

Is Space-Time Attention All You Need for Video Understanding?

Overview

- Introduction
- Related Works
- Analysis of Self-Attention Schemes
- TimeSformer Model
- Experiments / Results
- Conclusion

TimeSformer: Introduction

- Create scalable video architecture exclusively using transformers
- Build architectures using ViT tokenization

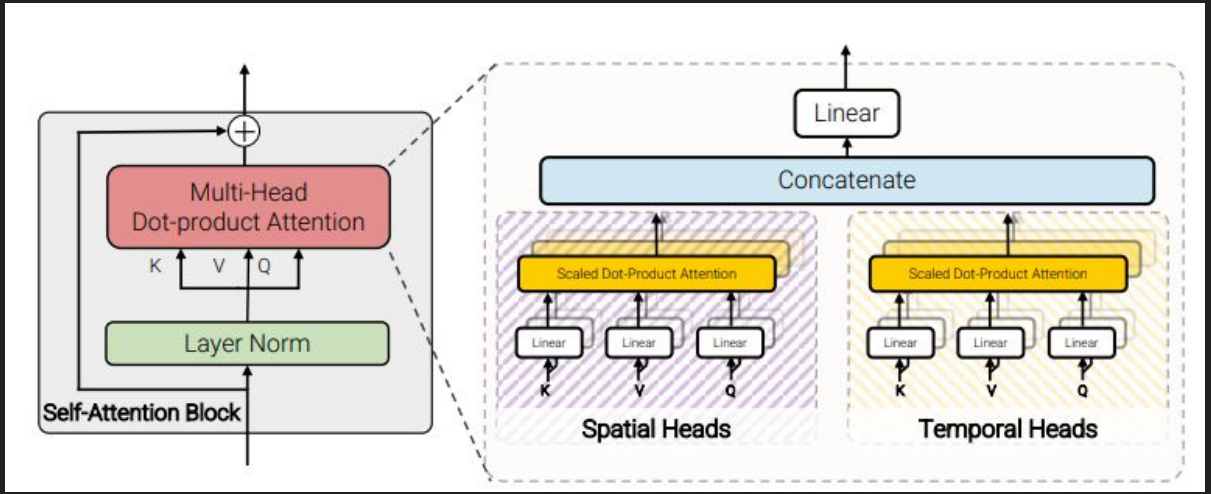
Related Works

- Self-Attention Networks
  - Non-Local Networks
  - 2D Self-Attention Mechanism
  - Hybrid CNN-Transformer networks
  - ViT
- ViViT*

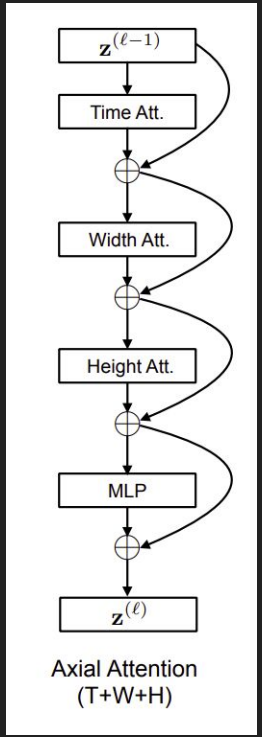
**introduced chronologically after TimeSformer*

Related Works: ViViT

ViViT Factorization



TimeSformer Factorization



Related Works: ViViT

ViViT Tokenization

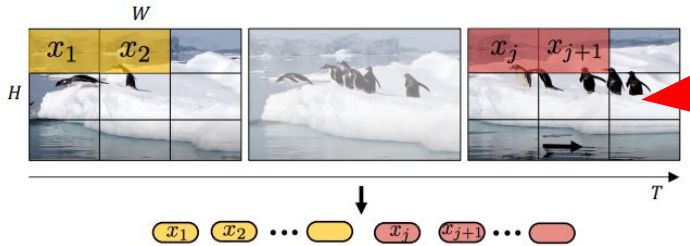


Figure 2: Uniform frame sampling: We simply sample n_t frames, and embed each 2D frame independently following ViT [18].

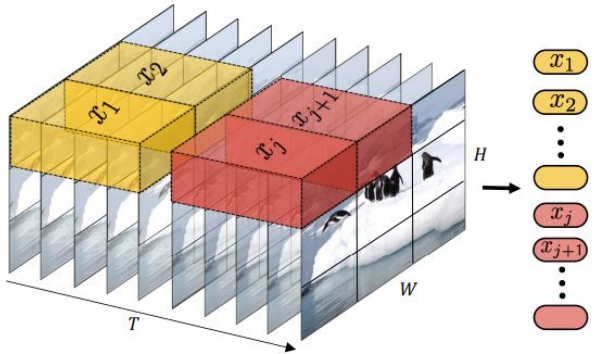
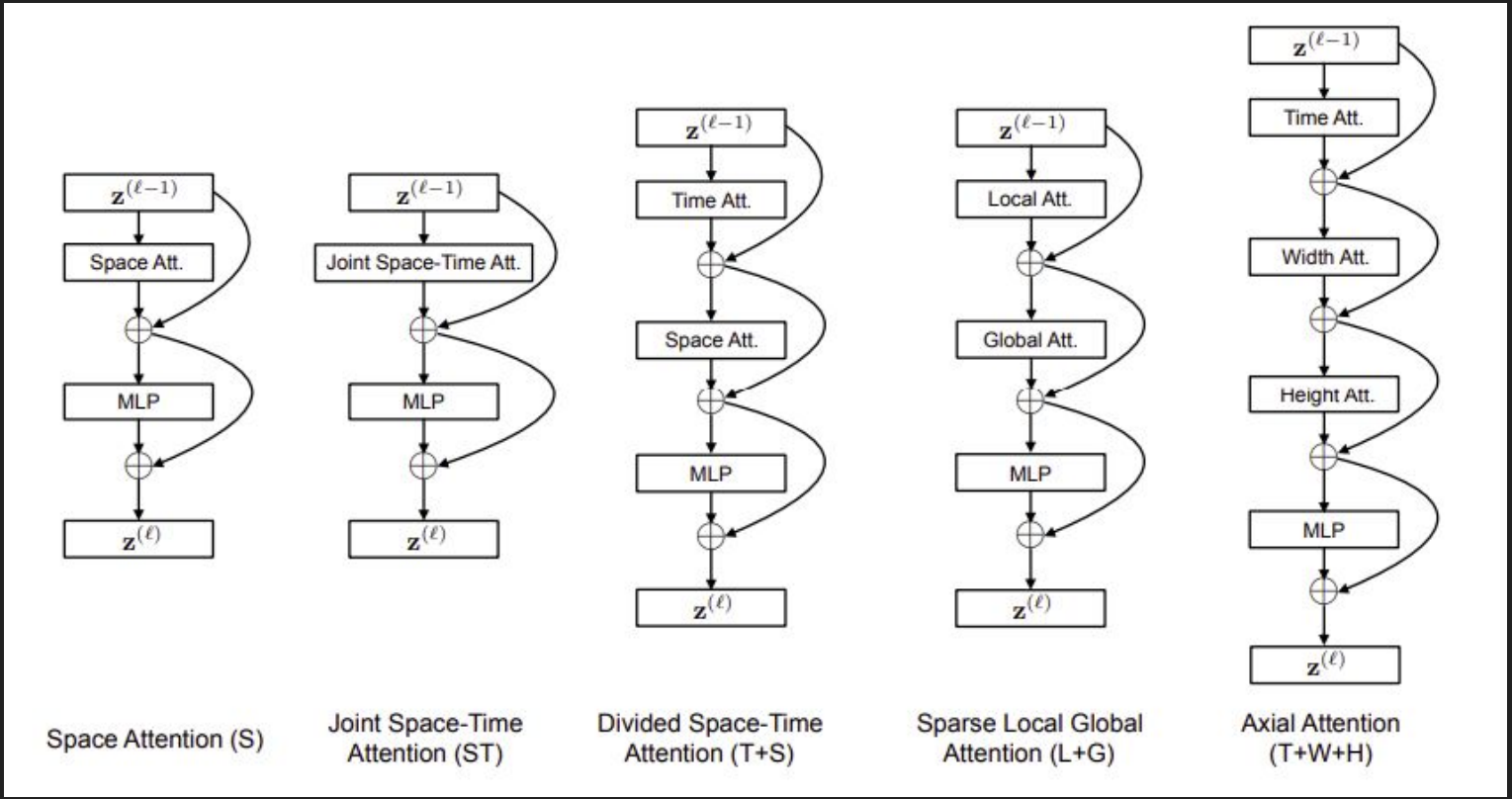
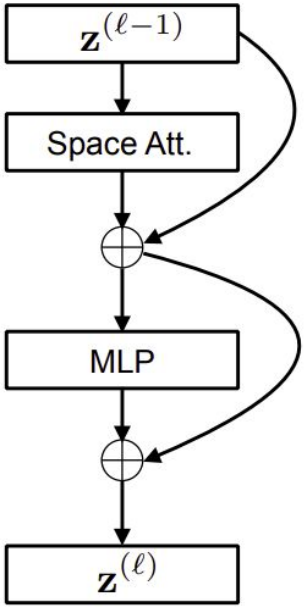


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

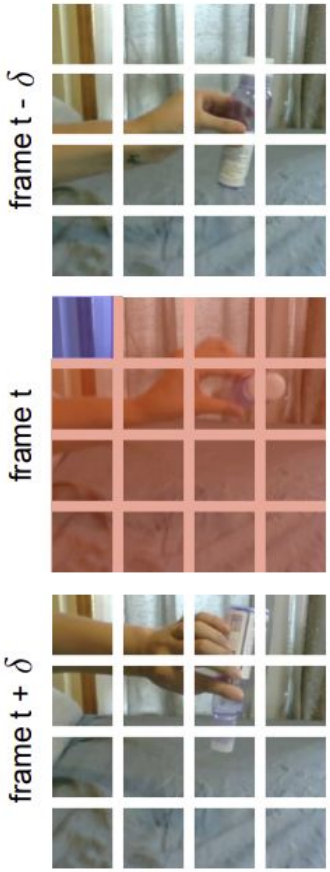
TimeSformer Tokenization

Glance at Space Time Attention Models

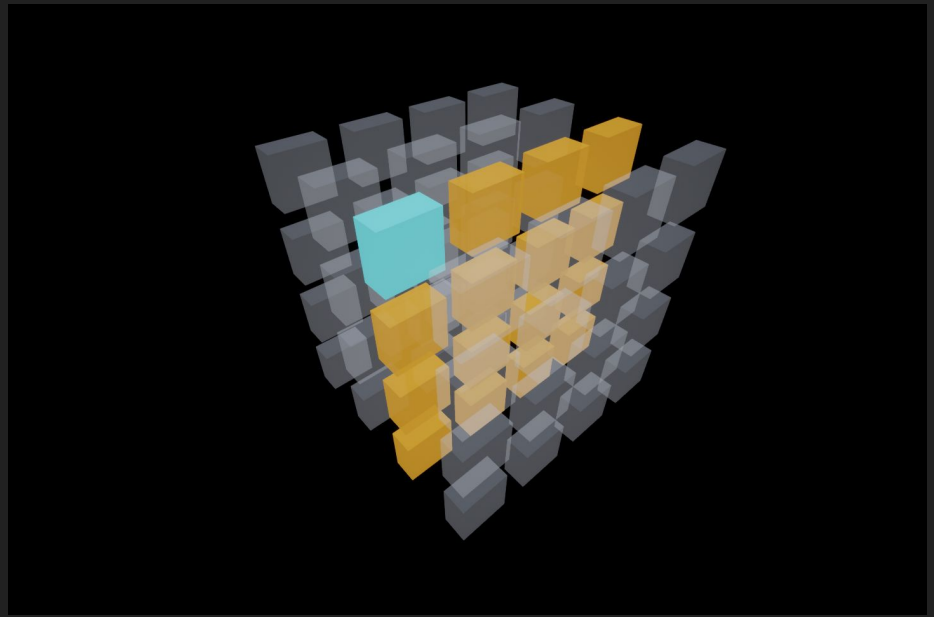


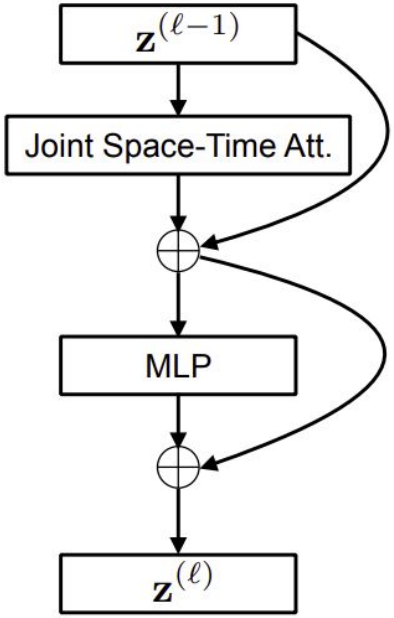


Space Attention (S)

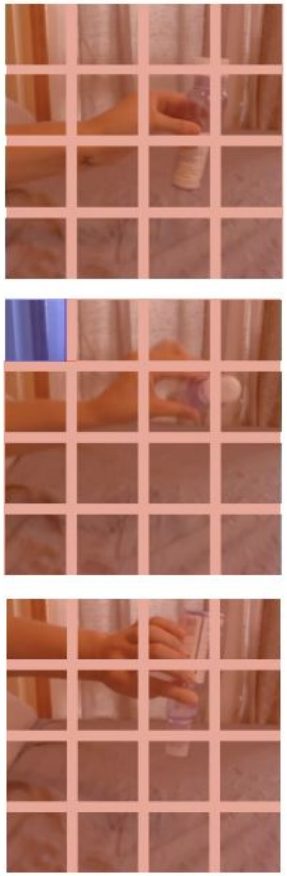


Space Attention (S)

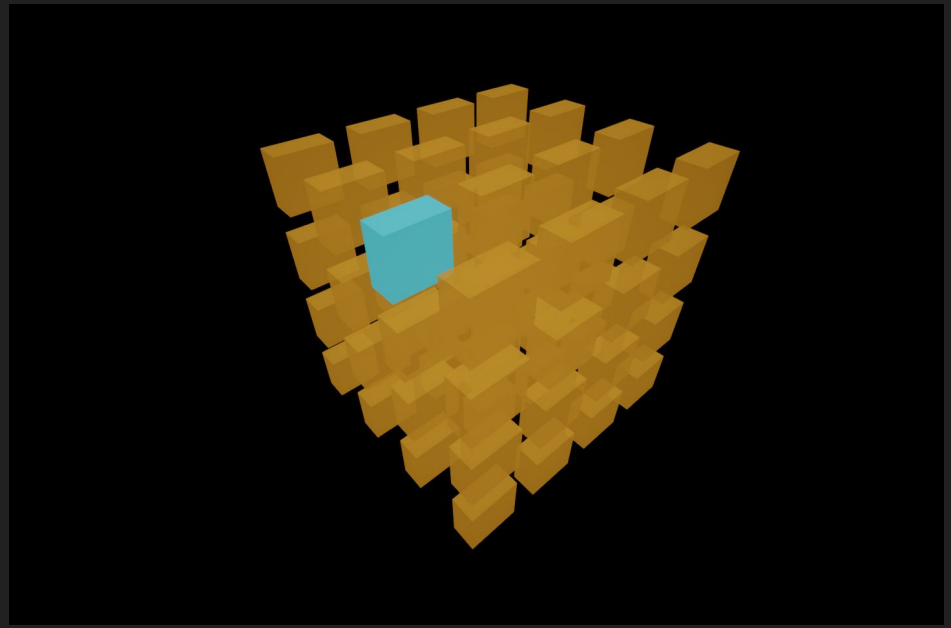


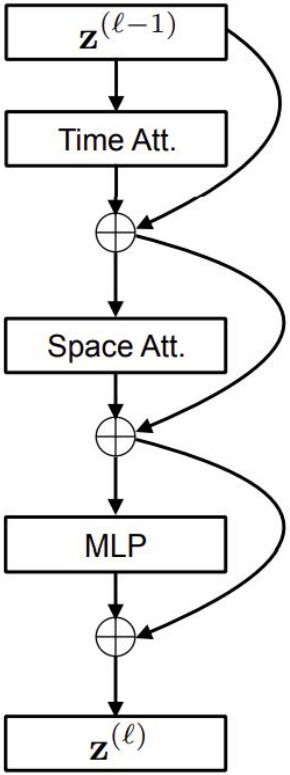


Joint Space-Time Attention (ST)

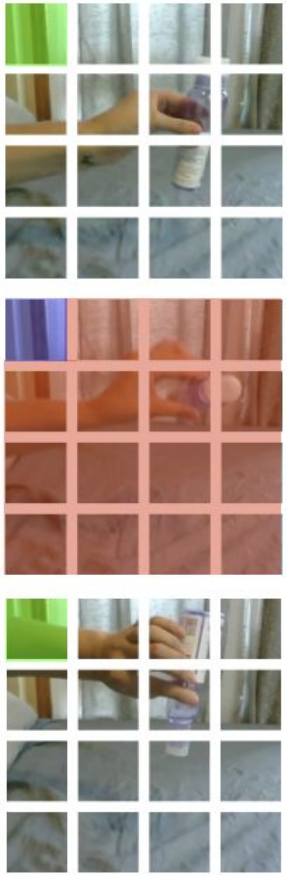


Joint Space-Time Attention (**ST**)

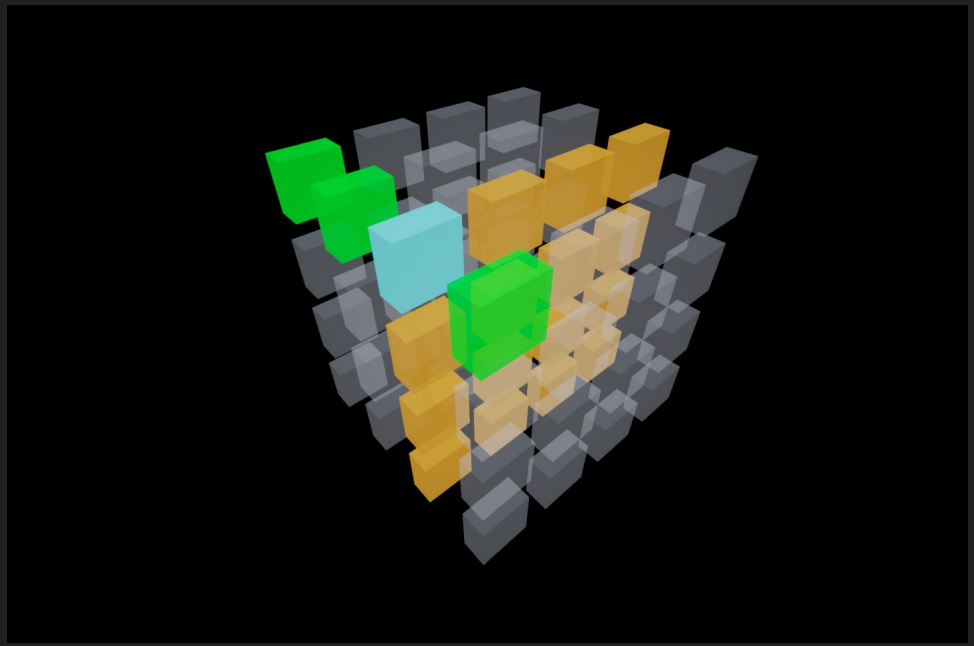


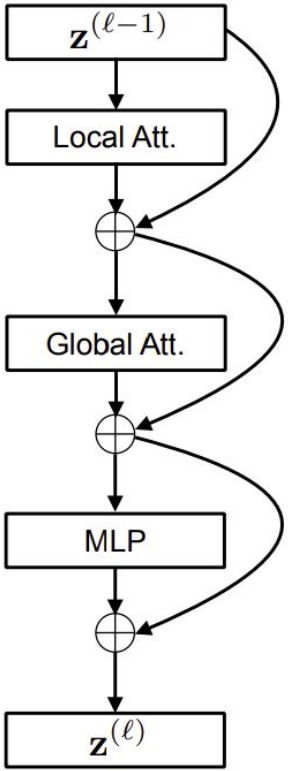


Divided Space-Time Attention (T+S)

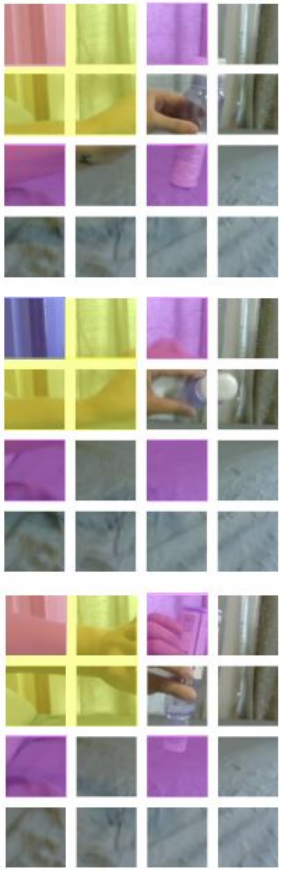


Divided Space-Time Attention (T+S)

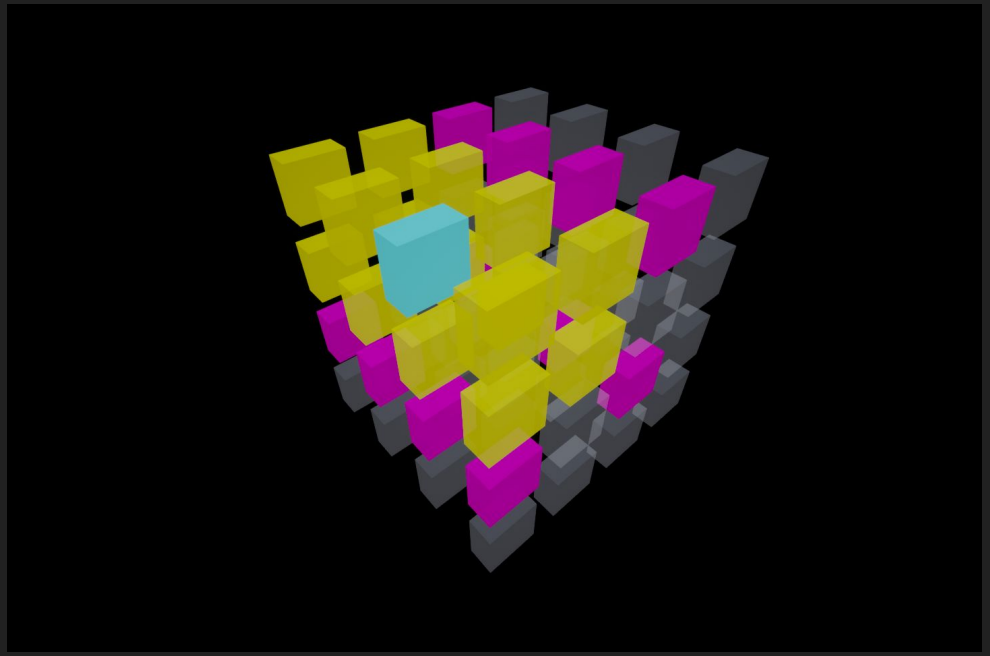


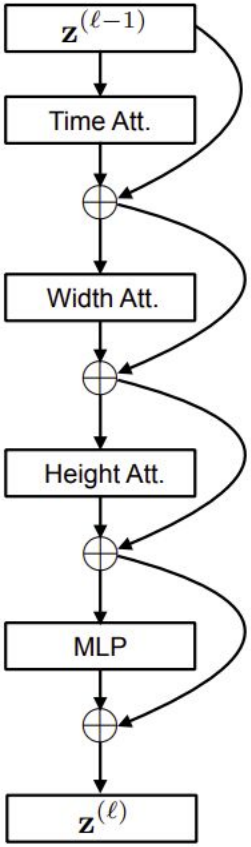


Sparse Local Global Attention (L+G)



Sparse Local Global Attention (L+G)

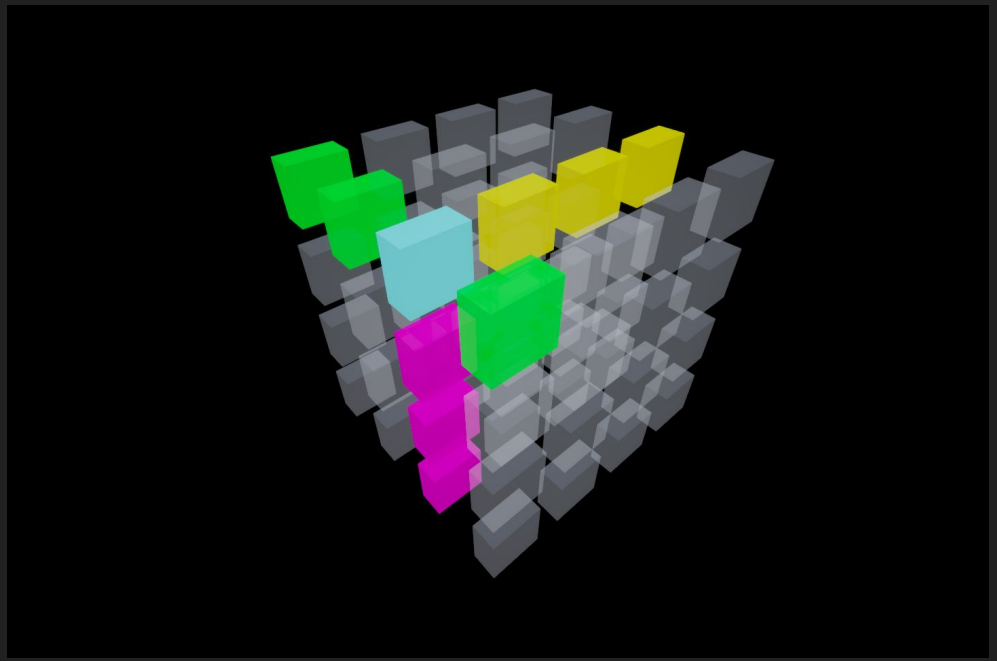




Axial Attention
(T+W+H)



Axial Attention
(T+W+H)



Analysis of Self-Attention Schemes

Attention	Params	K400	SSv2
Space	85.9M	76.9	36.6
Joint Space-Time	85.9M	77.4	58.5
Divided Space-Time	121.4M	**78.0**	**59.5**
Sparse Local Global	121.4M	75.9	56.3
Axial	156.8M	73.5	56.2

“The TimeSformer Model” -- Steps 1 & 2

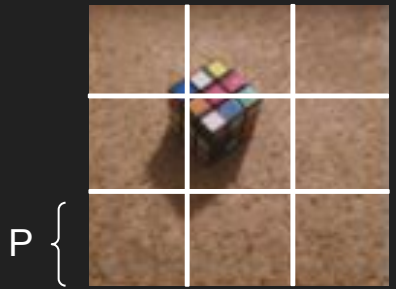
1. Input:
   - Clip of Size H * W * 3 * F
     - **F** RGB frames of size **H * W**



2. Decomposition into Patches:

- Each Frame is divided into N nonoverlapping patches of size P*P
  - Patches span the entire frame N = (H*W) / (P*P)

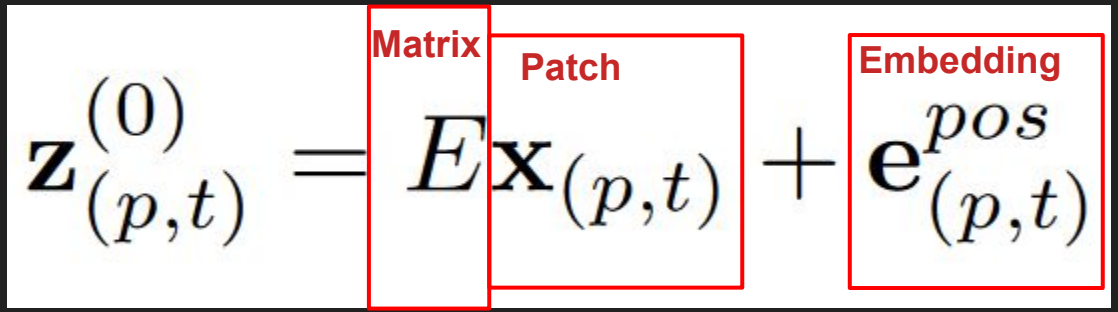
- Patches are flattened to vectors
  - $\mathbf{x}_{(p,t)} \in \mathbf{R}^{3(P*P)}$
    - p is the spatial location
    - t is the frame index



“The TimeSformer Model” -- Step 3

3. Linear Embeddings:

- Each patch ($x_{(p,t)}$) is linearly mapped into an embedding vector, z(0), of size D
  - This is done by means of a learnable matrix E, size $D*3*P^2$



“The TimeSformer Model” -- Step 4

4. Query-Key-Value Computation:

- There are L encoding blocks
  - At each block, ℓ, a query/key/value vector is computed from the representation z(ℓ − 1) encoded by the preceding block, for each patch

a = 1, …, A
Index over multiple attention heads

D_h = D / A
Latent dimensionality for each attent. head

$$\mathbf{q}_{(p,t)}^{(\ell,a)} = W_Q^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h}$$

$$\mathbf{k}_{(p,t)}^{(\ell,a)} = W_K^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h}$$

$$\mathbf{v}_{(p,t)}^{(\ell,a)} = W_V^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h}$$

“The TimeSformer Model” -- Step 5

5.1 Self-Attention Computation:

- Self-Attention Weights are computed via dot product.
  - The following general formula provides the self-attention weights for a query patch (p, t):

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t')}^{(\ell,a)}\right\}_{\substack{p'=1,\ldots,N \\ t'=1,\ldots,F}}\right]\right)$$

“The TimeSformer Model” -- Step 5

5.2 Self-Attention Computation:

- Reducing the attention to one dimension reduces the number of comparisons
  - E.g. Total query-key comparisons shift from NF + 1 previously to N + F + 2.

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)\text{time}} = \text{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p,t')}^{(\ell,a)}\right\}_{t'=1,\ldots,F}\right]\right)$$

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)\text{space}} = \text{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t)}^{(\ell,a)}\right\}_{p'=1,\ldots,N}\right]\right)$$

“The TimeSformer Model” -- Step 6

6. Encoding

At block ℓ the encoding, $z(\ell)$, is obtained by:

1. Computing the weighted sum of value vectors using self-attention coefficients from each attention head.

2. The vectors are then concatenated from all heads
   - Then projected and passed through an MLP
   - Residual connections are used after each operation

“The TimeSformer Model” -- Step 7

7. Classification Embedding

- Obtained from the final classification token
- A 1-hidden-layer MLP is appended to predict final classes of the videos

Experiments / Results

TimeSformer was evaluated on:

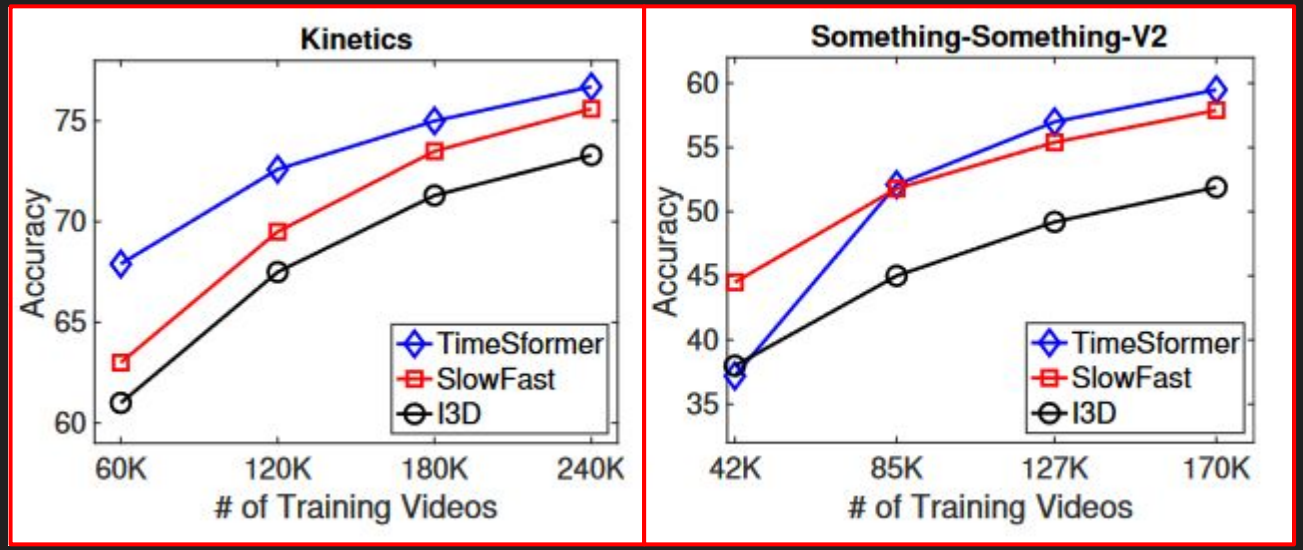
- Kinetics-400 (K400)
  - **spatial cues > temporal cues**
  - 160,000 10-second clips from YouTube videos
- Something-Something-V2 (SSv2)
  - **temporal cues > spatial cues**
  - 220,847 videos of people doing things with objects, 174 classes
- Diving-48
  - **temporal cues > spatial cues**
  - 18,000 fine-grained videos of competitive diving

Comparison to CNNs -- Importance of Pre-training

Model	Pretrain	K400 Training Time (hours)	K400 Acc.	Inference TFLOPs	Params
I3D 8x8 R50	ImageNet-1K	444	71.0	1.11	28.0M
I3D 8x8 R50	ImageNet-1K	1440	73.4	1.11	28.0M
SlowFast R50	ImageNet-1K	448	70.0	1.97	34.6M
SlowFast R50	ImageNet-1K	3840	75.6	1.97	34.6M
SlowFast R50	N/A	6336	76.4	1.97	34.6M
TimeSformer	ImageNet-1K	**416**	75.8	**0.59**	121.4M
TimeSformer	ImageNet-21K	**416**	**78.0**	**0.59**	121.4M

- ImageNet pretraining is on 8 × 224 × 224 video clips
- Authors did not have access to ResNet pretrained on ImageNet-21K

Impact on Video-Data Amount on Performance



- Kinetics-400 & SSv2
- Trained on subsets of the datasets (25%, 50%, 75%, 100%)

Importance of Pre-Training, # of Frames, & Resolution

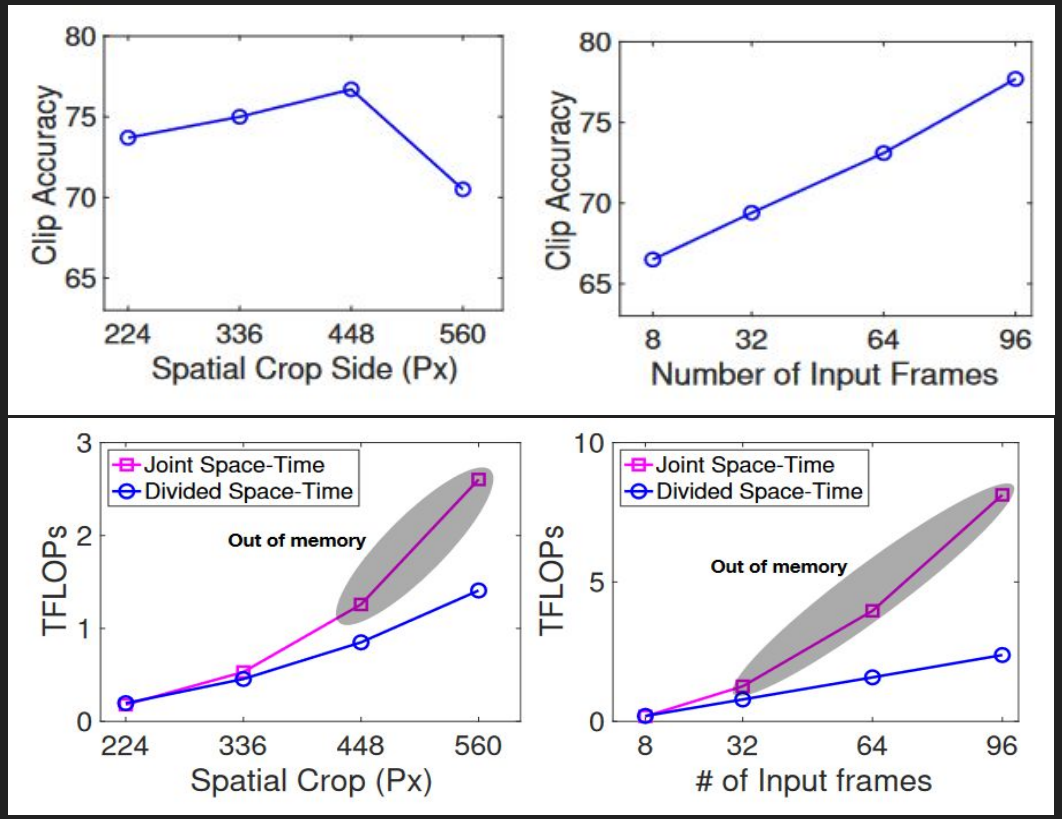
Method	Pretraining	K400	SSv2
TimeSformer	ImageNet-1K	75.8	**59.5**
TimeSformer	ImageNet-21K	**78.0**	**59.5**
TimeSformer-HR	ImageNet-1K	77.8	62.2
TimeSformer-HR	ImageNet-21K	**79.7**	**62.5**
TimeSformer-L	ImageNet-1K	78.1	**62.4**
TimeSformer-L	ImageNet-21K	**80.7**	62.3

TimeSformer: 8 × 224 × 224 Video Clips & Frames Sampled at a Rate of 1/32

TimeSformer-HR: 16 × 448 × 448 Video Clips & Frames Sampled at a Rate of 1/32

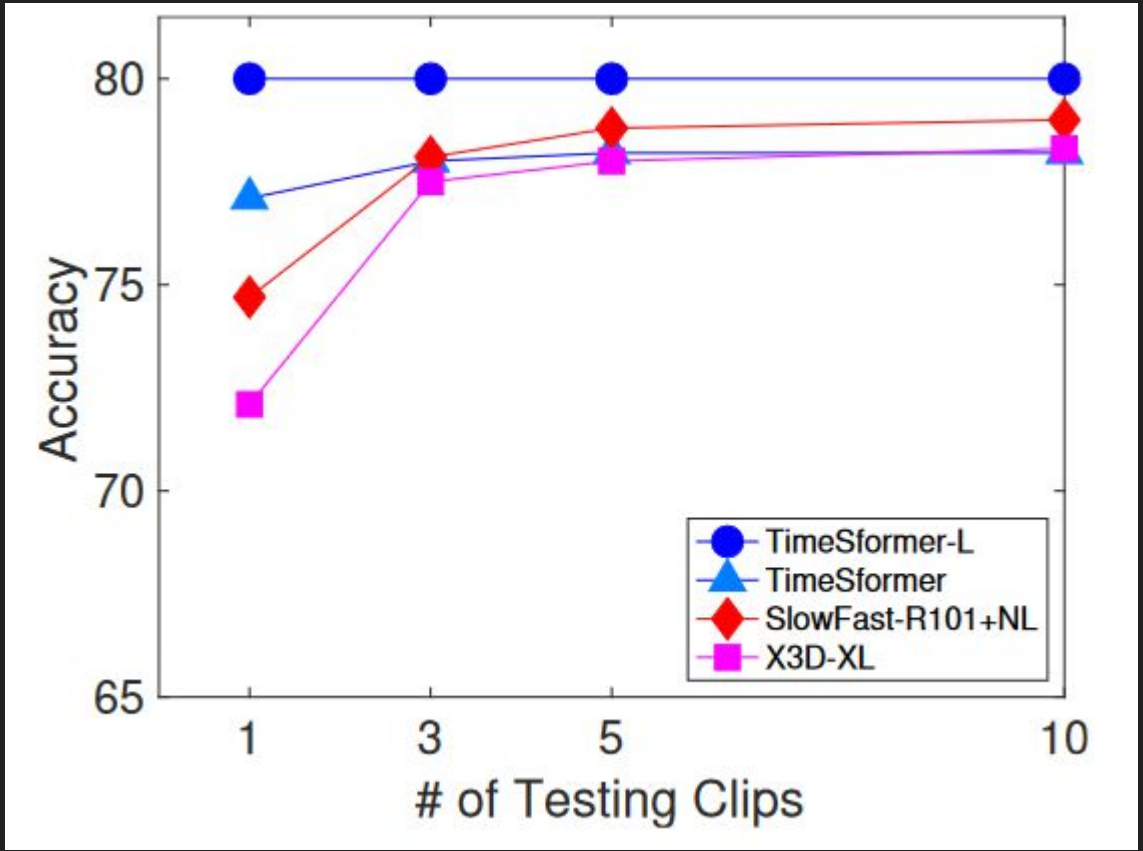
TimeSformer-L: 96 × 224 × 224 Video Clips & Frames Sampled at a Rate of 1/4

Accuracy and Performance Comparisons



Accuracy vs.# of Temporal Clips

- Given one spatial crop, how many clips needed?
- Timesformer-L can handle 12 second clips



Kinetics-400 Comparisons

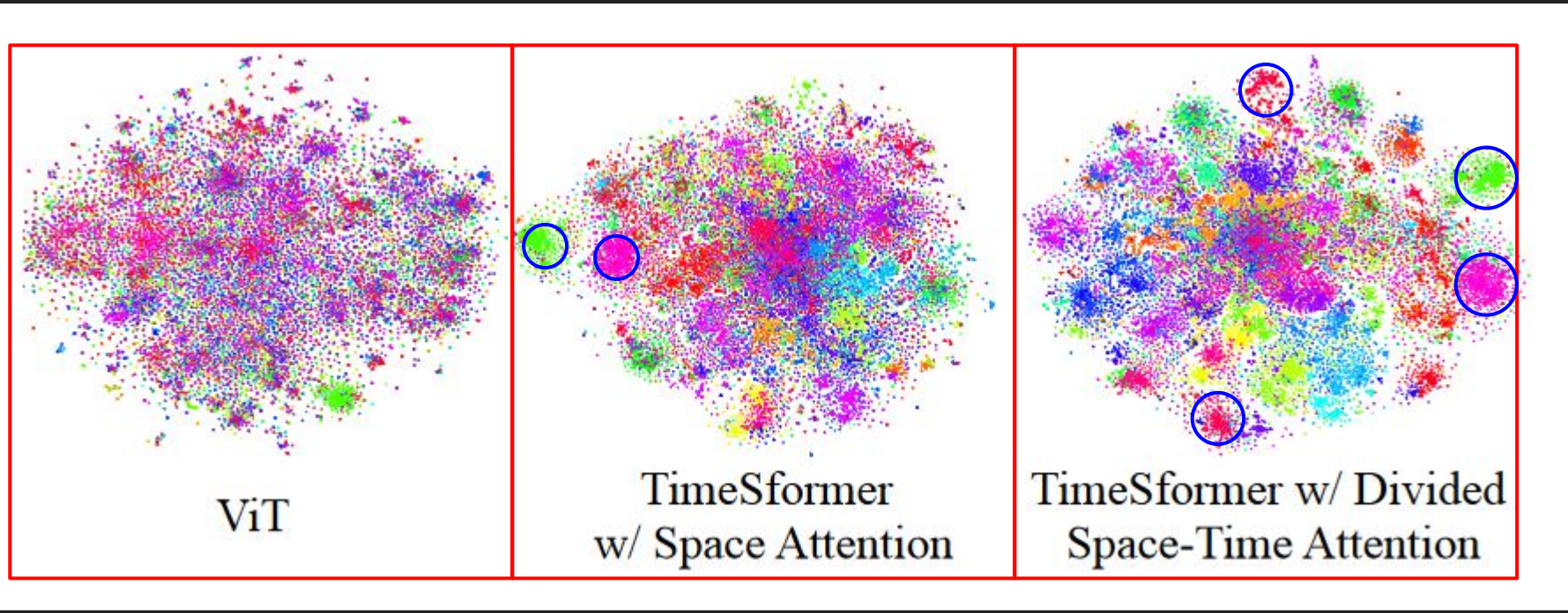
Method	Top-1	Top-5	TFLOPs
R(2+1)D (Tran et al., 2018)	72.0	90.0	17.5
bLVNet (Fan et al., 2019)	73.5	91.2	0.84
TSM (Lin et al., 2019)	74.7	N/A	N/A
S3D-G (Xie et al., 2018)	74.7	93.4	N/A
Oct-I3D+NL (Chen et al., 2019)	75.7	N/A	0.84
D3D (Stroud et al., 2020)	75.9	N/A	N/A
I3D+NL (Wang et al., 2018b)	77.7	93.3	10.8
ip-CSN-152 (Tran et al., 2019)	77.8	92.8	3.2
CorrNet (Wang et al., 2020a)	79.2	N/A	6.7
LGD-3D-101 (Qiu et al., 2019)	79.4	94.4	N/A
SlowFast (Feichtenhofer et al., 2019b)	79.8	93.9	7.0
X3D-XXL (Feichtenhofer, 2020)	80.4	94.6	5.8
TimeSformer	78.0	93.7	**0.59**
TimeSformer-HR	79.7	94.4	5.11
TimeSformer-L	**80.7**	**94.7**	7.14

SSv2 & Diving-48 Comparison

Method	SSv2	Diving-48**
SlowFast (Feichtenhofer et al., 2019b)	61.7	77.6
TSM (Lin et al., 2019)	63.4	N/A
STM (Jiang et al., 2019)	64.2	N/A
MSNet (Kwon et al., 2020)	64.7	N/A
TEA (Li et al., 2020b)	65.1	N/A
bLVNet (Fan et al., 2019)	**65.2**	N/A
TimeSformer	59.5	74.9
TimeSformer-HR	62.2	78.0
TimeSformer-L	62.4	**81.0**

t-SNE Comparison on SSv2

- More Separable Features w/ Divided Space-Time Attention



Overview: HowTo100M

- Long-Term Video Modeling Dataset
  - 1M instructional videos & 23k task categories
  - Average video duration of 7 minutes
- Selected only tasks with at least 100 videos
  - 120K videos & 1059 task categories
    - Randomly Selected: 85k Training, 35k Test

Comparison on HowTo100M

Method	# Input Frames	Single Clip Coverage	# Test Clips	Top-1 Acc
SlowFast	8	8.5s	48	48.2
SlowFast	32	34.1s	12	50.8
SlowFast	64	68.3s	6	51.5
SlowFast	96	102.4s	4	51.2
TimeSformer	8	8.5s	48	56.8
TimeSformer	32	34.1s	12	61.2
TimeSformer	64	68.3s	6	62.2
TimeSformer	96	102.4s	4	**62.6**

- Pre-Trained on Kinetics-400, Fine-tuned on HowTo100M
- All Sampled at a frame rate of 1/32

Advantages

- Reported accuracy is comparable/better than SOTA CNNs
- More efficient than comparable methods (less TFLOPS)
- Higher number of learnable parameters than SOTA methods
- Able to better capture time dependencies than 3D convolutions
    - Captures longer dependencies
    - Not reliant on composing smaller actions into larger actions like a CNN

Disadvantages

- Requires pre-training or a large dataset
  - Like other transformer networks
- Attention mechanism is limited in time
  - To cover longer clips, e.g. 7 minutes, clips must be sampled and have results averaged

ViViT Comparison to TimeSformer

- ViViT outperforms TimeSformer-L on the Kinetic-400 dataset by 1% but requires 1.67x more TFLOPs for same number of spatial views

Method	Top 1	Top 5	Views	TFLOPs
blVNet [19]	73.5	91.2	–	–
STM [33]	73.7	91.6	–	–
TEA [42]	76.1	92.5	10 × 3	2.10
TSM-ResNeXt-101 [43]	76.3	–	–	–
I3D NL [75]	77.7	93.3	10 × 3	10.77
CorrNet-101 [70]	79.2	–	10 × 3	6.72
ip-CSN-152 [66]	79.2	93.8	10 × 3	3.27
LGD-3D R101 [51]	79.4	94.4	–	–
SlowFast R101-NL [21]	79.8	93.9	10 × 3	7.02
X3D-XXL [20]	80.4	94.6	10 × 3	5.82
TimeSformer-L [4]	80.7	**94.7**	1 × 3	7.14
ViViT-L/16x2 FE	80.6	92.7	1 × 1	3.98
ViViT-L/16x2 FE	**81.7**	93.8	1 × 3	11.94

References

Bertasius, Gedas, Heng Wang, and Lorenzo Torresani. "Is space-time attention all you need for video understanding." arXiv preprint arXiv:2102.05095 2, no. 3 (2021): 4.

Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers." In European conference on computer vision, pp. 213-229. Springer, Cham, 2020.

Hu, Han, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. "Relation networks for object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3588-3597. 2018.