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

Gedas Bertasius, Heng Wang, Lorenzo, Torresani (Facebook)

Reviewed by Saehee Jeon

Abstract

“Convolution-free” Video classification

1. SOTA results on several action recognition benchmarks
2. Faster than other convolutional networks
3. Can be applied to much longer video clips

Introduction

Self-attention based methods in NLP

👉 Revolution (long-range dependencies among words)

Video understanding & NLP

In common :

1. both sequential
2. contextualized with the rest of the video/words in order to be fully disambiguated

Introduction

2D or 3D convolutions still represent the core operators!

In this work,

Performant **convolution-free** video architecture 🌟

(replacing the convolution operator with self-attention)

👉🏻 Overcome a few limitations of convolutional models for video analysis

Introduction

<A few **limitations** of **convolutional** models for video analysis>

1. **Strong inductive biases**(good for small data but limits the expressivity)

e.g. local connectivity and translation equivariance

👉 Transformers impose less restrictive inductive biases

2. Capture **short-range** spatiotemporal information

👉 cannot model dependencies that extend beyond the receptive field

🌟 **Self-attention** mechanism can be applied to capture both local/global long-range dependencies

Introduction

<A few **limitations** of **convolutional** models for video analysis>

3. Cost

Training deep CNN remains very costly

(especially when applied to high-resolution and long videos)

✌️ Transformers enjoy faster training and inference compared to CNNs

So...👀

Adapt the image model *Vision Transformer(ViT)* to video

(Extend the **self-attention** mechanism from the image space to the space-time 3D volume)

“TimeSformer”

Each patch is linearly mapped into an **embedding** and augmented with **positional information**.

👉🏻 Interpret the resulting sequence of vectors as token embeddings which can be fed to a Transformer encoder.

Wait... ☝️

Computing cost?⏱️💸

Transformer requires computing a similarity measure for all pairs of tokens.

👉 Computationally costly due to the large number of patches.😂

👉 ***“Divided attention”***

Separately applies temporal attention and spatial attention within each block of the network.

😀 SOTA accuaracy & can be used for long-range modeling of videos.

The TimeSformer Model

Input clip

$$X \in \mathbb{R}^{H \times W \times 3 \times F}$$

(F RGB frames of size HXW sampled from the original video)

Decomposition into patches

$$N = HW/P^2$$

Decompose each frame into N non-overlapping patches(each of size PXP)

& Flatten these patches into vectors $\mathbf{x}_{(p,t)} \in \mathbb{R}^{3P^2}$

$p = 1, \ldots, N$ denoting spatial locations and $t = 1, \ldots, F$ depicting an index over frames.

The TimeSformer Model

Linear embedding

Linearly map each patch $\mathbf{x}_{(p,t)}$ into an embedding vector $\mathbf{z}^{(0)}_{(p,t)} \in \mathbb{R}^{D}$

by means of a learnable matrix $E \in \mathbb{R}^{D \times 3P^2}$

Input to the Transformer

$$\mathbf{z}^{(0)}_{(p,t)} = E\mathbf{x}_{(p,t)} + \mathbf{e}^{pos}_{(p,t)}$$

where $\mathbf{e}^{pos}_{(p,t)} \in \mathbb{R}^{D}$ represents a learnable positional embedding

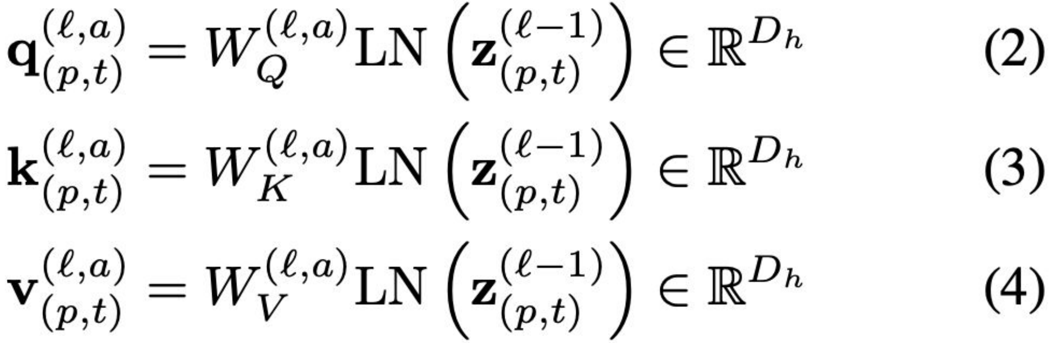
(added to encode the spatiotemporal position of each patch)

The TimeSformer Model

Query-Key-Value computation

Transformer consists of L encoding blocks.

At each block l,



$$\mathbf{q}_{(p,t)}^{(\ell,a)} = W_Q^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h} \quad (2)$$

$$\mathbf{k}_{(p,t)}^{(\ell,a)} = W_K^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h} \quad (3)$$

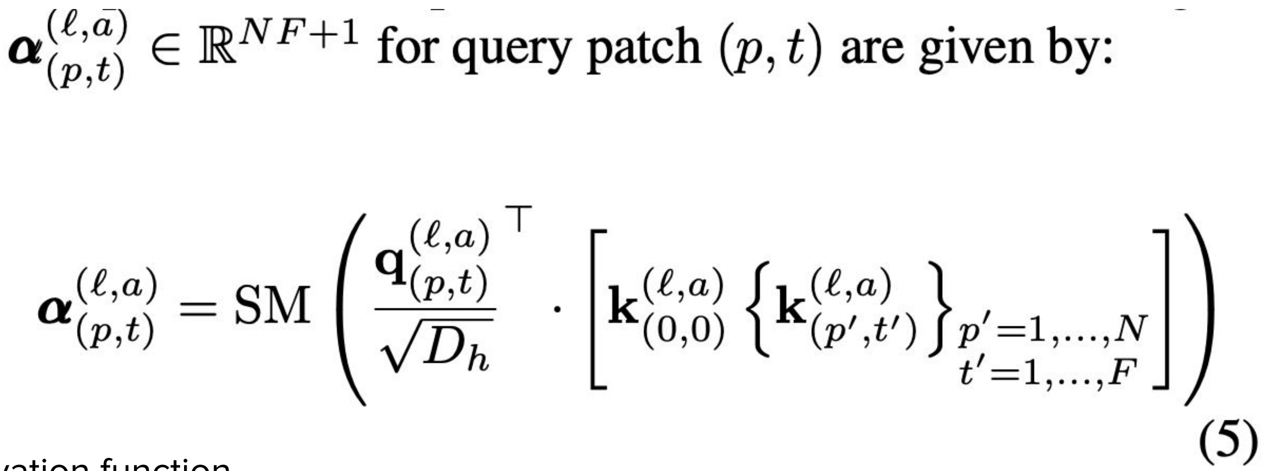
$$\mathbf{v}_{(p,t)}^{(\ell,a)} = W_V^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h} \quad (4)$$

LN : LayerNorm

The TimeSformer Model

Self-attention computation

Self-attention weights are computed via dot-product



$\boldsymbol{\alpha}^{(\ell,a)}_{(p,t)} \in \mathbb{R}^{NF+1}$ for query patch (p,t) are given by:

$$\boldsymbol{\alpha}^{(\ell,a)}_{(p,t)} = \mathrm{SM}\left(\frac{\mathbf{q}^{(\ell,a)}_{(p,t)}}{\sqrt{D_h}}^{\top} \cdot \left[\mathbf{k}^{(\ell,a)}_{(0,0)} \left\{\mathbf{k}^{(\ell,a)}_{(p',t')}\right\}_{\substack{p'=1,\dots,N \\ t'=1,\dots,F}}\right]\right) \quad (5)$$

SM : softmax activation function

The TimeSformer Model

Encoding

The encoding $\mathbf{z}_{(p,t)}^{(\ell)}$ at block ℓ obtained by first computing the weighted sum of value vectors using self-attention coefficients from each attention head.

$$\mathbf{s}_{(p,t)}^{(\ell,a)} = \alpha_{(p,t),(0,0)}^{(\ell,a)} \mathbf{v}_{(0,0)}^{(\ell,a)} + \sum_{p'=1}^{N} \sum_{t'=1}^{F} \alpha_{(p,t),(p',t')}^{(\ell,a)} \mathbf{v}_{(p',t')}^{(\ell,a)}. \tag{7}$$

The TimeSformer Model

Then, the **concatenation** of these vectors from all head is projected and passed through an **MLP** (using residual connections)

$$\mathbf{z'}^{(\ell)}_{(p,t)} = W_O \begin{bmatrix} \mathbf{s}^{(\ell,1)}_{(p,t)} \\ \vdots \\ \mathbf{s}^{(\ell,\mathcal{A})}_{(p,t)} \end{bmatrix} + \mathbf{z}^{(\ell-1)}_{(p,t)} \tag{8}$$

$$\mathbf{z}^{(\ell)}_{(p,t)} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z'}^{(\ell)}_{(p,t)}\right)\right) + \mathbf{z'}^{(\ell)}_{(p,t)}. \tag{9}$$

The TimeSformer Model

Classification embedding

Final clip embedding is obtained from the final block for the classification token

$$\mathbf{y} = \mathrm{LN}\left(\mathbf{z}_{(0,0)}^{(L)}\right) \in \mathbb{R}^{D}. \qquad (10)$$

On top of this representation, we append **a 1-hidden-layer MLP**
to predict the final video classes.

The TimeSformer Model

Space-Time Self-Attention Models

We can reduce the computational cost by replacing the spatiotemporal attention

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)}}{\sqrt{D_h}}^{\top} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t')}^{(\ell,a)}\right\}_{\substack{p'=1,\ldots,N \\ t'=1,\ldots,F}}\right]\right) \tag{5}$$

→

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)\mathrm{space}} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)}}{\sqrt{D_h}}^{\top} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t)}^{(\ell,a)}\right\}_{p'=1,\ldots,N}\right]\right) \tag{6}$$

But, this model neglects to capture temporal dependencies across frames.
(leads to degraded classification accuracy compared to full spatiotemporal attention)

The TimeSformer Model

Space-Time Self-Attention Models

👉**“Divided Space-Time Attention”(T+S)**

🌟 **Temporal attention** and **spatial attention** are separately applied one after the other.

Within each block l, first computer **temporal attention** by comparing each patch (p,t)

with all the patches at the same spatial location in the other frames.

$$\boldsymbol{\alpha}^{(\ell,a)\mathrm{time}}_{(p,t)} = \mathrm{SM}\left(\frac{\mathbf{q}^{(\ell,a)}_{(p,t)}}{\sqrt{D_h}}^{\top} \cdot \left[\mathbf{k}^{(\ell,a)}_{(0,0)} \left\{ \mathbf{k}^{(\ell,a)}_{(p,t')} \right\}_{t'=1,\ldots,F} \right] \right). \tag{11}$$

The TimeSformer Model

Space-Time Self-Attention Models

$$\mathbf{z}'^{(\ell)}_{(p,t)} = W_O \begin{bmatrix} \mathbf{s}^{(\ell,1)}_{(p,t)} \\ \vdots \\ \mathbf{s}^{(\ell,\mathcal{A})}_{(p,t)} \end{bmatrix} + \mathbf{z}^{(\ell-1)}_{(p,t)} \tag{8}$$

The encoding in (8) is the fed back for **spatial attention** computation instead of being passed to the MLP.

👉 $\mathbf{z}'^{(\ell)\text{space}}_{(p,t)}$ is passed to the MLP of (9)

$$\mathbf{z}^{(\ell)}_{(p,t)} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'^{(\ell)}_{(p,t)}\right)\right) + \mathbf{z}'^{(\ell)}_{(p,t)}. \tag{9}$$

Experiments

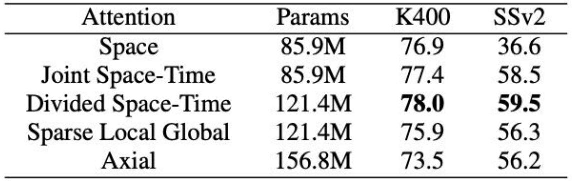
4 popular action recognition **datasets** :
Kinetics-400, Kinetics-600, Something-Something-V2 and Diving-48

Use clips of size **8X224X224** with frames sampled at a rate of **1/32**
patch size : **16X16**

Use 3 spatial crops(**top-left, center, bottom-right**) from temporal clip and obtain final prediction by averaging the scores for 3 crops.

Experiments

1. 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

Table 1. Video-level accuracy for different space-time attention schemes in TimeSformer. We evaluate the models on the validation sets of Kinetics-400 (K400), and Something-Something-V2 (SSv2). We observe that divided space-time attention achieves the best results on both datasets.

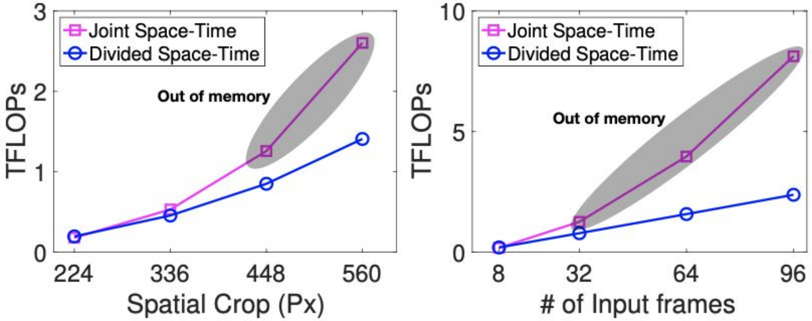
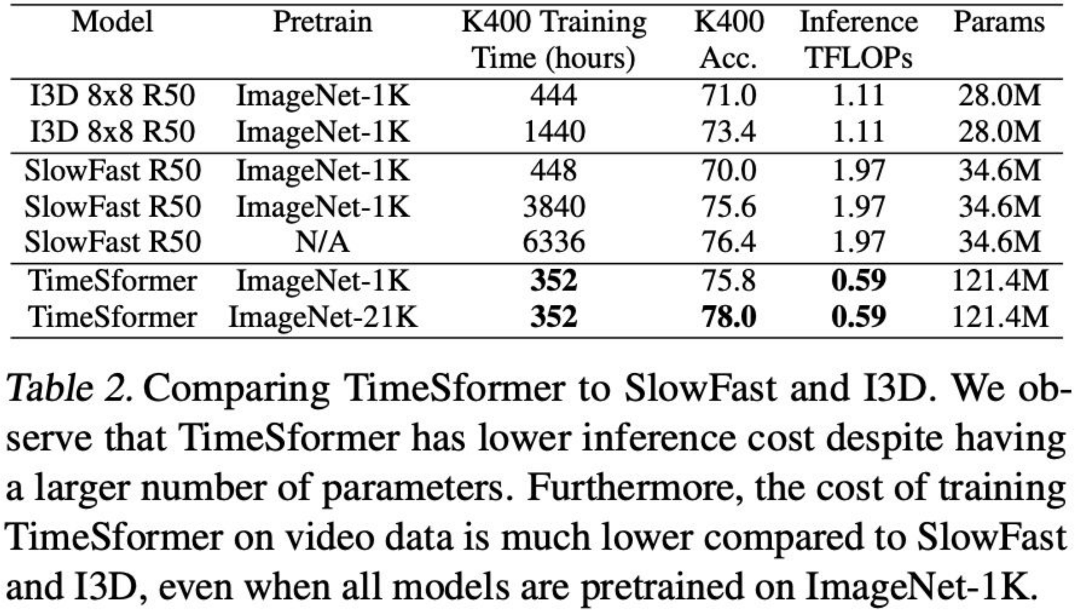


Figure 3. We compare the video classification cost (in TFLOPs) of Joint Space-Time versus Divided Space-Time attention. We plot the number of TFLOPs as a function of spatial crop size in pixels (left), and the number of input frames (right). As we increase the spatial resolution (left), or the video length (right), our proposed divided space-time attention leads to dramatic computational savings compared to the scheme of joint space-time attention.

Experiments

2. Comparison to 3D CNNs

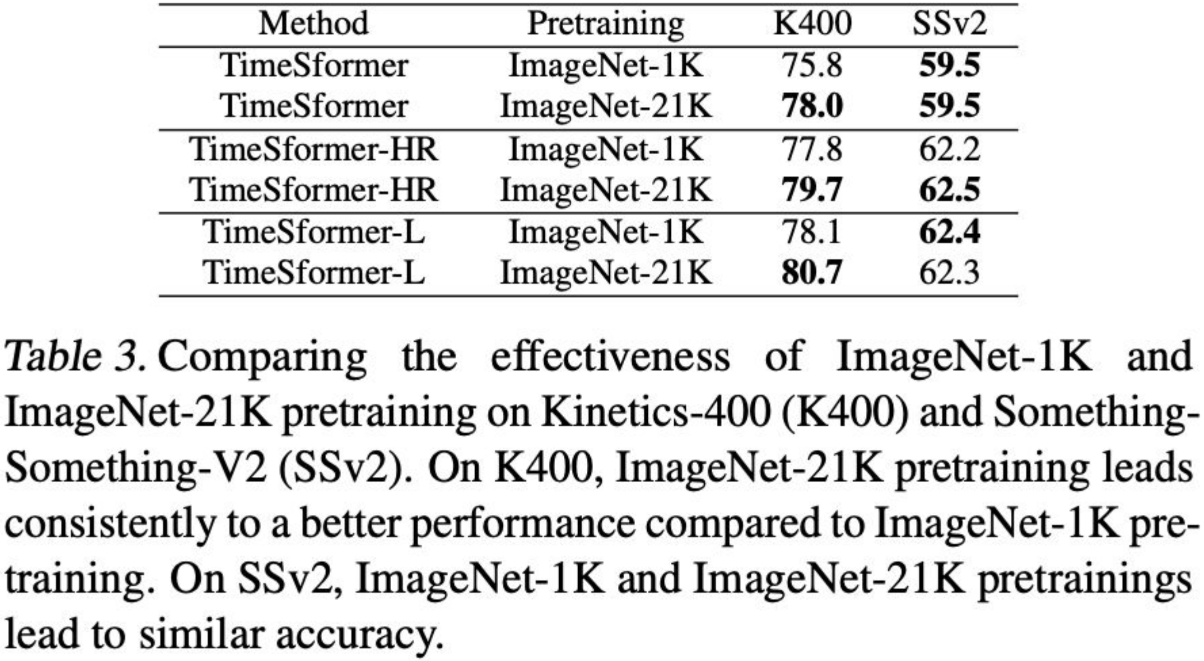


Model	Pretrain	K400 Training Time (hours)	K400 Acc.	Inference TFLOPs	Params
I3D 8x8 R50	ImageNet-1K	444	71.0	1.11	28.0M
I3D 8x8 R50	ImageNet-1K	1440	73.4	1.11	28.0M
SlowFast R50	ImageNet-1K	448	70.0	1.97	34.6M
SlowFast R50	ImageNet-1K	3840	75.6	1.97	34.6M
SlowFast R50	N/A	6336	76.4	1.97	34.6M
TimeSformer	ImageNet-1K	**352**	75.8	**0.59**	121.4M
TimeSformer	ImageNet-21K	**352**	**78.0**	**0.59**	121.4M

Table 2. Comparing TimeSformer to SlowFast and I3D. We observe that TimeSformer has lower inference cost despite having a larger number of parameters. Furthermore, the cost of training TimeSformer on video data is much lower compared to SlowFast and I3D, even when all models are pretrained on ImageNet-1K.

Experiments

2. Comparison to 3D CNNs



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

Table 3. Comparing the effectiveness of ImageNet-1K and ImageNet-21K pretraining on Kinetics-400 (K400) and Something-Something-V2 (SSv2). On K400, ImageNet-21K pretraining leads consistently to a better performance compared to ImageNet-1K pretraining. On SSv2, ImageNet-1K and ImageNet-21K pretrainings lead to similar accuracy.

Experiments

2. Comparison to 3D CNNs

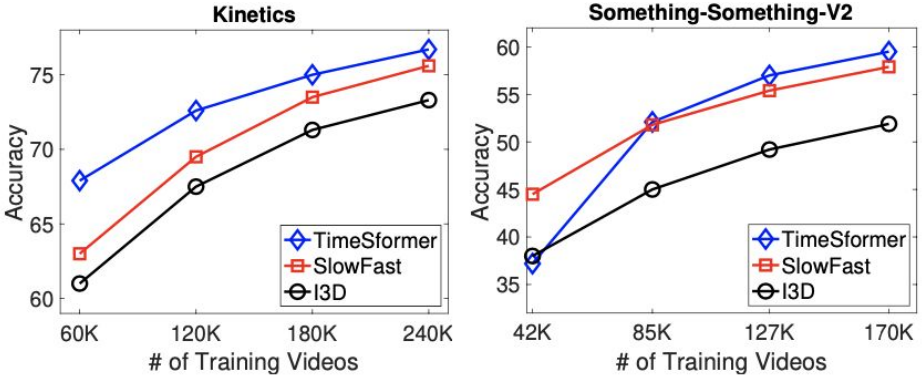


Figure 4. Accuracy on Kinetics-400 (K400), and Something-Something-V2 (SSv2) as a function of the number of training videos. On K400, TimeSformer performs best in all cases. On SSv2, which requires more complex temporal reasoning, TimeSformer outperforms the other models only when using enough training videos. All models are pretrained on ImageNet-1K.

Experiments

3. Varying the Number of Tokens

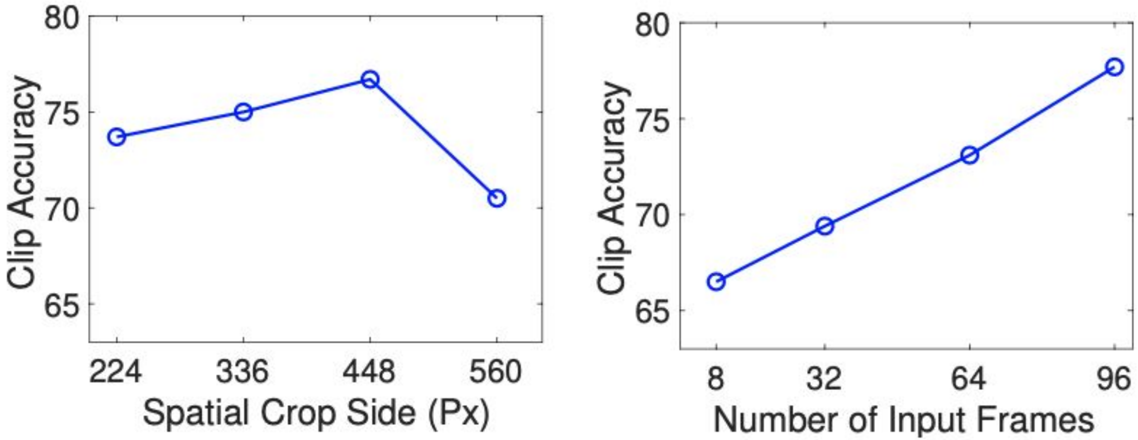
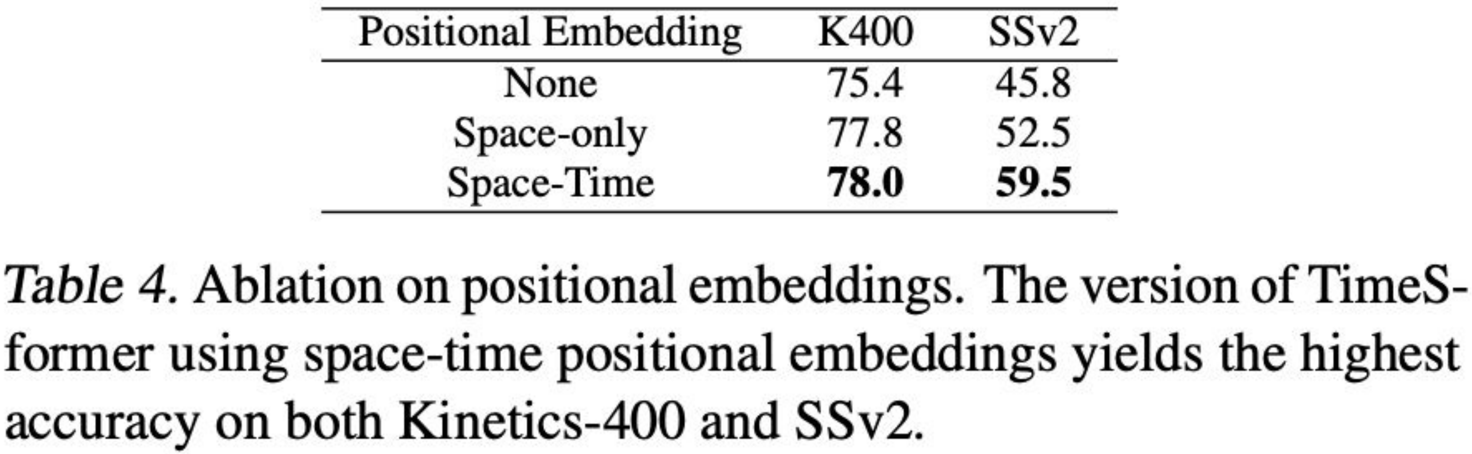


Figure 5. Clip-level accuracy on Kinetics-400 as a function of spatial crop size in pixels (left), and the number of input frames (right).

Experiments

4. The Importance of Positional Embeddings

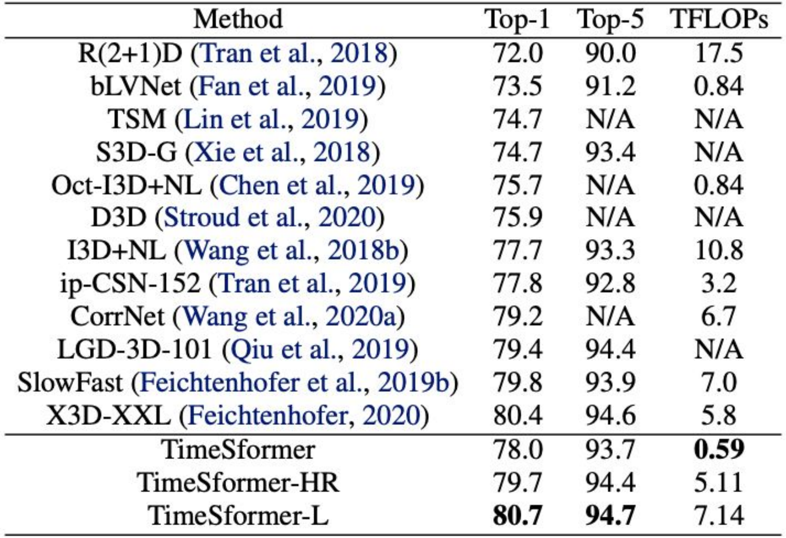


Positional Embedding	K400	SSv2
None	75.4	45.8
Space-only	77.8	52.5
Space-Time	**78.0**	**59.5**

Table 4. Ablation on positional embeddings. The version of TimeSformer using space-time positional embeddings yields the highest accuracy on both Kinetics-400 and SSv2.

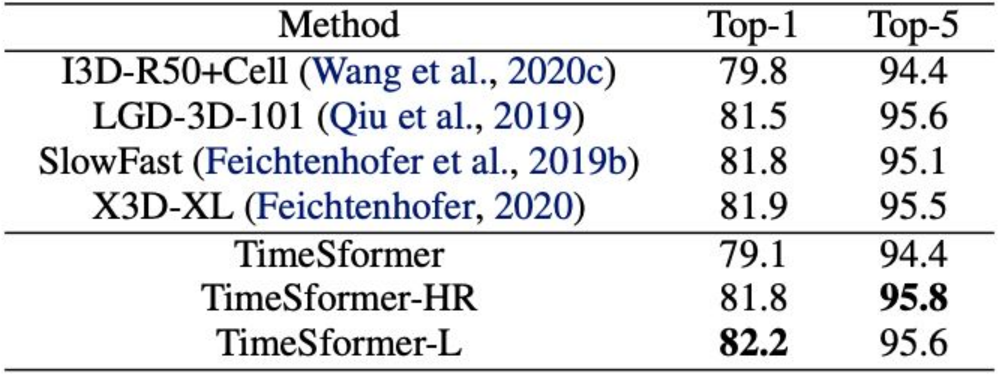
Experiments

5. Comparison to the State-of-the-Art



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

Table 5. Video-level accuracy on Kinetics-400.



Method	Top-1	Top-5
I3D-R50+Cell (Wang et al., 2020c)	79.8	94.4
LGD-3D-101 (Qiu et al., 2019)	81.5	95.6
SlowFast (Feichtenhofer et al., 2019b)	81.8	95.1
X3D-XL (Feichtenhofer, 2020)	81.9	95.5
TimeSformer	79.1	94.4
TimeSformer-HR	81.8	**95.8**
TimeSformer-L	**82.2**	95.6

Table 6. Video-level accuracy on Kinetics-600.

Experiments

5. Comparison to the State-of-the-Art

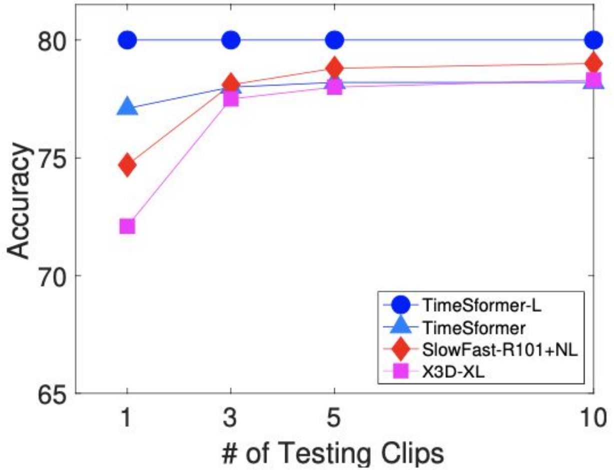
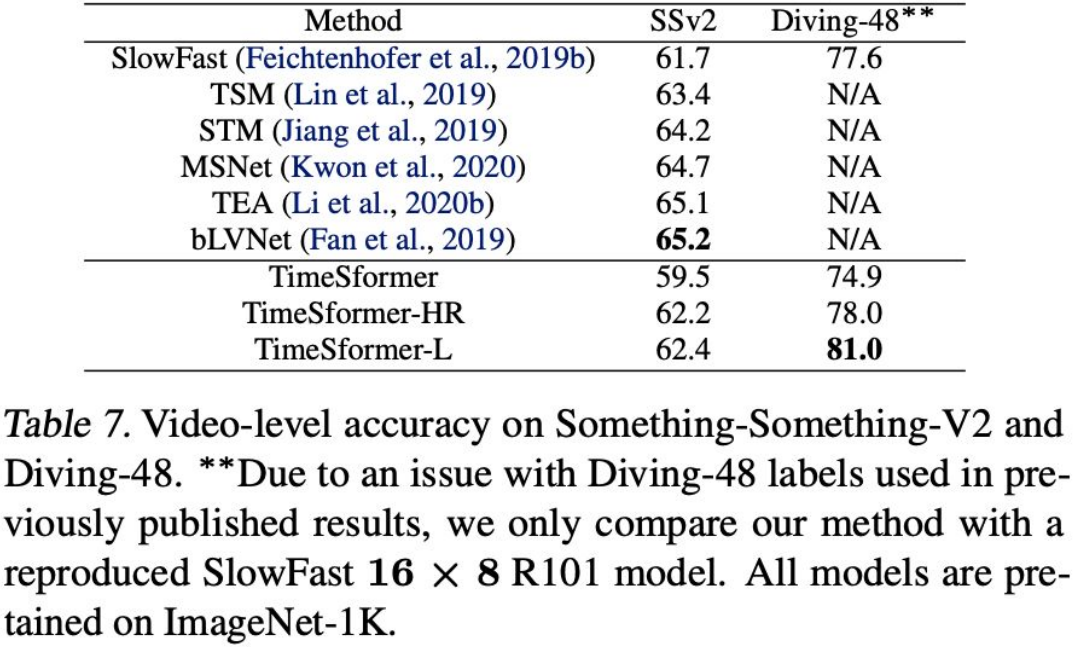


Figure 6. Video-level accuracy on Kinetics-400 vs the number of temporal clips used during inference. TimeSformer-L achieves excellent accuracy using a small number of clips, which leads to strong performance at low inference cost.

Experiments

5. Comparison to the State-of-the-Art

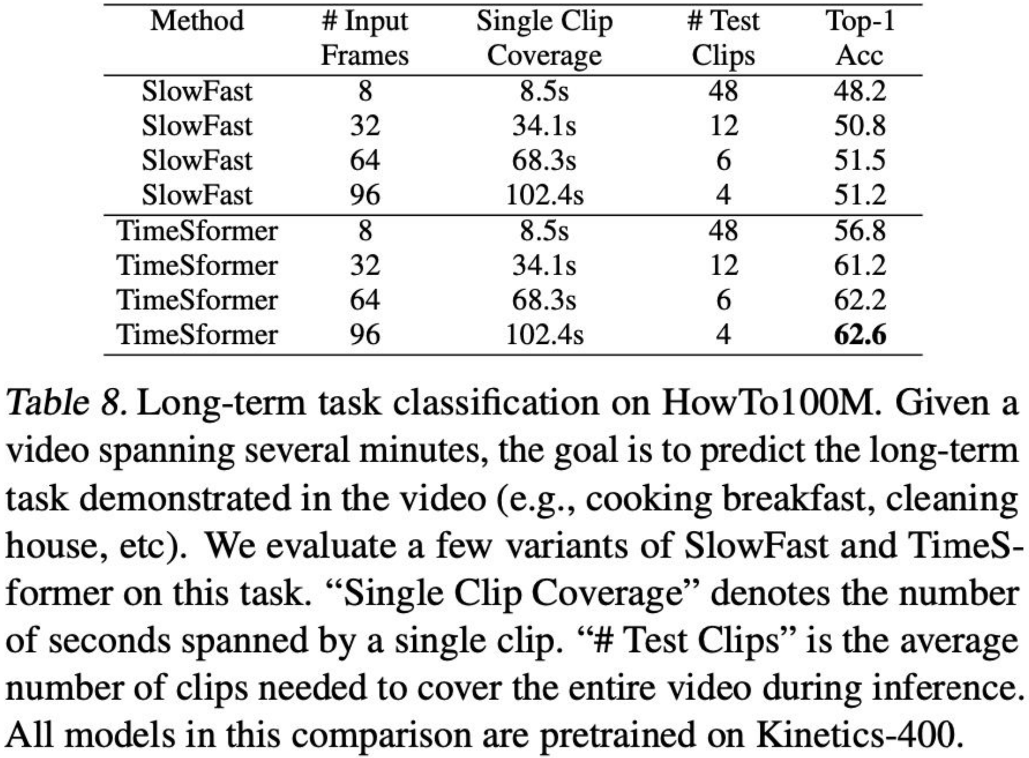


Method	SSv2	Diving-48**
SlowFast (Feichtenhofer et al., 2019b)	61.7	77.6
TSM (Lin et al., 2019)	63.4	N/A
STM (Jiang et al., 2019)	64.2	N/A
MSNet (Kwon et al., 2020)	64.7	N/A
TEA (Li et al., 2020b)	65.1	N/A
bLVNet (Fan et al., 2019)	**65.2**	N/A
TimeSformer	59.5	74.9
TimeSformer-HR	62.2	78.0
TimeSformer-L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

Table 7. Video-level accuracy on Something-Something-V2 and Diving-48. **Due to an issue with Diving-48 labels used in previously published results, we only compare our method with a reproduced SlowFast $\mathbf{16 \times 8}$ R101 model. All models are pretained on ImageNet-1K.

Experiments

6. Long-Term Video Modeling



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**

Table 8. Long-term task classification on HowTo100M. Given a video spanning several minutes, the goal is to predict the long-term task demonstrated in the video (e.g., cooking breakfast, cleaning house, etc). We evaluate a few variants of SlowFast and TimeSformer on this task. “Single Clip Coverage” denotes the number of seconds spanned by a single clip. “# Test Clips” is the average number of clips needed to cover the entire video during inference. All models in this comparison are pretrained on Kinetics-400.

Experiments

7. Additional Ablations

8. Qualitative Results

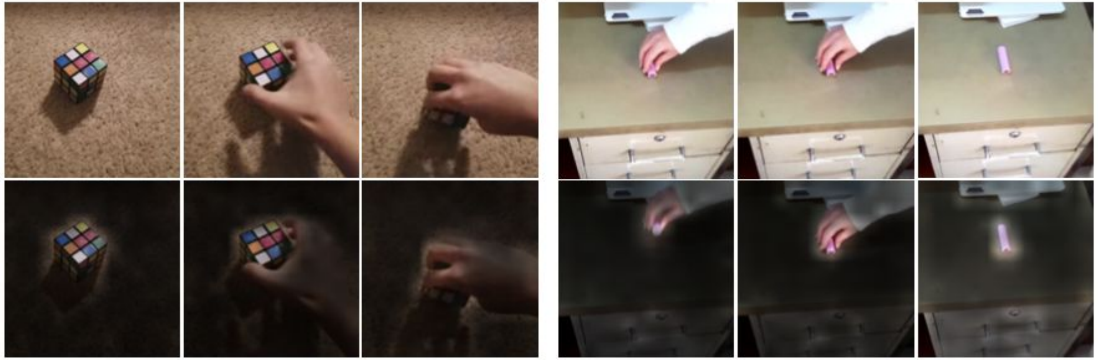
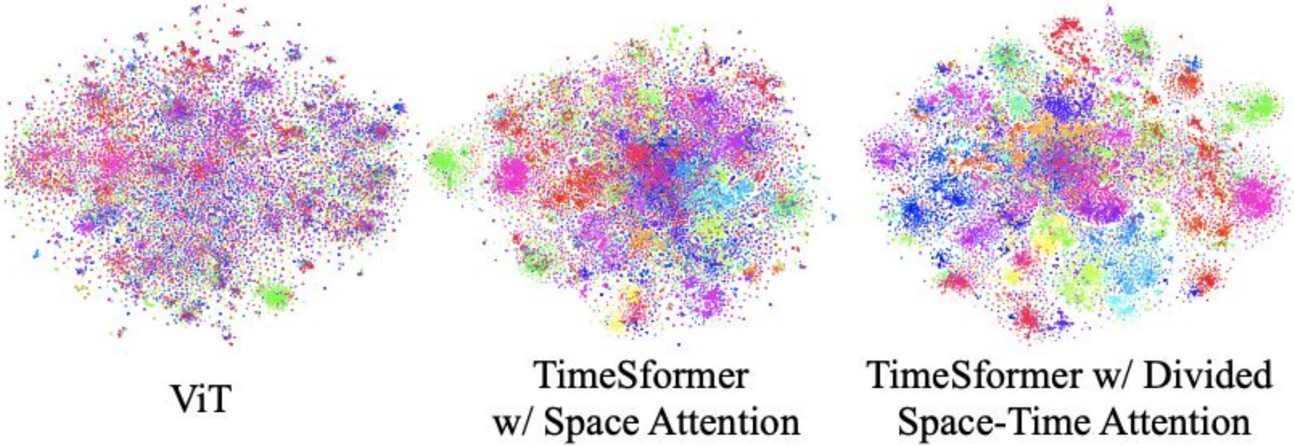


Figure 7. Visualization of space-time attention from the output token to the input space on Something-Something-V2. Our model learns to focus on the relevant parts in the video in order to perform spatiotemporal reasoning.



ViT

TimeSformer w/ Space Attention

TimeSformer w/ Divided Space-Time Attention

Figure 8. Feature visualization with t-SNE (van der Maaten & Hinton, 2008) on Something-Something-V2. Each video is visualized as a point. Videos belonging to the same action category have the same color. The TimeSformer with divided space-time attention learns semantically more separable features than the TimeSformer with space-only attention or ViT (Dosovitskiy et al., 2020).

Conclusion

TimeSformer

Different approach to video modeling

Effective and scalable video architecture built exclusively on space-time self-attention

(1) conceptually simple
(2) SOTA results
(3) low training and inference cost
(4) can be applied to clips of over one minute

Thank you