Shrinking Temporal Attention in Transformers for Video Action Recognition

Bonan Li[1*], **Pengfei Xiong**[2*], **Congying Han**[1†], **Tiande Guo**[1]

[1] University of Chinese Academy of Sciences
[2] PCG, Tencent
libonan16@mails.ucas.ac.cn, xiongpengfei2019@gmail.com, hancy@ucas.ac.cn, tdguo@ucas.ac.cn

Abstract

Spatiotemporal modeling in an unified architecture is key for video action recognition. This paper proposes a Shrinking Temporal Attention Transformer (STAT), which efficiently builts spatiotemporal attention maps considering the attenuation of spatial attention in short and long temporal sequences. Specifically, for short-term temporal tokens, query token interacts with them in a fine-grained manner in dealing with short-range motion. It then shrinks to a coarse attention in neighborhood for long-term tokens, to provide larger receptive field for long-range spatial aggregation. Both of them are composed in a short-long temporal integrated block to build visual appearances and temporal structure concurrently with lower costly in computation. We conduct thorough ablation studies, and achieve state-of-the-art results on multiple action recognition benchmarks including Kinetics400 and Something-Something v2, outperforming prior methods with 50% less FLOPs and without any pretrained model.

Introduction

Action recognition is a fundamental problem in video understanding tasks. Following the rapid development of online video, it becomes increasingly demanding applications with the rapid development of online video, in the fields of daily life, traffic surveillance, autonomous driving and so on. For most such applications, how to effectively build temporal structure and spatial appearances concurrently under different time length videos is a critical problem.

In recent years, end-to-end learning of transformer networks (Li et al. 2021; Arnab et al. 2021; Fan et al. 2021; Bertasius, Wang, and Torresani 2021; Liu et al. 2021a,b) has emerged as the prominent paradigm for video classification and action recognition, due to their excellent capabilities at capturing long-range temporal relationships. Though temporal structure is important for action recognition, it is also important to model the visual appearances under an unified architecture. The previous works propose several pure-transformer architectures which factorise different components of the transformer encoder over the space and time dimensions. Both the spatial and temporal features are divided into frame-level tokens to incorporate cross self-attention on spatiotemporal neighborhood in joint, mixed, divided, and interactive manners. However, dense dual attention brings huge computational overhead due to quadratic computational in time and space, while progressive self-attention lead model lack the ability to modelling long range spatial contextual relationships. It remains unclear how to model the spatiotemporal structure in an unified architecture effectively and efficiently.

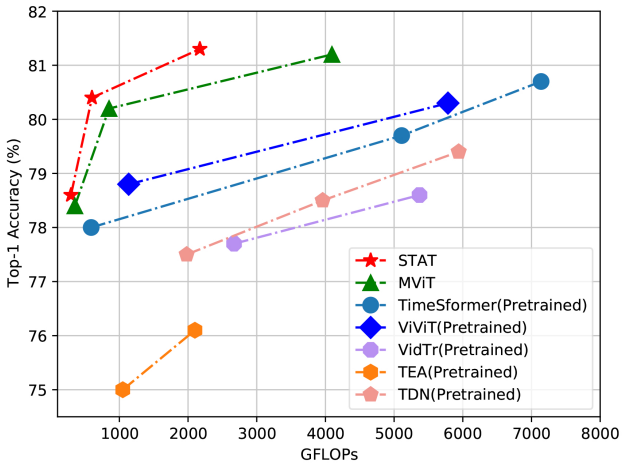
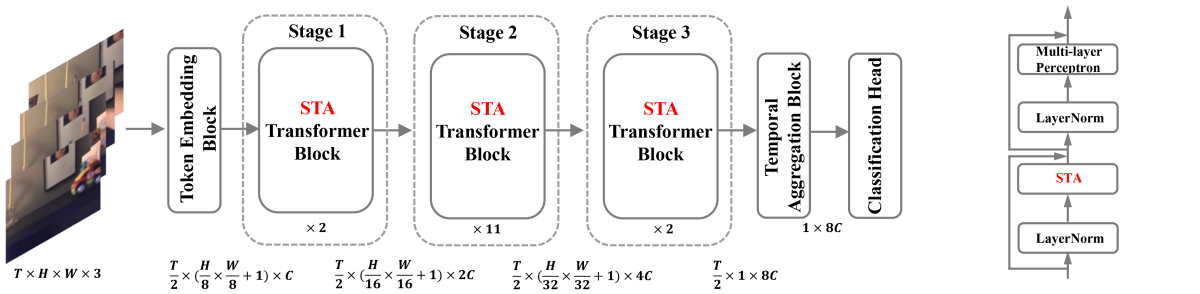


Figure 1: Top-1 accuracy and GFLOPs comparisons on Kinetics400. Results of existing state-of-the-art transformer-based models(MViT (Fan et al. 2021), TimeSformer (Bertasius, Wang, and Torresani 2021), ViViT (Arnab et al. 2021), VidTr (Li et al. 2021)) and CNN-based models(TEA (Li et al. 2020b), TDN (Wang et al. 2021)) are compared. Our proposed STAT outperforms all previous models with less FLOPs and without any pretrained models.

Commonly, spatiotemporal modeling can be separated two parts: short-range motion between adjacent frames and long-range visual aggregation. When dealing with short-range motion, fine grained details need to be considered. On the contrary, while in the long-term sequence, the influence of motion becomes smaller, and a larger receptive field is needed to obtain spatial aggregation. Motivated by this, we propose a Shrinking Temporal Attention Transformer

*Equal contribution, during internship at Tencent
†Corresponding author



(a) Architecture (b) STA Transformer Block

Figure 2: Overview of the Shrinking Temporal Attention Transformer. The main Shrinking Temporal Attention Block are adopted in a general transformer framework.

(STAT) to incorporate multi-level temporal structure into encoded features.

In detail, the proposed STAT contains three different self-attention modules: Current Attention, Short-term Attention and Long-term Attention. For Current Attention, we model interactions between tokens located at the same temporal index to capture rich spatial semantic information. In particular, we add an additional spatial class token to each frame to efficiently aggregate global spatial information. For Short-term Attention, the token is expected to focus on the object with actions in adjacent frames. To this end, a window based method is applied to obtain the tokens of adjacent frames and then we pool them along spatial dimension to ignore action independent information. Finally, for Long-term Attention, we use spatial class tokens of long-range frames as attending tokens, thus aggregating the visual information of the whole video. We test the proposed STAT on two standard benchmarks, Kinetics400 and Something-Something V2. STAT achieves 81.3 Top-1 accuracy with 2172 GFLOPs. While implemented with less frames, the accuracy still stays in 78.6 with only 296 GFLOPs, better than most of the state-of-the-art action recognition methods with 50% less FLOPs and higher performance, as shown in Figure 1.

Our main contributions are summarized as follows: (1) A new Shrinking Temporal Attention module (STA) is designed to encode the complementary spatiotemporal features in an unified framework, and it can be easily inserted into existing transformer architecture. (2) Two attention modules, short-term attention and long-term attention are presented respectively to provide different temporal token attentions. (3) We propose a simple yet effective network referred as STAT with our STA blocks with limited computation cost, and present a new record on action recognition benchmarks.

Related Work

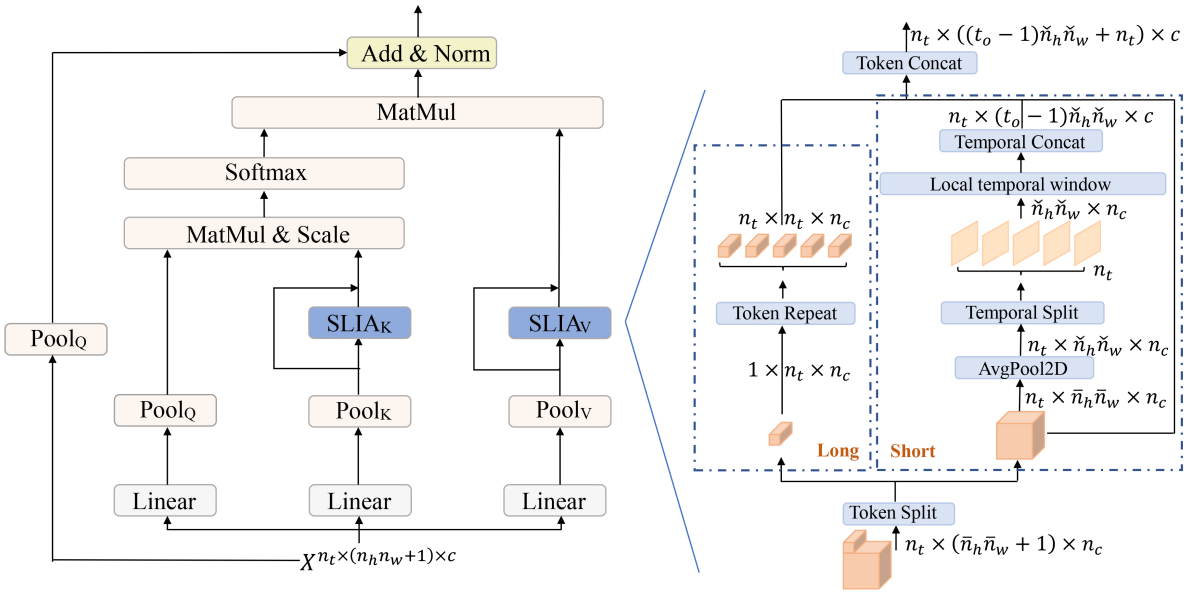
Action Recognition Current solutions can be broadly classified into two categories: CNN- and Transformer-based approaches. Early CNN-based approaches (Simonyan and Zisserman 2014; Feichtenhofer, Pinz, and Wildes 2017; Karpathy et al. 2014; Ji et al. 2012; Tran et al. 2015; Wang et al. 2018) adopt 3D convolution to joint modeling spatio and temporal information. SlowFast (Feichtenhofer et al. 2019) uses two pathways to focus on learning spatial semantics information and capturing rapidly changing motion, respectively. Transformer-based approaches (Li et al. 2021; Bertasius, Wang, and Torresani 2021; Arnab et al. 2021) are considered the current state-of-the-art as they can typically capture long-term information via self-attention mechanism. ViViT (Arnab et al. 2021) firstly uses spatial encoder to model interactions within the same frames and then to fuse temporal information with temporal encoder. Others explore different variants of joint space-time attentions. These methods are proved effective but often not efficient due to huge computational overhead.

Video Transformer Recently ViT (Dosovitskiy et al. 2020) achieves state-of-the-art results by replacing convolutions blocks with transformer blocks in image classification. Hence, many variants of transformer (Li et al. 2021; Bertasius, Wang, and Torresani 2021; Girdhar et al. 2019) have also been proposed to efficiently fuse long-term spatialtemporal information in video via self-attention mechanism. Although these models obtain the current state-of-the-art performance, they strongly rely on the vanilla ViT pretrained on large-scale datasets such as ImageNet (Deng et al. 2009).

Efficient Action Recognition In order to reduce the complexity of 3D CNN models, recent works (Feichtenhofer 2020; Sun et al. 2015; Tran et al. 2019; Xie et al. 2018; Wang et al. 2020; Zhou et al. 2018) attempt to factory convolutions across spatial and temporal dimensions with group convolutions. Another typical efficient video action recognition approach is based on (2+1)D CNN (Jiang et al. 2019; Lin, Gan, and Han 2019; Li et al. 2020b; Wang et al. 2021) which can reduce the heavy computations. STM (Jiang et al. 2019) learns feature-level motion features and spatiotemporal features with two separate blocks. Recently, a lightweight Transformer MViT (Fan et al. 2021) has also been proposed which is using multiscale hierarchies to learn the feature at distinct level. However, the quadratic attention complexity along time and space is still high.

Shrinking Temporal Attention Transformer

In this section, we start by describing Shrinking Temporal Attention (STA) which is the core component of Shrinking



(a) Shrinking Temporal Attention Module (b) Short-Long Integrated Attention

Figure 3: Overview of the Shrinking Temporal Attention.

Temporal Attention Transformer (STAT). The key idea of STA is to interact with frames at different temporal terms through Current Attention, Short-term Attention and Long-term Attention, respectively. Then, the implementation of STAT will be explained in details and Figure 2 is the diagram of our network.

Shrinking Temporal Attention Module

Shrinking Temporal Attention is an "factor" attention operator that focuses on current, short-term and long-term frames at distinct granularities, as shown in Figure 3.

Current Attention Semantic information is the basis of capturing motion, so we use fine-grained to learn spatial features of current frame. Concretely, given a video tensor $X \in \mathbb{R}^{n_t \times (n_h n_w + 1) \times n_c}$, where n_t denotes the video length, n_h and n_w denotes the feature map height and width, and n_c denotes the number of channel. It is worth noting that an extra spatial class token $X^{(:,0,:)}$ for each temporal index is introduced to represent global spatio information. Following Multi-Headed Self Attention(MHSA) (Dosovitskiy et al. 2020), three linear projectors are applied to generate query tensor $Q^{n_t \times (n_h n_w + 1) \times n_c}$, key tensor $K^{n_t \times (n_h n_w + 1) \times n_c}$ and value tensor $V^{n_t \times (n_h n_w + 1) \times n_c}$.

$$Q = \mathcal{F}_q(X) \quad K = \mathcal{F}_k(X) \quad V = \mathcal{F}_v(X) \tag{1}$$

where $\mathcal{F}_q(\cdot)$, $\mathcal{F}_k(\cdot)$ and $\mathcal{F}_v(\cdot)$ denote projection layers with weights of dimensions $n_c \times n_c$.

High resolution feature maps often result in unacceptable computational overhead that are difficult to apply to real-world scenarios. We pool $Q^{(:,1:,:)}$, $K^{(:,1:,:)}$, $V^{(:,1:,:)}$ with pooling operator to obtain tensors $\hat{Q}^{n_t \times \check{n}_h \check{n}_w \times n_c}$, $K_{CA}^{n_t \times \bar{n}_h \bar{n}_w \times n_c}$, $V_{CA}^{n_t \times \bar{n}_h \bar{n}_w \times n_c}$.

$$\hat{Q} = \mathcal{P}_{CA}(Q; \theta_Q)\ K_{CA} = \mathcal{P}_{CA}(K; \theta_K)\ V_{CA} = \mathcal{P}_{CA}(V; \theta_V) \tag{2}$$

where $\mathcal{P}_{CA}(;\theta)$ is the pooling operator and we implement it with convolution. θ is the hyperparameter required for convolution, such as kernel, padding and stride. It should be emphasized that Q will be pooled only when we need to downsample the feature map. For K and V, we pool them at each STA layer and make $\theta_K = \theta_V$. In addition, the clip length and feature dimension remain the same.

Short-Term Attention Compared to long range frames, continuous frames in short ranges tend to be highly correlated and can provide rich action cues. However, overly complex interactions between frames would prevent the model from focusing on learning semantic information, so we generate sub-fine grained tokens with short-term block for local motion modeling in short-term frames. Specifically, we perform secondary down sampling on K_{CA}, V_{CA} to obtain $K_{SA}^{n_t \times \check{n}_h \check{n}_w \times n_c}$, $V_{SA}^{n_t \times \check{n}_h \check{n}_w \times n_c}$ with pooling operator $\mathcal{P}_{SA}(;\theta)$.

$$K_{SA} = \mathcal{P}_{SA}(K_{CA}; \theta_{K_{CA}})\ V_{SA} = \mathcal{P}_{SA}(V_{CA}; \theta_{V_{CA}}) \tag{3}$$

In order to control the amount of parameters and computation, we use 2D-Avgpooling to realize $\mathcal{P}_{SA}(;\theta)$. To obtain short-term continuous frames, we set a local temporal window with size equal to t_o and it slides along temporal axis. When the query token located in i^{th} frame, we propose to calculate attention with $K_{SA}^{(\{i-\lfloor \frac{t_o}{2} \rfloor : i; i+1: i+\lfloor \frac{t_o}{2} \rfloor + 1\}, :, :)}$, $V_{SA}^{(\{i-\lfloor \frac{t_o}{2} \rfloor : i; i+1: i+\lfloor \frac{t_o}{2} \rfloor + 1\}, :, :)}$. It is not difficult to find that $K_{SA}^{(i,:,:)}$ is not used as attending tokens. This design can bring

two benefits: (1) reducing the computational effort. (2) reducing the risk of over-fitting to spatial information.

Long-Term Attention To fully capture temporal information over the entire video, we also introduce token for long-term motion modeling. Naturally, the correlation between two frames decreases with increasing interval. Therefore, the coarse grained tokens are applied to perform Long-term Attention. Here, instead of using global pooling in width and height, spatial class tokens $X^{(:,0,:)}$ are adopted to represent global information. The motivation for this is that pooling can not retain the position information related to the motion, while spatial class token can keep the multi-dimensional attribute of motion through continuous optimization. By employing long-term block, the spatial class tensors are firstly extracted from K, V as $K^{(:,0,:)} \in \mathbb{R}^{n_t \times n_c}$, $V^{(:,0,:)} \in \mathbb{R}^{n_t \times n_c}$. After that, we expand an additional dimension on them and replicate T times on this dimension. Assuming $x^{n_t \times n_c}$,

$$x_e = Expand(x) \in \mathbb{R}^{1 \times n_t \times n_c} \tag{4}$$

$$x_r = Repeat(x_e) \in \mathbb{R}^{n_t \times n_t \times n_c} \tag{5}$$

where $Expand(\cdot)$ and $Repeat(\cdot)$ denote expand and repeat operations respectively. According Equation 4 and Equation 5, the key and value tensors for Long-term Attention can be obtained by following:

$$\begin{aligned} K_{LA} &= Repeat(Expand(K^{(:,0,:)})) \\ V_{LA} &= Repeat(Expand(V^{(:,0,:)})) \end{aligned} \tag{6}$$

For convenience, Short-term Attention and Long-term Attention are named as Short-Long Integrated Attention. As such, we got all attending tokens at Current Attention, Short-term Attention and Long-term Attention. In the following, we elaborate the attention computation of STA.

Attention Computation Before describing the attention calculation method, we clarify the query tensor of query tokens and the key/value tensor of attending tokens again.

$$\begin{aligned} \hat{Q} &= Concat(Q^{(:,0:1,:)}, \hat{Q}) \\ \hat{K} &= Concat(K_{CA}, K_{SA}, K_{LA}) \\ \hat{V} &= Concat(V_{CA}, V_{SA}, V_{LA}) \end{aligned} \tag{7}$$

where $\hat{Q} \in \mathbb{R}^{n_t \times (\tilde{n}_h \tilde{n}_w + 1) \times n_c}$ and $\hat{K}, \hat{V} \in \mathbb{R}^{n_t \times (\bar{n}_h \bar{n}_w + (t_o - 1)\tilde{n}_h \tilde{n}_w + n_t) \times n_c}$. $Concat(\cdot)$ denotes operator which can concatenate the given sequence of tensors in 2^{th} dimension. Attention is now computed on these tensors, with the following operation:

$$Attention(\hat{Q}, \hat{K}, \hat{V}) = Softmax(\frac{\hat{Q}\hat{K}^T}{\sqrt{n_c}})\hat{V} \tag{8}$$

where $\sqrt{n_c}$ is the normalization factor and the output of $Attention(\cdot)$ has the same shape with $\hat{Q}^{n_t \times (\tilde{n}_h \tilde{n}_w + 1) \times n_c}$.

Shrinking Temporal Attention Transformer

Our whole model builds upon three components: Token Embedding Block, Shrinking Temporal Attention Block and Temporal Aggregation Block. Token Embedding Block is used to dice the video to tokens and STA Blocks are applied to efficiently learn spatiotemporal information. In addition, we adopt a light Temporal Aggregation Block to fuse global temporal features.

Modules	GFLOPs	Top-1	Top-5
CA	58.2	77.3	92.9
CA + SA	59.0	77.8	93.3
CA + LA	58.4	78.3	93.5
CA + SA + LA	**59.1**	**78.6**	**93.8**

Table 1: Effect of different attention components in STA. CA denotes current attention, SA denotes short-term attention and LA denotes long-term attention.

Token Embedding Block By given a video $Vi \in \mathbb{R}^{T \times H \times W \times D}$, different from simple tokenization in ViT, we propose to decompose it into $n_t n_h n_w$ overlapping tokens $x \in \mathbb{R}^{1 \times n_c}$ with multiple consecutive 3D-convolutional layers. Some concurrently proposed works (Wu et al. 2021; Guo et al. 2021) also attempt to use this method to initialize the token and refer to it as conv-stem. However, unlike existing work, our motivation is mainly based on the following two points: (1) convolution operation is more advantageous in extracting local features. (2) no need to focus on long-term temporal features when extracting the underlying features. In the model implementation, instead using residual block, three vanilla 3D-convolutional layers are used to continuously downsample the input video and make $n_t = \frac{T}{2}, n_h = \frac{H}{8}, n_w = \frac{W}{8}, n_c = 192$. Perhaps even better performance can be obtained using new operators, such as T2T (Yuan et al. 2021a) and VOLO (Yuan et al. 2021b), but this is not the concern of this paper. Then, we introduce an additional spatial class token for each frame to indicate global spatial information. Therefore, token embedding block actually outputs a total of $n_t(n_h n_w + 1)$ tokens to next block. Finally, we add two positional embeddings to initial visual tokens for spatial and temporal, respectively.

Shrinking Temporal Attention Block Shrinking Temporal Attention Block is the core module of STAT, which is based on STA. Considering X_{in} to be the input of this block, the output X_{out} of this single transformer block can be computed by following:

$$X_a = STA(LN(X_{in})) + X_{in} \tag{9}$$

$$X_{out} = MLP(LN(X_a)) + X_a \tag{10}$$

where $STA(\cdot)$ denotes shrinking temporal attention, $LN(\cdot)$ denotes Layer Normalization and $MLP(\cdot)$ denotes Multi-layer Perceptron. By default, we set the dimension of MLP hidden layer as $4n_c$.

We realize the main part of STAT by stacking multiple STA blocks. Blocks are split into multiple stages and the blocks in the same stage are operated on the same scale. As the index of the stages increases, the feature map resolution decreases and the channel dimension increases gradually. Concretely, we set 15 STA blocks and assign them to 3 stages. When transitioning from one stage to the next, we

WinSize (t_o)	GFLOPs	Top-1	Top-5
SA_3	118.6	79.9	94.3
SA_5	120.1	**80.4**	**94.6**
SA_7	121.5	**80.4**	94.4
SA_9	123.0	80.2	94.4

Table 2: Effect of various window size in short-term attention. The input cliplength is set to 32 in order to perform multiple comparison experiments.

Global spatial embedding	Top-1	Top-5
spatial avg-pooling	77.4	93.2
spatial class token	**78.6**	**93.8**

Table 3: Different strategies of global spatial embedding.

expand the dimension to $2\times$ through MLP, and use pooling operator to down sample the resolution of feature map to $4\times$. For clarity, we list the details of our network in appendix.

Temporal Aggregation Block Simply averaging spatial class tokens across frames and using it for final classification would seriously ignore the clues from temporal information. Here, a light temporal aggregation block is proposed to fuse temporal information. In particular, a learnable x_{cls} token is introduced to the whole video and interacts with all spatial class tokens using l_{ta} Multi-Headed Self Attention layers. Afterwards, we feed x_{cls} into classifier and calculate loss with cross-entropy.

Experiment

Datasets and Experiment Setting

Datasets The proposed model is trained and evaluated on the two public large-scale action recognition datasets, Kinetics 400 and Something-Something V2(SSv2). Kinetics400 (Kay et al. 2017) consists of 240k training videos and 20k validation 10 second videos sampled at 25fps and labeled using 400 classes. As small fraction of the download URLs is no longer valid, we note the versions of the datasets used in this paper consist of approximately 260k samples. Something-Something V2(SSv2) (Goyal et al. 2017) dataset consists about 220K $2 \sim 6$ second short videos collected by performing the same action with different objects. In constract to Kinetics400, SSv2 is a temporal-related datasets that focus on the motion property than scene context. For both of these two datasets, the methods are learned on the training set and evaluated on the validation set. And all of our ablation experiments are performed on Kinetics400.

Implementation Details In our experiment, we initialize the network weights from the random initialization *without any pretrained model* and train it with AdamW optimizer as the recipe following (Touvron et al. 2021). For the temporal domain, we randomly sample a frame from each segment to obtain one input sequence with $T = 16, T = 32$ or $T = 64$ frames. Meanwhile, we fix the short side of these frames to 256 and perform data argumentation following MViT (Fan et al. 2021) to obtain the data with size 224×224 for the spatial domain. Specific implementation details can be found in the appendix.

During the test, different from common practice, we perform less sampling along spatial and temporal axis because STAT has high efficiency in spatio-temporal modeling. Specifically, for Kinetics400, we report average results for 5×1 views (5 temporal clips and 1 spatial crops) when $T = 16, 32$ and 3×3 views (3 temporal clips and 3 spatial crops) when $T = 64$. For SSv2, we report average results for 1×3 views (1 temporal clip and 3 spatial crops) for all setting.

Temporal aggregation	0	1	2	4
Top-1	75.7	**78.6**	**78.6**	78.5
Top-5	92.2	93.8	**93.9**	93.8

Table 4: The effect of varying the number of temporal aggregation layers. Note that $l_{ta} = 0$ demonstrates the result of average pooling method.

Conv layers	Kernel	Stride	Top-1
1	$[3, 7, 7]$	$[2, 4, 4]$	78.2
3	$[3, 3, 3]$ $[3, 3, 3]$ $[3, 3, 3]$	$[2, 2, 2]$ $[1, 2, 2]$ $[1, 2, 2]$	**78.6**
5	$[3, 3, 3]$ $[3, 3, 3]$ $[3, 3, 3]$ $[3, 3, 3]$ $[3, 3, 3]$	$[2, 2, 2]$ $[1, 2, 2]$ $[1, 2, 2]$ $[2, 2, 2]$ $[1, 2, 2]$	78.3

Table 5: The effect of progressively adding convolutional layers in token embedding block.

Ablation of Shrinking Temporal Attention

Effect of Various Attention For this first set of experiments, the ablation study is conducted to evaluate the effectiveness of each of the attention components individually and in combination on Kinetics400. In Table 1, we present the results of four settings while the computational complexity measured in one clip with $T = 16$. It can be observed that both CA+SA and CA+LA obtain a better performance than CA, which can be attributed to the fact that the temporal information allows the network to focus on learning objects with motions. Meanwhile, long-term information brings 1% performance increase, which indicates that global temporal information is crucial for action recognition task. Naturally, CA+SA+LA yields the best results.

Effect of Window Size The ablation in Table 2 analyzes the Top-1 accuracy of STAT by varying the window size t_o introduced in short-term attention. Here, the input clip length is set to 32 in order to perform multiple comparison experiments. We find that the accuracy initially increases with increasing window size until it achieves its peak accuracy of 80.4% at $t_o = 5$ and flattens until $t_o = 7$ after which it declines upon further increase of the window size.

Method	Backbone	Pre-train	Frames×Clips×Crops	GFLOPs	Top-1	Top-5
2D+3D CNNs:						
SlowFast	ResNet-50	-	(8+32)×10×3	1971	77.0	92.6
SlowFast	ResNet-101	-	(16+64)×10×3	6390	78.9	93.5
X3D	ResNet-50	-	16×10×3	1452	79.1	93.9
X3D	ResNet-101	-	16×10×3	5823	80.4	94.6
ip-CSN	-	Sports1M	32×10×3	3264	79.2	93.8
I3D_NL	ResNet-50	ImageNet	128×10×3	8,460	76.5	92.6
I3D_NL	ResNet-101	ImageNet	128×10×3	10,800	77.7	93.3
(2+1)D CNNs:						
TSM	ResNet-50	ImageNet	16×10×3	650	74.7	-
MSNet	ResNet-50	ImageNet	16×10×1	670	76.4	-
bLVNet	bLResNet-50	ImageNet	24×3×3	840	73.5	91.2
TEA	ResNet-50	ImageNet	8×10×3	1050	75.0	91.8
TEA	ResNet-50	ImageNet	16×10×3	2100	76.1	92.5
TDN	ResNet-101	ImageNet	8×10×3	1980	77.5	93.6
TDN	ResNet-101	ImageNet	16×10×3	3960	78.5	93.9
STM	ResNet-50	ImageNet	16×10×3	2010	73.7	91.6
Transformers:						
MViT	-	-	16×5×1	353	78.4	93.5
MViT	-	-	32×5×1	850	80.2	94.4
MViT	-	-	64×3×3	4,095	81.2	95.1
VidTr	ViT-L	ImageNet	16×10×3	5370	78.6	93.5
VidTr	ViT-L	ImageNet	32×10×3	10530	79.1	93.9
TimeSformer	ViT-L	ImageNet	96×1×3	7140	80.7	94.7
ViViT	ViT-L	ImageNet	16×4×3	17352	80.6	94.7
STAT	-	-	16×5×1	**296**	**78.6**	**93.8**
STAT	-	-	32×5×1	**601**	**80.4**	**94.6**
STAT	-	-	64×3×3	**2172**	**81.3**	**95.1**

Table 6: Comparison results of STAT with previous methods on Kinetics400(Kay et al. 2017) validation set. The state-of-the-art methods SlowFast (Feichtenhofer et al. 2019), X3D (Feichtenhofer 2020), ip-CSN (Tran et al. 2019), I3D (Wang et al. 2018), TSM (Lin, Gan, and Han 2019), bLVNet (Fan et al. 2019), TEA (Li et al. 2020b), STM (Jiang et al. 2019), MSNet (Kwon et al. 2020) TDN (Wang et al. 2021), MViT (Fan et al. 2021), VidTr (Li et al. 2021), TimeSformer (Bertasius, Wang, and Torresani 2021), ViViT (Arnab et al. 2021) are adopted.

These suggest that the introduction of short-term information would be beneficial to the network for capturing fast motions. However, a lager window can lead to an inability to focus on learning spatial semantic information. Hence, considering both computational complexity and performance, we set $t_o = 3, 5, 7$ when $T = 16, 32, 64$, respectively.

Importance of Spatial Class Token For shrinking temporal attention, the global spatio information is the key to implement long-term attention. Herein, we compare two strategies to generate global spatio embedding in Table 3. Our results show that the an extra spatio class token outperform the embedding obtained by spatio avg-pooling 1.2% and 0.6% for Top-1 and Top-5, respectively. This phenomenon is in line with our expectation, since global averaging pooling in the spatial dimension would ignore the essential location information. In contrast, spatial class token can retain the position information related to the motion while keep the multi-dimensional attribute of motion through attention mechanism. In addition, adding an extra token to each frame only bring minor computational overhead.

Ablation of Other Modules

Token Embedding Block Design As mentioned in the section above, we maintain that using more convolutional layers could effectively learning low-level features. We vary the number of 3D-convolutional layers in token embedding block from 1 to 5 in Table 5. Specifically, we replace part of previous transformer blocks with convolutional blocks while ensuring that computational overhead and numbers of parameters remain approximately the same. As expected, increasing the number of convolutional layers from 1 to 3 yields the performance gain of 0.4%. Meanwhile, we note that too many convolutional layers would lead to accuracy degradation. One possible reason is that too many convolution layers lead the model have unsufficient capacity to learn temporal information.

Varying the Number of Temporal Aggregation Layers Temporal aggregation Block is used in the final stage to fuse global spatio-temporal information. To investigate the importance of this block, we first average the final spatial class tokens of each frame and then feed it to classification head. As shown in Table 4, replacing temporal averaging

Method	Backbone	Pre-train	Frames×Clips×Crops	GFLOPs	Top-1	Top-5
(2+1)D CNNs:						
bLVNet	ResNet-101	ImageNet	$8\times1\times1$	32	60.2	87.1
TSM	ResNet-50	ImageNet	$16\times2\times3$	390	63.4	88.5
SmallBigNet	ResNet-50	ImageNet	$16\times2\times3$	-	63.8	88.9
MSNet	ResNet-50	ImageNet	$(16+8)\times10\times1$	1010	67.1	**91.0**
STM	ResNet-50	ImageNet	$16\times10\times3$	2,010	64.2	89.8
Transformers:						
MViT	-	K400	$16\times1\times3$	212	64.7	89.2
MViT	-	K400	$32\times1\times3$	510	67.1	90.8
MViT	-	K400	$64\times1\times3$	1365	**67.7**	90.9
TimeSformer	ViT-L	ImageNet	$96\times1\times3$	7,140	62.4	-
ViViT	ViT-L	ImageNet	$16\times4\times3$	11,892	65.4	89.8
STAT	-	K400	$16\times1\times3$	**178**	**65.1**	**89.4**
STAT	-	K400	$32\times1\times3$	**361**	**67.3**	**90.8**
STAT	-	K400	$64\times1\times3$	**724**	67.6	90.9

Table 7: Comparison results of STAT with previous methods on Something-Something V2(SSv2) (Goyal et al. 2017) validation set. The state-of-the-art methods bLVNet (Fan et al. 2019), TSM (Lin, Gan, and Han 2019), SmallBigNet (Li et al. 2020a), MSNet (Kwon et al. 2020), STM (Jiang et al. 2019), MViT (Fan et al. 2021), TimeSformer (Bertasius, Wang, and Torresani 2021), ViViT (Arnab et al. 2021) are adopted.

with one temporal aggregation layer improves the Top-1 accuracy from 75.7% to 78.6%. We attribute this performance gain to the fact that temporal aggregation block can efficiently fuse temporal information, whereas the average pooling approach loses valid information from temporal cues. However, there is no significant change in accuracy when the number of layers is further increased. Consequently, we set $l_{ta} = 1$ for all experiments.

Comparisons with the State-of-the-Arts

In this section, we compare STAT with the existing state-of-the-art action recognition methods on Kinetics400 and Something-Something V2(SSv2). The comprehensive statistics, include classification results, inference protocols, and the corresponding FLOPs.

Kinetics Table 6 presents that our proposed STAT outperform the state-of-the-art methods on Kinetics400. We take 1 spatial crops for 5 temporal view following standard practice in (Fan et al. 2021). Three types of methods based on 3D CNNs, (2+1)D CNNs and Transformers are compared respectively. Due to the high computation costs of 3D CNNs, the FLOPs of methods in the first compartment are typically higher than others. Among all these existing methods, the most effective and accurate one is X3D(Feichtenhofer 2020), with 5823 GFLOPs and 80.4 Top-1 accuracy. Compared with it, our proposed STAT achieves the same accuracy with only 601 GFLOPs (0.1X), and better accuracy (81.3 vs 80.4) with half FLOPs. In the cluster of (2+1)D CNNs, although they has made optimizations in the amount of calculation, the accuracy also decreases. TDN(Wang et al. 2021) achieves balance between FLOPs and accuracy. The calculation of TDN is 1980 FLOPs and the Top-1 accuracy is only 77.5, which is about 6x FLOPs of STAT while similar performance is obtained. The same conclusion can be reflected in the results of transformer based models. Limited by transformer structure, VidTr(Li et al. 2021), TimeSformer(Bertasius, Wang, and Torresani 2021) and ViViT(Arnab et al. 2021) adopt very complicated networks to achieve state-of-the-art performance. MViT(Fan et al. 2021) has made many improvements. Nevertheless, the quadratic computational in time and space still brings huge computation. Compared with MViT, the proposed STAT achieves a little higher accuracy with 50% less FLOPs (2172 vs 4095). Furthermore, While implemented with less frames, the Top-1 accuracy decreases from 81.3 to 78.6, which is still a competitive result compared with different structure of previous models.

SSv2 The similar conclusion is presented in Table 7 on Something-Something V2 dataset. SSv2 differs from other datasets. The backgrounds and objects are quite similar in that. It needs the highly effective model to recognise fine-grained motion patterns across different classes. However, STAT still achieves state-of-the-art Top-1 accuracies with different frames, especially compared with MViT(Fan et al. 2021) and MSNet(Kwon et al. 2020). The results suggest that Shrinking Temporal Attention is an effective approach for modeling the short-range motion and long-range visual aggregation in an unified network.

Conclusion

This paper proposes an unified network STAT to build spatiotemporal attention considering both the short-term motion and long-term aggregation. It adopts an Shrinking Temporal Attention manner to separately handle with the variety of short-range and long-range videos. It greatly reduces the huge computational overhead of quadratic computational in time and space, and obtaines compact spatiotemporal feature representation. Experiments show that, STAT achieves state-of-the-art results on multiple action recognition benchmarks with 50% less FLOPs and without any pretrained model.

Acknowledgements

This paper is supported by the National key research and development program of China (2021YFA1000403), the National Natural Science Foundation of China (Nos. U19B2040), the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA27000000, and the Fundamental Research Funds for the Central Universities. We are grateful for discussions with Xuecheng Nie and Han Fang.

References

Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; and Schmid, C. 2021. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is Space-Time Attention All You Need for Video Understanding? *arXiv preprint arXiv:2102.05095*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of Computer Vision and Pattern Recognition*, 248–255.

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

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*.

Fan, Q.; Chen, C.-F.; Kuehne, H.; Pistoia, M.; and Cox, D. 2019. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. *arXiv preprint arXiv:1912.00869*.

Feichtenhofer, C. 2020. X3d: Expanding architectures for efficient video recognition. In *Proceedings of Computer Vision and Pattern Recognition*, 203–213.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of International Conference on Computer Vision*, 6202–6211.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of Computer vision and Pattern Recognition*, 4768–4777.

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *Proceedings of Computer Vision and Pattern Recognition*, 244–253.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of International Conference on Computer Vision*, 5842–5850.

Guo, J.; Han, K.; Wu, H.; Xu, C.; Tang, Y.; Xu, C.; and Wang, Y. 2021. CMT: Convolutional Neural Networks Meet Vision Transformers. *arXiv preprint arXiv:2107.06263*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231.

Jiang, B.; Wang, M.; Gan, W.; Wu, W.; and Yan, J. 2019. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of International Conference on Computer Vision*, 2000–2009.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of Computer Vision and Pattern Recognition*, 1725–1732.

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

Kwon, H.; Kim, M.; Kwak, S.; and Cho, M. 2020. Motionsqueeze: Neural motion feature learning for video understanding. In *Proceedings of European Conference on Computer Vision*, 345–362.

Li, X.; Wang, Y.; Zhou, Z.; and Qiao, Y. 2020a. Smallbignet: Integrating core and contextual views for video classification. In *Proceedings of Computer Vision and Pattern Recognition*, 1092–1101.

Li, X.; Zhang, Y.; Liu, C.; Shuai, B.; Zhu, Y.; Brattoli, B.; Chen, H.; Marsic, I.; and Tighe, J. 2021. VidTr: Video Transformer Without Convolutions. *arXiv preprint arXiv:2104.11746*.

Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; and Wang, L. 2020b. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of Computer Vision and Pattern Recognition*, 909–918.

Lin, J.; Gan, C.; and Han, S. 2019. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of International Conference on Computer Vision*, 7083–7093.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021a. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv preprint arXiv:2103.14030*.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2021b. Video Swin Transformer. *arXiv preprint arXiv:2106.13230*.

Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of Neural Information Processing Systems*, 568–576.

Sun, L.; Jia, K.; Yeung, D.-Y.; and Shi, B. E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of International Conference on Computer Vision*, 4597–4605.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *Proceedings of International Conference on Machine Learning*, 10347–10357.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of International Conference on Computer Vision*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; and Feiszli, M. 2019. Video classification with channel-separated convolutional networks. In *Proceedings of International Conference on Computer Vision*, 5552–5561.

Wang, H.; Tran, D.; Torresani, L.; and Feiszli, M. 2020. Video modeling with correlation networks. In *Proceedings of Computer Vision and Pattern Recognition*, 352–361.

Wang, L.; Tong, Z.; Ji, B.; and Wu, G. 2021. TDN: Temporal difference networks for efficient action recognition. In *Proceedings of Computer Vision and Pattern Recognition*, 1895–1904.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of Computer Vision and Pattern Recognition*, 7794–7803.

Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; and Zhang, L. 2021. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of European Conference on Computer Vision*, 305–321.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.; Tay, F. E.; Feng, J.; and Yan, S. 2021a. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*.

Yuan, L.; Hou, Q.; Jiang, Z.; Feng, J.; and Yan, S. 2021b. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018. Temporal relational reasoning in videos. In *Proceedings of European Conference on Computer Vision*, 803–818.