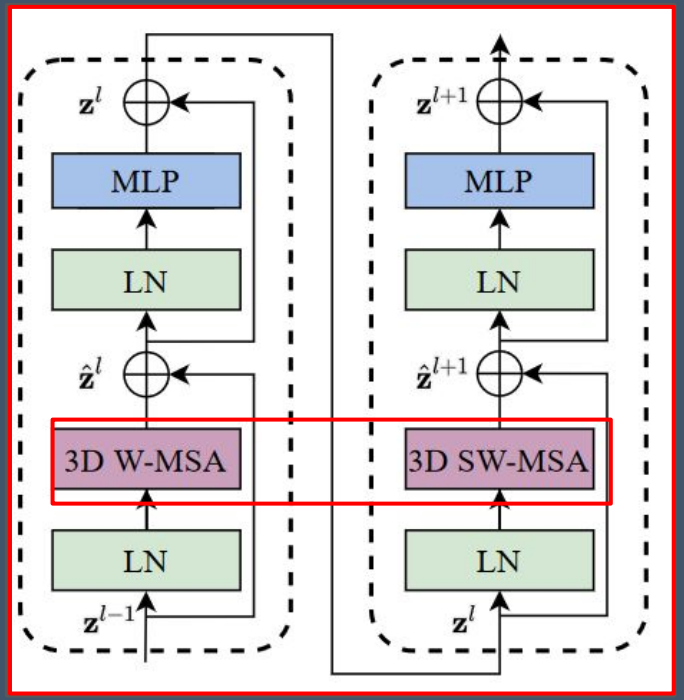


Swin vs Video Swin Transformer Block Architecture

SWIN

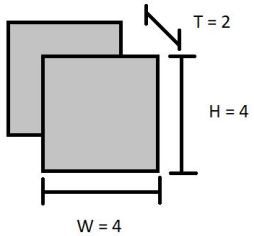


Video SWIN



Window Construction

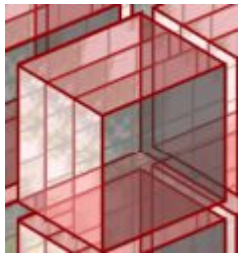
Token



(2 x 4 x 4 x 3)

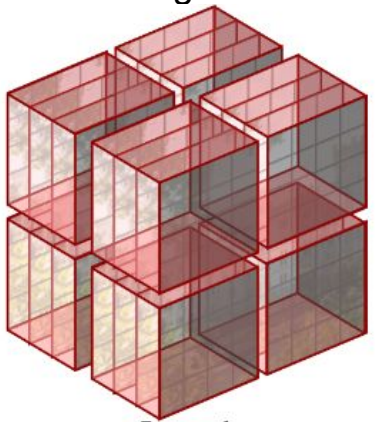
(T x H x W x RGB)

Window

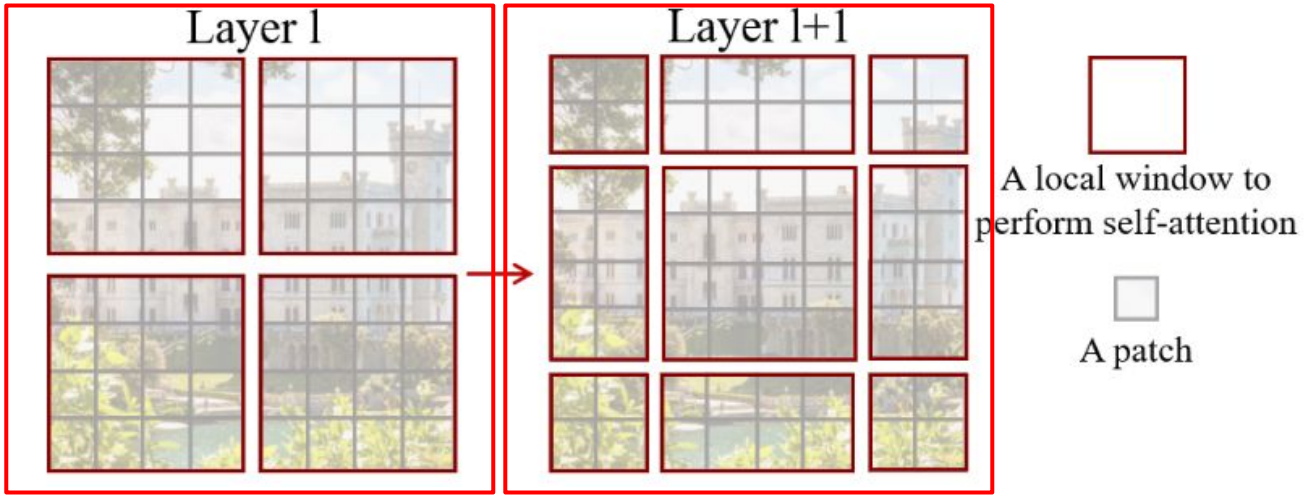


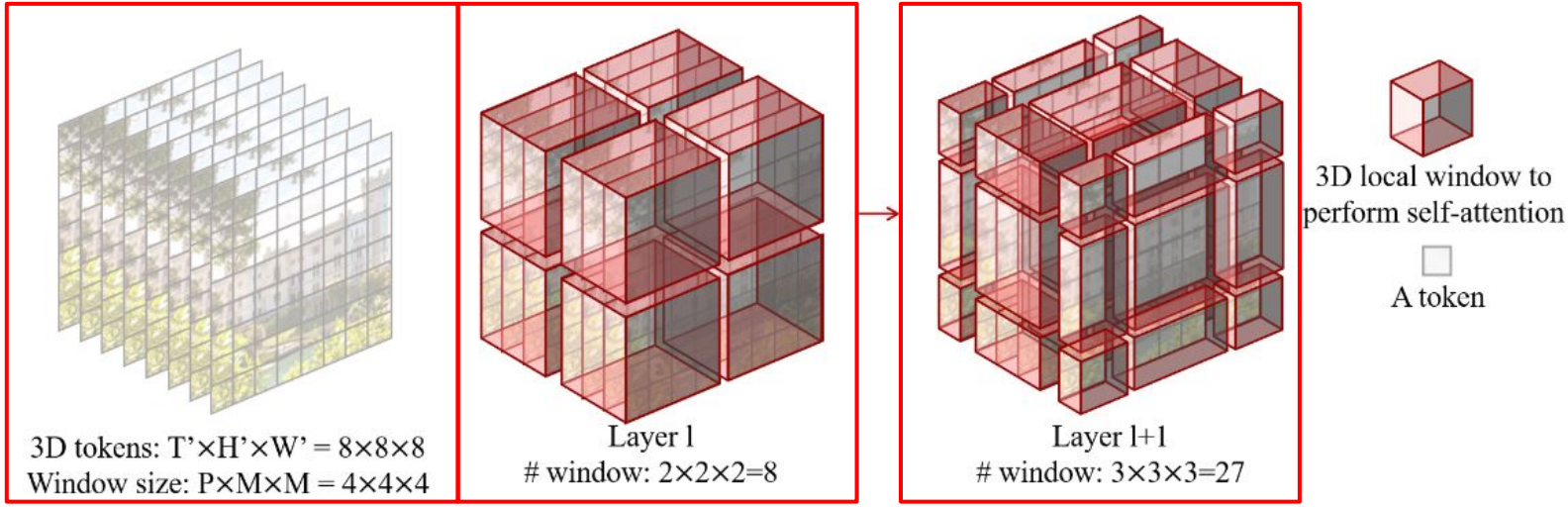
(4 x 4 x 4)

Configuration



(2 x 2 x 2)


Layer l
Layer l+1
A local window to perform self-attention
A patch


3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4
Layer l
window: 2×2×2=8
Layer l+1
window: 3×3×3=27
3D local window to perform self-attention
A token

3D Shifted Window MSA Module

- Global Self-Attention would be too costly for videos

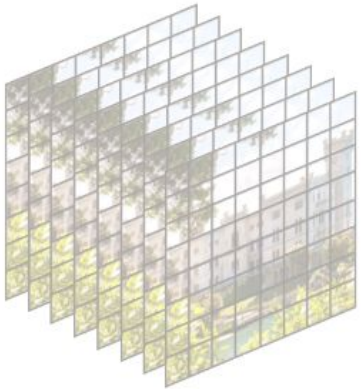
- Solution, as in Swin Transformer:
  - Video Swin advocates for using the spatiotemporal locality of videos
    - Pixels closer in spatiotemporal distance are more likely correlated

Multi-head self-attention (Non-overlapping 3D Windows)

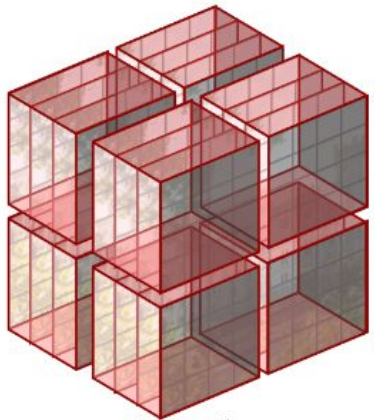
T'×H'×W' 3D tokens and a 3D window size of P×M×M

- T' is # of 3D tokens in the temporal dimension
- H' is # of 3D tokens high
- W' is # of 3D tokens wide

$\lceil \frac{T'}{P} \rceil \times \lceil \frac{H'}{M} \rceil \times \lceil \frac{W'}{M} \rceil$ non-overlapping 3D windows



3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4

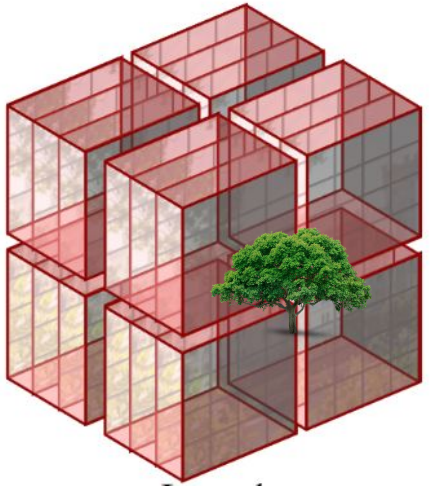


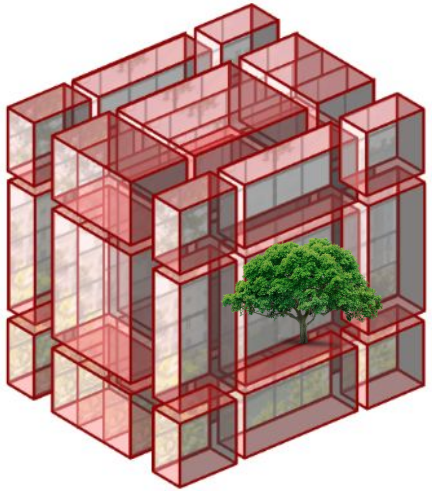
Layer l
window: 2×2×2=8

3D Shifted Windows

- MSA is applied within each non-overlapping 3D window
  - There are no connection between windows

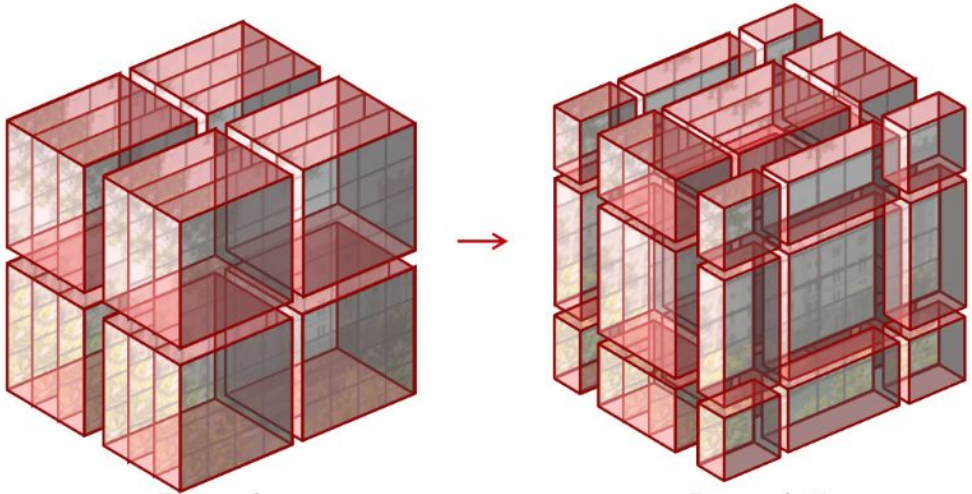
- Shifted windows allow for cross-window connections
  - Maintains computational efficiency





3D Shifted Windows (Cont.)

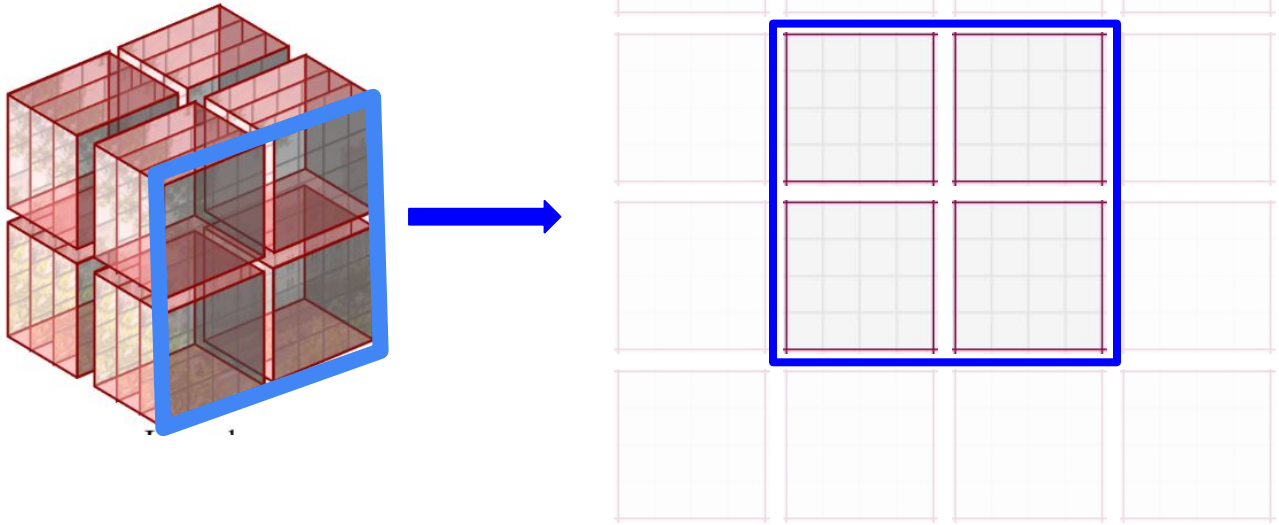
We shift by (P / 2 , M / 2 , M / 2) tokens from that of the preceding layer.



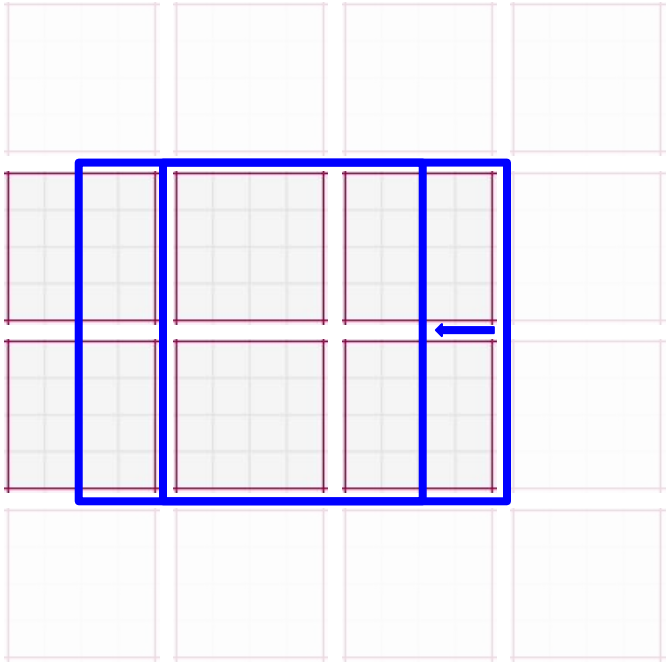
Layer l
window: 2×2×2=8

Layer l+1
window: 3×3×3=27

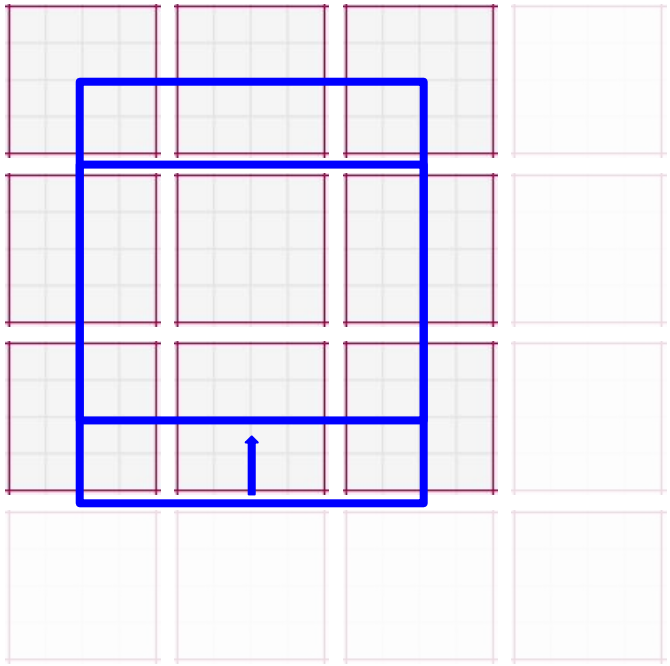
Shift by (P / 2 , M / 2 , M / 2)



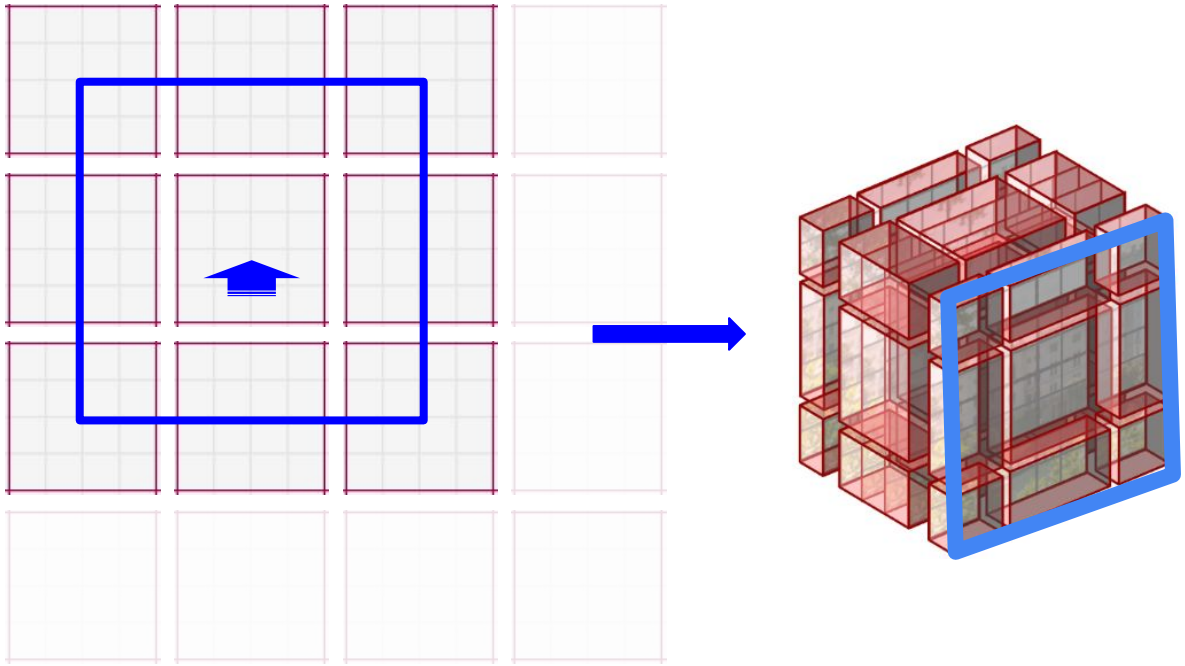
Shift by (P / 2 , **M / 2** , M / 2)



Shift by (P / 2 , M / 2 , **M / 2**)

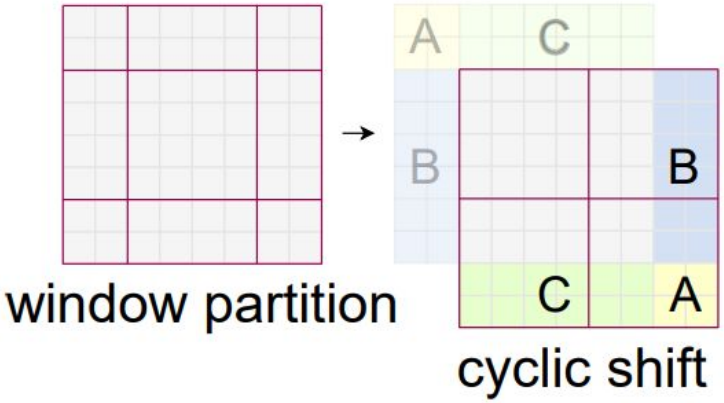


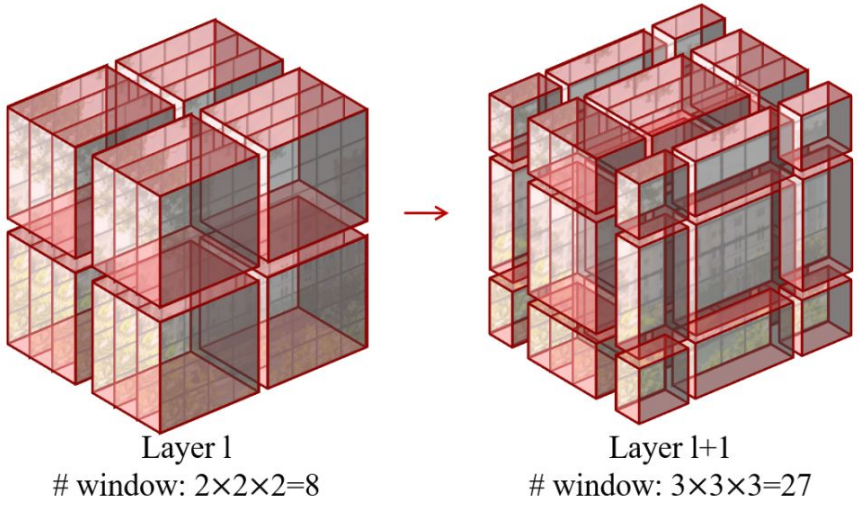
Shift by (**P / 2** , M / 2 , M / 2)

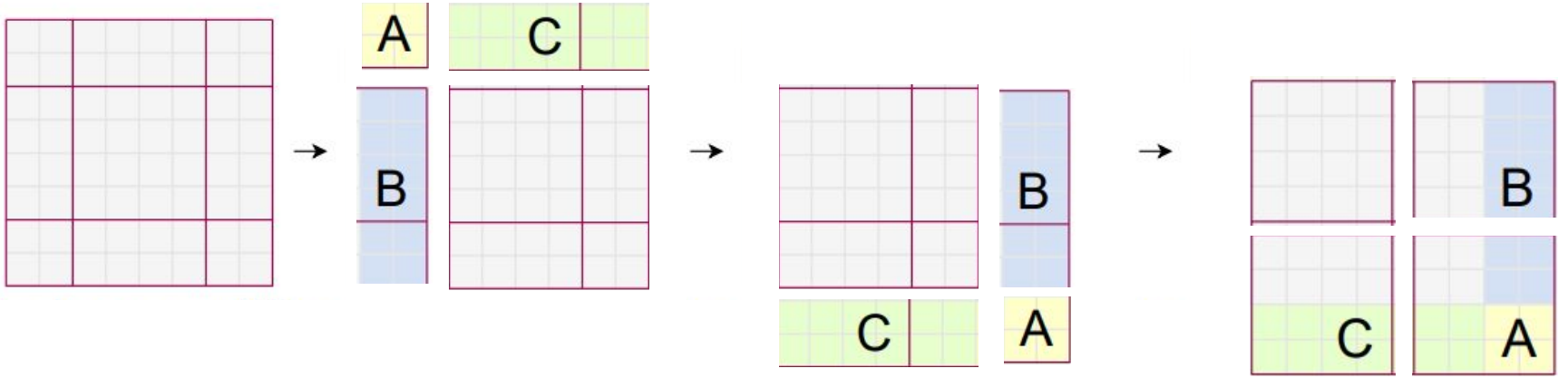


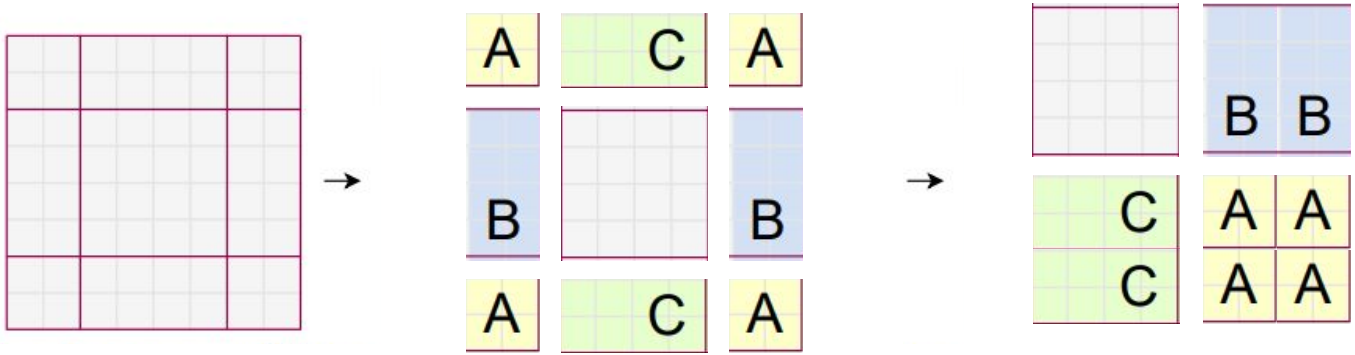
(You will not see the time shift on a 2D space!)

Efficient Batch Computation (Maintained From Swin Transformer)






A
C
B
B
C
A
B
C
A

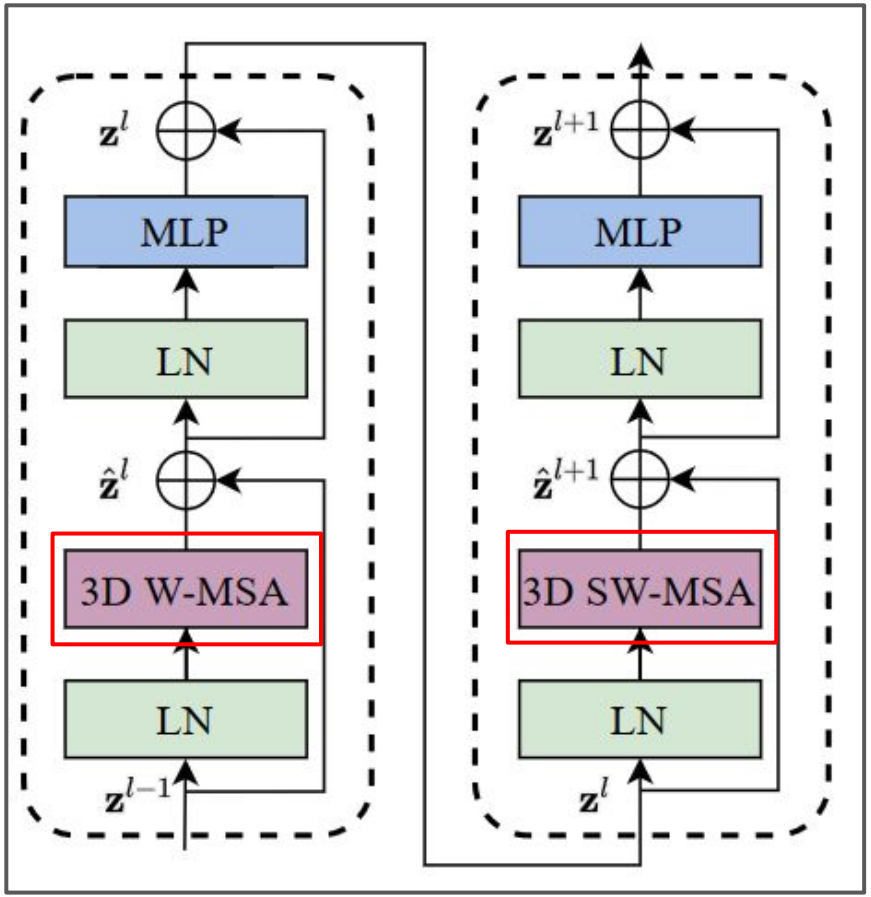

A
C
A
B
B
A
C
A
B B
C
A A
C
A A

3D Shifted Windows (Cont.)

Two consecutive Video Swin Transformer Blocks are computed as:

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

This design introduces connections between neighboring non-overlapping windows.



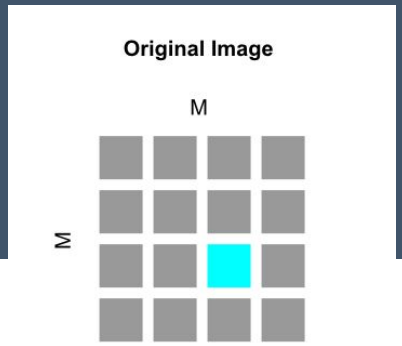
$$\hat{\mathbf{z}}^{l} = \text{3DW-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1},$$
$$\mathbf{z}^{l} = \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l}\right)\right) + \hat{\mathbf{z}}^{l},$$
$$\hat{\mathbf{z}}^{l+1} = \text{3DSW-MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right)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

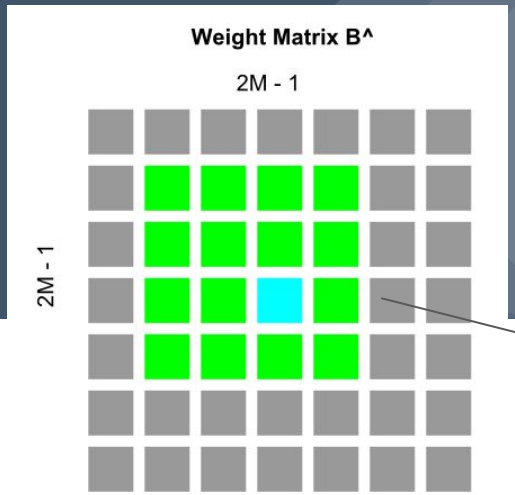
3D Relative Position Bias

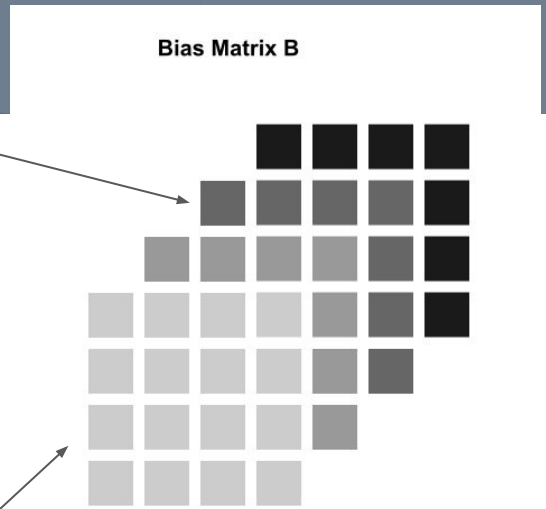
$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V,$$

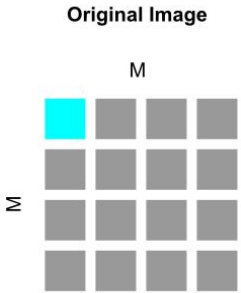
Matrix B: Size $(PM^2)^2$ → Matrix B^: Size $(2P-1)\times(2M-1)^2$

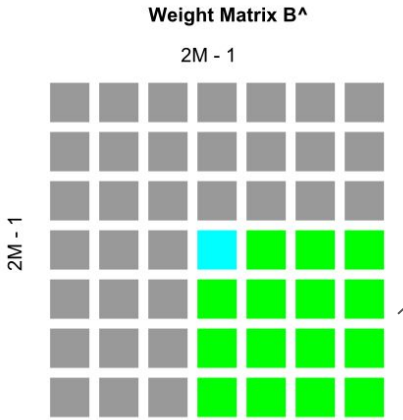
- PM^2 = tokens in a window


Original Image
M
M


Weight Matrix B^
2M - 1
2M - 1


Bias Matrix B


Original Image
M
M

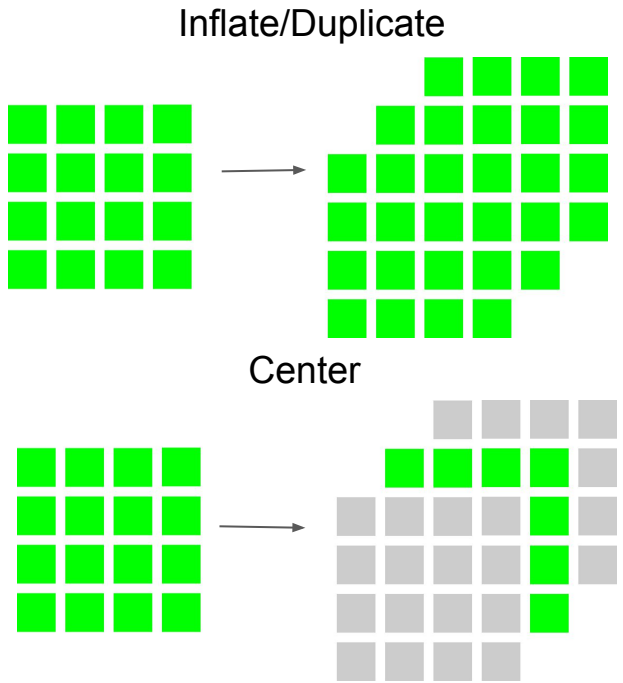

Weight Matrix B^
2M - 1
2M - 1

Initialization from Pre-trained Model

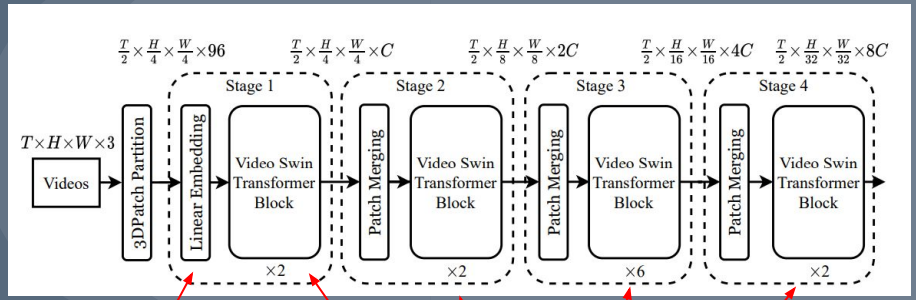
Mismatched Component	Swin 2D	Swin 3D	Solution
Linear Embedding Layer	48xC	96xC	Duplicate original matrix values, then rescale by 0.5 to fix mean and variance
Relative Position Bias Matrix	(2M-1, 2M-1)	(2P-1, 2M-1, 2M-1)	Duplicate original spatial matrix values along temporal dimension.

Ablation on Pretrained Initialization Methods

- Linear Embedding Layer
  - Inflate = average values across time
  - Center = copy values to middle frame
  - No significant change
- Relative Position Bias
  - Duplicate = copy values across time
  - Center* = copy values to middle frame, mask rest
  - No change at all



Architecture Variants



Model Type	Scale	Linear Embedding Dimension	Transformers per stage
Swin-T	0.25x	96	2, 2, 6, 2
Swin-S	0.5x	96	2, 2, 18, 2
Swin-B	1.0x	128	2, 2, 18, 2
Swin-L	1.5x	192	2, 2, 18, 2

Experiments

Datasets

- Kinetics-400
- Kinetics-600
- Something-Something v2

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
MViT-B, 32×3 [10]	-	80.2	94.4	1 × 5	170	36.6
MViT-B, 64×3 [10]	-	81.2	95.1	3 × 3	455	36.6
TimeSformer-L [3]	ImageNet-21K	80.7	94.7	1 × 3	2380	121.4
ViT-B-VTN [29]	ImageNet-21K	78.6	93.7	1 × 1	4218	11.04
ViViT-L/16x2 [1]	ImageNet-21K	80.6	94.7	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	ImageNet-21K	81.3	94.7	4 × 3	3992	310.8
ip-CSN-152 [36]	IG-65M	82.5	95.3	10 × 3	109	32.8
ViViT-L/16x2 [1]	JFT-300M	82.8	95.5	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	JFT-300M	83.5	95.5	4 × 3	3992	310.8
ViViT-H/16x2 [1]	JFT-300M	84.8	95.8	4 × 3	8316	647.5
Swin-T	ImageNet-1K	78.8	93.6	4 × 3	88	28.2
Swin-S	ImageNet-1K	80.6	94.5	4 × 3	166	49.8
Swin-B	ImageNet-1K	80.6	94.6	4 × 3	282	88.1
Swin-B	ImageNet-21K	82.7	95.5	4 × 3	282	88.1
Swin-L	ImageNet-21K	83.1	95.9	4 × 3	604	197.0
Swin-L (384↑)	ImageNet-21K	84.6	96.5	4 × 3	2107	200.0
Swin-L (384↑)	ImageNet-21K	**84.9**	**96.7**	10 × 5	2107	200.0

*Comparison on K400; Some entries (mostly CNN-based) have been removed from table

Kinetics-600

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
SlowFast R101+NL [13]	-	81.8	95.1	10 × 3	234	59.9
X3D-XL [12]	-	81.9	95.5	10 × 3	48	11.0
MViT-B-24, 32×3 [9]	-	83.8	96.3	5 × 1	236	52.9
TimeSformer-HR [3]	ImageNet-21K	82.4	96	1 × 3	1703	121.4
ViViT-L/16x2 320 [1]	ImageNet-21K	83.0	95.7	4 × 3	3992	310.8
ViViT-H/16x2 [9]	JFT-300M	85.8	96.5	4 × 3	8316	647.5
Swin-B	ImageNet-21K	84.0	96.5	4 × 3	282	88.1
Swin-L (384↑)	ImageNet-21K	85.9	97.1	4 × 3	2107	200.0
Swin-L (384↑)	ImageNet-21K	**86.1**	**97.3**	10 × 5	2107	200.0

*FLOPs are in gigaFLOPs

Something-Something-v2

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
TimeSformer-HR [3]	ImageNet-21K	62.5	-	1 × 3	1703	121.4
SlowFast R101, 8×8 [13]	Kinetics-400	63.1	87.6	1 × 3	106	53.3
TSM-RGB [27]	Kinetics-400	63.3	88.2	2 × 3	62	42.9
MSNet [23]	ImageNet-21K	64.7	89.4	1 × 1	67	24.6
TEA [26]	ImageNet-21K	65.1	89.9	10 × 3	70	-
blVNet [11]	SSv2	65.2	90.3	1 × 1	129	40.2
ViViT-L/16x2 [1]	-	65.4	89.8	-	903	352.1
MViT-B, 64×3 [10]	Kinetics-400	67.7	90.9	1 × 3	455	36.6
MViT-B-24, 32×3 [10]	Kinetics-600	68.7	91.5	1 × 3	236	53.2
Swin-B	Kinetics-400	**69.6**	**92.7**	1 × 3	321	88.8

Ablation Study:
Attention with Swin-T on K400

	Top-1	Top-5	FLOPs	Param
Default: joint	78.8	93.6	88	28.2
ViViT: split	76.4	92.1	83	42.0
TimeSformer: factorized	78.5	93.5	95	36.5

- Not running into memory constraints due to shifted windows
- Split model has less efficient temporal modeling
- Factorized approach has to compute temporal and spatial information separately

Ablation Study
Pre-Training and Learning Rate

ratio	Pretrain	Top-1	Top-5
0.1×	ImageNet-1K	80.6	94.6
1.0×	ImageNet-1K	80.2	94.2
0.1×	ImageNet-21K	82.6	95.7
1.0×	ImageNet-21K	82.0	95.3

- By lowering the learning rate of the backbone relative to the head, the model maintains pre-trained parameters longer
- Allows for better generalization
- Future Study: How to better use pre-trained weights?

Conclusion: Advantages

- Smaller model size
- Requires less pre-training data
- Optimization allows for:
  - Joint spatiotemporal self-attention without running out of memory
  - Lower computation costs

Conclusion: Disadvantages

- Requires pre-training
- All ablation studies were on K400
  - Missing potentially interesting evaluation on temporally-intensive dataset
- Not much novelty
  - Swin applied to videos

References

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

Liu, Ze, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. "Video swin transformer." arXiv preprint arXiv:2106.13230 (2021).

Both from Microsoft Research Asia