CROSS-STAGE TRANSFORMER FOR VIDEO LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer network has been proved efficient in modeling long-range dependencies in video learning. However, videos contain rich contextual information in both spatial and temporal dimensions, *e.g.*, scenes and temporal reasoning. In traditional transformer networks, stacked transformer blocks work in a sequential and independent way, which may lead to the inefficient propagation of such contextual information. To address this problem, we propose a cross-stage transformer paradigm, which allows to fuse self-attentions and features from different blocks. By inserting the proposed cross-stage mechanism in existing spatial and temporal transformer blocks, we build a separable transformer network for video learning based on ViT structure, in which self-attentions and features are progressively aggregated from one block to the next. Extensive experiments show that our approach outperforms existing ViT based video transformer approaches with the same pre-training dataset on mainstream video action recognition datasets of Kinetics-400 (Top-1 accuracy **81.8%**) and Kinetics-600 (Top-1 accuracy **84.0%**). Due to the effectiveness of cross-stage transformer, our proposed method achieves comparable performance with other ViT based approaches with much lower computation cost ($e.g.$, **8.6%** of ViViT's FLOPs) in inference process. As an independent module, our proposed method can be conveniently added on other video transformer frameworks.

1 INTRODUCTION

Convolution neural network (CNN) has been successfully applied on computer vision tasks, such as classification (Krizhevsky et al. (2012); He et al. (2016)), detection (Girshick (2015); Ren et al. (2015)) and segmentation (He et al. (2017)). However, due to the limited receptive field, CNN lacks the ability of modeling long-range dependencies, which is an obstacle to capture the spatial and temporal contexts in video learning. To overcome this weakness, self-attention mechanism is introduced into CNN structure and obtains excellent performance (Wang et al. (2018); Guo et al. (2021)). Recently, convolution-free transformer structure consisting of self-attention layers (Vaswani et al. (2017)) is also investigated in vision domain (Dosovitskiy et al. (2020); Carion et al. (2020)).

Transformer achieved extreme success in natural language processing (NLP) (Vaswani et al. (2017); Devlin et al. (2018); Yang et al. (2019); Dai et al. (2019)). The inherent similar requirement between video and language learning, $i.e.$, capturing the long-range contextual information, makes people believe that it can also work for video tasks. The first attempt to apply pure transformer network for vision is Vision Transformer (ViT) (Dosovitskiy et al. (2020)), which aims at image classification. The input images are split into several patches, which are then linearly embedded into tokens for the transformer blocks. A classification head is attached at the top of these transformer blocks for final prediction. Bertasius $et\ al.$ (Bertasius et al. (2021)) and Arnab $et\ al.$ (Arnab et al. (2021)) extend the scheme to video learning by adding temporal transformer blocks.

Pure transformer network shows comparable performance with CNN based methods, as well as the potential in vision domain. However, there are still uncertainties by processing video data in the way analogous to language. On one hand, video patches contain rich spatial and temporal contents, so that it is difficult to map them into precise semantic tokens like words. Thus, the correlations established by transformer blocks may lead to ambiguous semantics. This drawback becomes even worse for videos with complex scenes and actions. On the other hand, the absence of convolutions in a transformer network will damage local contexts capturing, so that the features built across transformer blocks may have inefficient information propagation.

To tackle the aforementioned problems, we try to re-design the transformer blocks. Inspired by the empirically long-standing principle in CNN based approaches, $i.e.$, features extracted from different stages can be fused together to improve learning (Lin et al. (2017a); He et al. (2017); Lin et al. (2017b); Redmon & Farhadi (2018)), we expect the cross-stage fusion can also help improve the performance of transformer.

Based on above analysis, we propose a novel cross-stage transformer block which consists of **c**ross-**s**tage **s**elf-**a**ttention (CSSA) and cross-stage **f**eature **a**ggregation **m**odule (FAM). The former aims to progressively enhance the self-attention maps by adding shortcuts between self-attentions from two consecutive transformer blocks. The later fuses the features from different stages to achieve better outputs. We then build up a separable spatial-temporal transformer network, in which spatial cross-stage transformer and temporal cross-stage transformer are sequentially stacked. Extensive experiments show that, under the same conditions, *i.e.*, base transformer structure and pre-training dataset, our approach outperforms existing ViT based video transformers on video action recognition tasks. Due to the effectiveness of cross-stage fusion, our method can achieve comparable performance to ViViT (Arnab et al. (2021)) with much fewer FLOPs in inference process. As a generic module, the cross-stage transformer can also be inserted into other transformer based frameworks.

The contributions of this work can be summarized as follows:

1. A novel cross-stage transformer block, consisting of cross-stage self-attention module and cross-stage feature aggregation module, is proposed. Meanwhile, we also establish a separable cross-stage transformer network for video learning.
2. Extensive experiments are conducted to provide sufficient information for better understanding our approach, thereby provide an insight into the design of transformer in video learning.
3. Using the same pre-training dataset as existing transformer methods, our approach outperforms other ViT based video transformers and CNN methods on video action recognition datasets, including Kinetics-400 and Kinetics-600. It can also be added in other frameworks to promote the performance.

2 Related Work

Video action recognition. Extensive efforts have been put on video action recognition in recent years. The mainstream approaches usually utilize 2D or 3D based CNN for video feature extraction (Carreira & Zisserman (2017); Christoph & Pinz (2016); Tran et al. (2015); Ji et al. (2012); Tran et al. (2018); Simonyan & Zisserman (2014); Wang et al. (2016)). I3D (Carreira & Zisserman (2017)) is a representative of 3D based methods, which inflates 2D convolution layers into 3D to save the huge computational cost in pre-training 3D networks. Non-Local Neural Networks (Wang et al. (2018)) introduces self-attention into CNN, which can capture long-range dependencies and richer information of input video frames. Guo et al. (2021) proposes a separable self-attention network and achieve excellent performance on video action recognition. SlowFast (Feichtenhofer et al. (2019)) proposes a two-pathway network, using slow and fast temporal rates of video frames at the same time, in which features are fused from fast pathway into slow one. X3D (Feichtenhofer (2020)) explores different network settings based on SlowFast, and significantly boosts the performance. Recently, the research efforts are shifting to transformer based methods.

Image transformer networks. Self-attention network (Vaswani et al. (2017)), also known as transformer, has achieved state-of-the-art performance in NLP domain (Vaswani et al. (2017); Devlin et al. (2018); Yang et al. (2019); Dai et al. (2019)). This success inspires more and more research efforts on applying transformer to computer vision tasks. ViT (Dosovitskiy et al. (2020)) and DeiT (Touvron et al. (2020)) successfully show that pure transformer network can achieve state-of-the-art performance in image classification task. In Carion et al. (2020), a transformer-based network is proposed for object detection, and obtains comparative performance with Faster-RCNN (Ren et al. (2015)). SETR (Zheng et al. (2020)) proposes segmentation transformer network, which achieves desirable performance in semantic segmentation. Wu et al. (2021); Li et al. (2021) incorporated convolution design into transformer network by adding locally inductive biases. Swin Transformer (Liu et al. (2021)) proposes a hierarchical transformer structure to flexibly model feature representation at various scales. These work showcases the potential of transformer in vision domain.

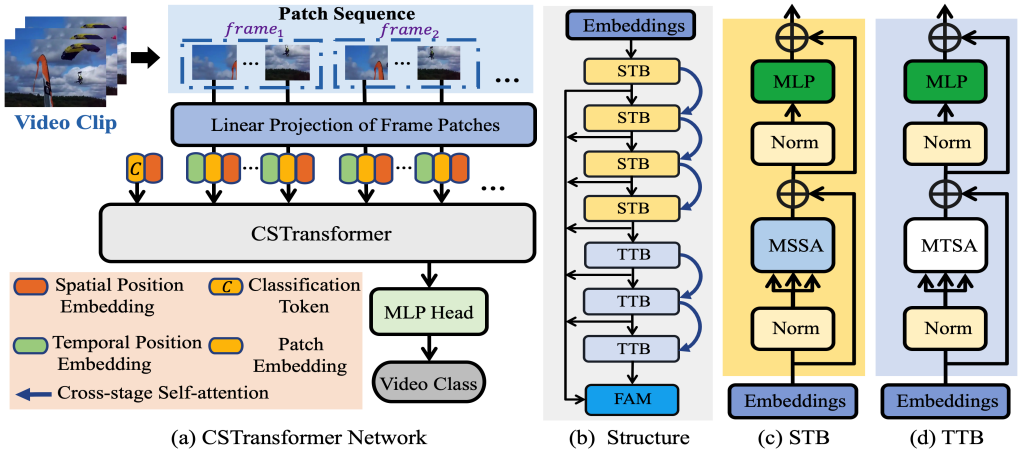

Figure 1: **The illustration of cross-stage transformer network.** (a) The framework of cross-stage transformer for video learning. (b) Cross-stage transformer structure. STB indicates spatial transformer block, and TTB indicates temporal transformer block. Blue arrows represent the direction of the cross-stage self-attention. The self-attention from each block is fused with that from the next one. The features from different blocks are aggregated together in cross-stage feature aggregation module (FAM). Note that we only use 4 $\times$ STB and 3 $\times$ TTB for simplicity. (c) STB. (d) TTB. MSSA and MTSA denote multi-head spatial and temporal self-attention respectively.

Video transformer networks. With the achievements of transformer in image domain, there also appears transformer networks for video. VTN (Neimark et al. (2021)) proposed a generic framework for video recognition, which consists of a 2D spatial backbone for feature extraction, a temporal attention-based encoder for modeling temporal dependencies of the spatial features, and a MLP head for classification. Timesformer (Bertasius et al. (2021)) adapted image transformer (Dosovitskiy et al. (2020)) architecture to video, and proposed several different self-attention schemes for transformer network design. STAM (Sharir et al. (2021)) presented a spatial-temporal transformer network, which processes sampled frames by a spatial transformer and a temporal transformer sequentially. ViViT (Arnab et al. (2021)) also proposed a pure-transformer architecture for video classification, and developed several variants, which can separate transformer's self-attention along spatial and temporal dimensions. There are also some works on adding shortcuts between transformer blocks in NLP and image domains to evolve the features (Wang et al. (2021); He et al. (2020)). Our work is inspired by these works but more challenging. Since video learning need to capture more complex information from spatial and temporal dimensions, simple shortcuts cannot efficiently work in existing video transformers.

3 Proposed Method

In Sec. 3.1, we introduce the video learning process of cross-stage transformer network. Then in Sec. 3.2, we explain the proposed cross-stage self-attention (CSSA) in details. Finally in Sec. 3.3, the cross-stage feature aggregation module (FAM) is described.

3.1 Cross-stage Transformer Network

The **c**ross-**s**tage **t**ransformer (CSTransformer) network is illustrated in Figure 1. We explain each component of the workflow as follows.

Input video clips. We employ ViT (Dosovitskiy et al. (2020)) as our baseline by extending transformer blocks to temporal dimension. Then we build up video learning network by adding cross-stage attention and feature fusion. Let $X \in \mathbb{R}^{B\times C\times T\times H\times W}$ be the input video clip. B denotes batch size. C denotes the number of input channels. T represents the length of the clip. W and H denote width and height of input frames respectively. We use constant W and H in our experiments.

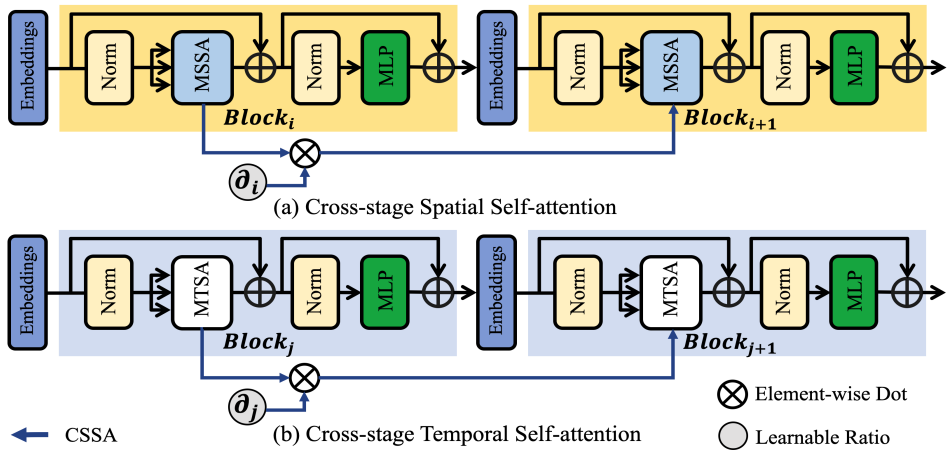

Figure 2: **The illustration of cross-stage self-attention (CSSA).** (a) Cross-stage spatial self-attention. (b) Cross-stage temporal self-attention. For simplicity, we only show cross-stage self-attention flow of two consecutive transformer blocks.

Patch embedding. In order to convert input frames into spatial patches, we firstly reshape X as $X \in \mathbb{R}^{(B\times T)\times C\times H\times W}$, then split X into P non-overlapped patches. The size of each patch is $M \times M$, and P is $(H \times W)/M^2$. A linear layer is employed to change the channels of each patch, after which the shape of the embedding is $V \in \mathbb{R}^{(B\times T)\times C'\times \frac{H}{M}\times \frac{W}{M}}$, where C' represents channel dimension after the linear layer. After that, we will flatten embedding V along spatial dimensions and transpose the last two dimensions, resulting in the shape of embedding $V \in \mathbb{R}^{(B\times T)\times P\times C'}$.

Classification token. After converting X into patch embedding V, we initialize a classification token $V_{cls} \in \mathbb{R}^{1\times 1\times C'}$ as 0, and repeat the classification token V_{cls} among the first dimension of V, *i.e.*, $V_{cls} \in \mathbb{R}^{(B\times T)\times 1\times C'}$.

Position encoding. In this step, spatial position embedding is firstly added into classification token V_{cls}. This operation is formulated in equation (2), where $P_s \in \mathbb{R}^{1\times(1+P)\times C'}$ denotes spatial position embedding. In equation (1), $P_s^{cls} \in \mathbb{R}^{1\times 1\times C'}$ and $P_s^V \in \mathbb{R}^{1\times P\times C'}$ are used to update classification token V_{cls} and token V respectively, $Concat$ means concatenation operation.

$$P_s = Concat[P_s^{cls}, P_s^V] \tag{1}$$

$$V_{cls} = V_{cls} + P_s^{cls} \tag{2}$$

Through equation (2), spatial position information can be combined with classification token V_{cls}. Since video clips contain temporal correlations, we also introduce temporal position embedding $P_t \in \mathbb{R}^{T\times 1\times C'}$ together with V and P_s. Spatial and temporal position information are fused into patch embedding V through equation (3). Finally, as shown in equation (4), classification token V_{cls} will be appended into patch embedding V to form $V_0 \in \mathbb{R}^{(B\times T)\times(1+P)\times C'}$, and V_0 will be fed into cross-stage transformer as input embedding sequence.

$$V = V + P_s^V + P_t \tag{3}$$

$$V_0 = Concat[V_{cls}, V] \tag{4}$$

Cross-stage structure. Our proposed method consists of several **s**patial **t**ransformer **b**locks (STBs) and **t**emporal **t**ransformer **B**locks (TTBs). STB/TTB consists of of layer normalisation (LN) (Ba et al. (2016)), **m**ulti-head **s**patial **s**elf-**a**ttention (MSSA)/**m**ulti-head **t**emporal **s**elf-**a**ttention (MTSA)

and MLP blocks. MSSA is used to compute self-attention of spatial patches within each frame to handle the relationship between objects and scenes, which is similar to MSA in ViT (Dosovitskiy et al. (2020)). While MTSA mainly focuses on computing self-attention of co-located patches along temporal dimension, so that temporal relationships of frames can be captured. Note that input shapes for MSSA and MTSA will be reshaped as $\mathbb{R}^{(B\times T)\times(1+P)\times C'}$ and $\mathbb{R}^{(B\times(1+P))\times T\times C'}$ respectively. The operations of STB and TTB are formulated in equation (5) and (6), where L represents the total blocks of cross-stage transformer. V_i is the output of ith transformer block. MSA represents multi-head self-attention process, which covers MSSA for spatial transformer blocks and MTSA for temporal transformer blocks. And MLP contains two linear layers with a GELU non-linearity.

$$V_i' = MSA(LN(V_{i-1})) + V_{i-1}, i = 1, ..., L \tag{5}$$

$$V_i = MLP(LN(V_i')) + V_i', i = 1, ..., L \tag{6}$$

Features from different spatial/temporal transformer blocks will then go through FAM for cross-stage fusion, as described in equation (7), where Y is the aggregated feature. The details of cross-stage self-attention and feature aggregation will be clarified in Sec. 3.2 and Sec. 3.3.

$$Y = FAM(V_i), i = 1, ..., L \tag{7}$$

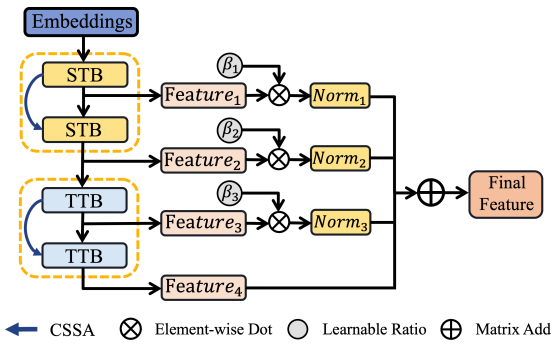

Figure 3: The illustration of cross-stage feature aggregation module (FAM).

MLPs in STB and TTB. MLPs in ViT (Dosovitskiy et al. (2020)) usually contain two fully connected (*fc*) layers. Let d denotes the dimension of input feature. The first *fc* layer will expand the dimension d into $4 \times d$. Our STB follows this style, while TTB keeps the original dimension. We find that this design can achieve better accuracy and computation trade-off. Experiments of different configurations are summarized in table 1d. In the second *fc* layer, the dimension is changed back to d.

MLP head for classification. Finally, the aggregated feature Y from FAM will go through a MLP head consisting of a LN layer and a linear layer for final video class prediction.

3.2 Cross-stage Self-attention

The proposed cross-stage self-attention (CSSA) approach is simple yet efficient. The purpose of this design is to progressively fuse the self-attention from different stages to achieve better attention maps. As shown in Figure 2, the self-attention map from each STB/TTB will firstly perform an element-wise multiplication with a corresponding learnable ratio α, which can dynamically adjust the scale of corresponding self-attention. Then the scaled self-attention will be added to the self-attention from the next stage. The whole process can be defined as Equation 8:

$$CrossA_i = Softmax(A_i + \alpha_i \cdot A_{i-1}), i = 1, ..., L \tag{8}$$

where $CrossA_i$ and A_i represent cross-stage self-attention and self-attention of ith transformer block respectively. When i equals 1, cross-stage self-attention is original self-attention A_1. α_i is the learnable ratio of ith transformer block and $(\cdot)$ is element-wise dot product. Note that A_i is the pairwise similarity derived by the multiplication of query matrix and key matrix. $CrossA_i$ is then used to multiply with value matrix as output. Our experiments demonstrate the effectiveness of this module in both objective and subjective measurements.

3.3 CROSS-STAGE FEATURE AGGREGATION

Cross-stage feature aggregation module (FAM) provides a global path for the features from different stages to better capture contextual information. The details of FAM are shown in Figure 3. We only use 4 transformer blocks for illustration. Specially, the feature from each transformer block will multiply a corresponding learnable parameter with an element-wise dot product. This parameter can scale its input feature globally. The scaled output will then be fed into a norm layer. Here we use LN layer for normalization. After that, all normalized results will be fused together for the output of FAM. It's noteworthy that our proposed cross-stage transformer block can be easily implemented by introducing only a few additional parameters, which is negligible in complexity. The fusion process is as follows:

$$Y = \sum LN(\beta_i \cdot V_i) + V_L, i = 1, ..., L-1 \tag{9}$$

where β_i is the learnable ratio for ith transformer block and Y is the aggregated feature.

4 EXPERIMENTS

In this section, we clarify relevant experimental settings and evaluate our proposed approach in several datasets to validate its effectiveness. We introduce the datasets for evaluation in Sec. 4.1. Then we show the implementation details of our approach in Sec. 4.2. Extensive ablation studies are conducted for fully understanding the proposed approach in Sec. 4.3. In Sec. 4.4, we visualize the self-attention maps from our approach and the baseline to better understand the efficiency of cross-stage transformer. Finally, we compare our approach with other state-of-the-art methods in Sec. 4.5.

4.1 DATASETS

We evaluate our approach on two large-scale video action recognition datasets, $i.e.$, Kinetics-400 (Kay et al. (2017)) and Kinetics-600 (Carreira et al. (2018)). The details of the datasets are described below.

Kinetics-400 dataset. The kinetics-400 dataset consists of training, validation and testing splits. Specifically, it contains 246536 training videos and 19761 validation videos, there are 400 human action categories, which are extracted from original YouTube videos. However, due to expired of Youtube links, there are only 234584 videos of training split. Videos in Kinetics are relatively longer and more complex, which are trimmed to around 10 seconds.

Kinetics-600 dataset. The kinetics-600 dataset follows the same style of Kinetics-400 dataset, except that it extend 400 categories into 600, and the training split consists of 366,016 videos. We also use training and validation splits for training model and evaluation.

4.2 IMPLEMENTATION DETAILS

Network structure. For all experiments, we adopt "Base" architecture of ViT model (Dosovitskiy et al. (2020)) with temporal extension as our baseline, which is trained in ImageNet dataset (Krizhevsky et al. (2012)). For fair comparison, we only include the approaches using the same pre-training dataset ($i.e.$, ImageNet-21K (Deng et al. (2009))). The structure of transformer layers in ViT-Base is the same as STB in CSTransformer network. We vary the numbers of STB and TTB, then evaluate these variants to showcase the impact of layers on our design. Top-1 accuracy of these variants are reported in table 1a, $i.e.$, CSTransformer-V1, CSTransformer-V2 and CSTransformer-V3, which will be explained in details in ablation study part.

Data processing. In our experiments, we sample 8, 16 and 32 frames with temporal stride of 32, 16 and 8 respectively as input clips. The sampled input clips will be processed by color normalization, random scale jittering and uniform crop. The scale jittering range is [256, 320] and the uniform crop will slice frames into 3 spatial crops (top left, center and bottom right) of size 224×224. The patch size is 16×16.

Model	# of STB	# of TTB	GFLOPs	Top-1 Accuracy (%)
CSTransformer-V1	10	10	345.2	76.3
CSTransformer-V2	12	6	**339.6**	78.7
CSTransformer-V3	12	8	365.6	**78.9**

(a) Comparison of different model variants

Spatial	Temporal	Top-1 Accuracy (%)
		77.6
✓		78.4
✓	✓	**78.7**

(b) The contribution of position encoding

8	16	32	Top-1 Accuracy (%)
✓			78.7
	✓		80.1
		✓	**81.2**

(c) Different input clip lengths

Model	1 ×	2 ×	4 ×	GFLOPs	Top-1 Accuracy (%)
CSTransformer-V2	✓			**339.6**	78.7
CSTransformer-V2		✓		359.9	78.8
CSTransformer-V2			✓	400.7	**79.0**

(d) Comparison of different MLP configurations in TTB

Table 1: Ablation study for model variants, position encoding, clip lengths and MLP configurations.

Training details. For all experiments, we use 8 $\times$ NVIDIA V100 devices. Initial learning rate is 0.005, and total epoch is 18. We use SGD optimizer with weight decay of 10^{-4} and momentum of 0.9 for training. Learning rate drops 10 times at epoch 5, 14 and 16.

Inference settings. Whereas most existing methods use 10 temporal clips with 3 spatial crops (top-left, center and bottom-right) for inference, we only use 1 temporal clip (which is sampled in the middle of video clips) with 3 spatial crops for default setting. The final prediction is averaged softmax scores of all predictions.

4.3 Ablation Studies

In this section, we conduct various ablation studies on Kinetics-400, which can allow us to better understand different components' effects for CSTransformer.

Cross-stage transformer. In table 2b, we report the ablation study of the main components in cross-stage transformer, *i.e.*, cross-stage self-attention (CSSA) and feature aggregation module (FAM). We also show the result of baseline, which employs separable spatial and temporal transformer structure as depicted in Figure 1 (b) without cross-stage operations. From the table, we can see that both CSSA and FAM help improve the performance. When using them together, *i.e.*, the whole cross-stage transformer, the performance can be boosted from 77.8% to 78.7%.

Model variants. We stack different number of STB and TTB to form CSTransformer, $i.e.$, CSTransformer-V1, CSTransformer-V2 and CSTransformer-V3. The length of the input clips is 8. The detailed comparisons of various settings are shown in table 1a. Since CSTransformer-V2 structure has obtained optimal accuracy and computation trade-off, we employ it in other experiments.

Does positional encoding help? In order to further understand the performance of spatial and temporal position encoding for CSTransformer. We evaluate CSTransformer-V2 with input clip length of 8. The results of using position encoding can be seen in table 1b. We can observe that by adding spatial position embedding, model's performance has been improved from 77.6% to 78.4%. And introducing temporal embedding can further boost its performance into 78.7%.

The effect of input clip length. Different clip lengths can impact the performance of proposed approach. We compare three clip lengths, $i.e.$, 8, 16 and 32. The performance of CSTransformer-V2 is illustrated in table 1c. A reasonable result can be observed that model's performance increases as clip length becomes larger.

MLP in TTB. As mentioned before, in TTB, the first *fc* layer in MLP doesn't expand the original dimension, which is different from original ViT (Dosovitskiy et al. (2020)) design. We compare the performance of several expansion ratios, including 1 $\times$, 2 $\times$ and 4 $\times$ in table 1d, and find that

1×3	4×3	10×3	Top-1 Accuracy (%)
✓			78.7
	✓		79.3
		✓	**79.5**

(a) Influence of different inference settings

Baseline	CSSA	FAM	Top-1 Accuracy (%)
✓			77.8
✓	✓		78.3
✓	✓	✓	**78.7**

(b) Influence of different components

Table 2: Ablation study for inference settings and different components.

expanding the dimension in cross-stage transformer doesn't help. Therefore we keep the original dimension for TTB as in ViT.

How does inference views influence performance? In video learning experiments, one needs to sample $x \times y$ video clips for evaluation, where x and y denotes the number of temporal clips and spatial crops. We wonder how does this inference sampling strategy influence the model's performance. Therefore, we perform multiple inference settings including 1×3, 4×3 and 10×3. The results of CSTransformer-V2, which is trained with input clip length of 8, is shown in table 2a.

4.4 VISUALIZATION

To intuitively understand the proposed method, we visualize the self-attention map of cross-stage transformer in this section. Figure 4 shows the visualization for cross-stage transformer and the baseline in video clips from Kinetics-400. We can observe that the proposed approach can pay more attention to those areas such as hands and bee box, which are very important to understand the video contents. And it's also interesting to see that our approach shows much less attentions in non-relevant regions such as the background. We conjecture that cross-stage self-attention and feature aggregation can propagate important semantic information across different transformer blocks. As a result, those attentions for important areas can be gradually evolved and highlighted.

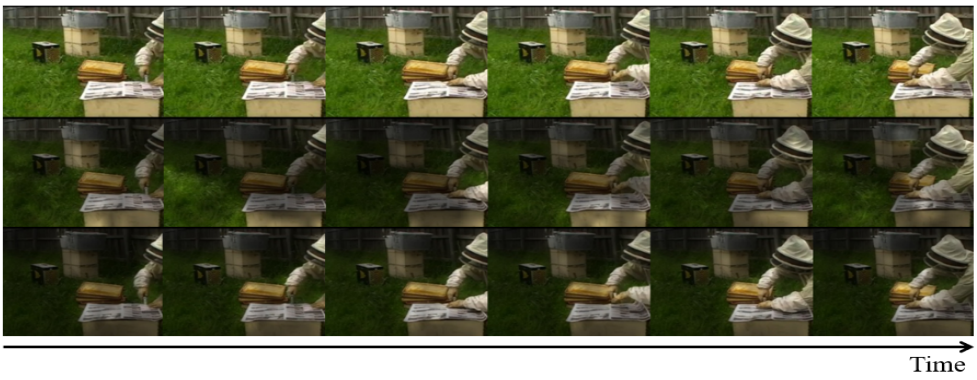

Figure 4: **The visualization of self-attention map from the output token of a video clip, namely "A person was keeping bees".** The top row is the original video clip. The second row is the result of the baseline without using cross-stage self-attention and feature. The third row shows the result of CSTransformer. Brighter area means more attention has been focused on.

4.5 COMPARISON WITH THE STATE-OF-THE-ART

In this section, we compare our method with several state-of-the-art approaches in terms of accuracy metrics and inference costs with total number of spatial and temporal views. We employ the input frame length of 32 in our evaluations. For fair comparison, we only report transformer methods' results using the same pre-training dataset, $i.e.$, ImageNet-21K. Moreover, since our method is implemented based on ViT structure, we only compare the video transformers based on ViT, $i.e.$, VTN (Neimark et al. (2021)), ViViT (Arnab et al. (2021)) and TimeSformer (Bertasius et al. (2021)), to demonstrate the effectiveness of our method more clearly. There are also other video transformers (Liu et al. (2021); Fan et al. (2021)) which are much different from original ViT structure and report

high performances. We will add our approach on these frameworks for comparison in the future. Note that ViT-L-ViViT with crop size 320 $\times$ 320 (the total inference cost is 3992 GFLOPs $\times$ 4 $\times$ 3 $\approx$ 47.9 TFLOPs) is compared in our experiment.

Kinetics-400 dataset. The comparison results on Kinetics-400 are shown in table 3. In addition to accuracy metrics, we also report inference views and inference cost in terms of TFLOPs. When the inference view is 1 $\times$ 3, our approach achieves 81.2% top-1 accuracy and 94.8% top-5 accuracy. With 4 $\times$ 3 views, CSTransformer outperforms existing CNN and ViT based transformer approaches. Our approach achieves comparable performance with ViT-L-ViViT by only 8.6 % of its inference cost, since we use less views (1$\times$3 vs. 4$\times$3) and layers (ViT-Base vs. ViT-Large).

Kinetics-600 dataset. We also evaluate our proposed approach on Kinetics-600. The results are shown in table 4. CSTransformer network achieves superior performance as well. Furthermore, Our approach consumes much less inference cost than other ViT based transformers (Bertasius et al. (2021); Arnab et al. (2021)) under the same inference views.

Method	Venue	Top-1	Top-5	Views	TFLOPs
I3D NL (Wang et al. (2018))	CVPR'18	77.7	93.3	10 $\times$ 3	10.8
LGD-3D R101 (Qiu et al. (2019))	CVPR'19	79.4	94.4	-	-
SlowFast R101-NL (Feichtenhofer et al. (2019))	CVPR'19	79.8	93.9	10 $\times$ 3	7.0
STM (Jiang et al. (2019))	ICCV'19	73.7	91.6	-	-
TSM-ResNeXt-101 (Lin et al. (2019))	ICCV'19	76.3	-	-	-
ip-CSN-152 (Tran et al. (2019))	ICCV'19	77.8	92.8	10 $\times$ 3	3.2
bLVNet (Fan et al. (2019))	NIPS'19	73.5	91.2	-	0.84
TEA (Li et al. (2020))	CVPR'20	76.1	92.5	10 $\times$ 3	-
CorrNet-101 (Wang et al. (2020a))	CVPR'20	79.2	-	10 $\times$ 3	6.7
X3D-XXL (Feichtenhofer (2020))	CVPR'20	80.4	94.6	10 $\times$ 3	5.8
ViT-B-VTN (Neimark et al. (2021))	Arxiv'21	78.6	93.7	10 $\times$ 3	4.2
TimeSformer-L (Bertasius et al. (2021))	ICML'21	80.7	94.7	1 $\times$ 3	7.1
ViT-L-ViViT (Arnab et al. (2021))	ICCV'21	81.3	94.7	4 $\times$ 3	47.9
Ours	-	81.2	94.8	1 $\times$ 3	4.1
Ours	-	**81.8**	**95.2**	4 $\times$ 3	16.4

Table 3: Comparison with existing methods on Kinetics-400 dataset.

Method	Venue	Top-1	Top-5	Views	TFLOPs
LGD-3D R101 (Qiu et al. (2019))	CVPR'19	81.5	95.6	-	-
SlowFast R101-NL (Feichtenhofer et al. (2019))	CVPR'19	81.8	95.1	10 $\times$ 3	7.0
AttentionNAS (Wang et al. (2020b))	ECCV'20	79.8	94.4	-	1.0
X3D-XL (Feichtenhofer (2020))	CVPR'20	81.9	95.5	10 $\times$ 3	1.5
TimeSformer-L (Bertasius et al. (2021))	ICML'21	82.2	95.5	1 $\times$ 3	7.1
ViT-L-ViViT (Arnab et al. (2021))	ICCV'21	83.0	95.7	4 $\times$ 3	47.9
Ours	-	83.5	95.8	1 $\times$ 3	4.1
Ours	-	**84.0**	**96.1**	4 $\times$ 3	16.4

Table 4: Comparison with existing methods on Kinetics-600 dataset.

5 CONCLUSION

In this paper, we propose a novel cross-stage transformer network for video learning, which can effectively learn video representations. In specific, we design a CSTransformer block which consists of cross-stage self-attention module (CSSA) and cross-stage feature aggregation module (FAM). We then build up a separable CSTransformer network, in which spatial CSTransformer blocks and temporal CSTransformer blocks are sequentially stacked. Extensive experiments show that our approach outperforms existing state-of-the-art CNN and ViT based transformer methods on video action recognition tasks. Due to the effectiveness of CSTransformer block, our method can achieve comparable performance to ViViT with much fewer inputs and FLOPs in inference process. Since our proposed CSSA and FCM act as independent modules, they can also be added on other video transformer frameworks.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

Nicolas Carion, 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, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

R Christoph and Feichtenhofer Axel Pinz. Spatiotemporal residual networks for video action recognition. *Advances in Neural Information Processing Systems*, pp. 3468–3476, 2016.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

Quanfu Fan, Chun-Fu Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. *arXiv preprint arXiv:1912.00869*, 2019.

Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 203–213, 2020.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6202–6211, 2019.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Xudong Guo, Xun Guo, and Yan Lu. Ssan: Separable self-attention network for video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12618–12627, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. *arXiv preprint arXiv:2012.11747*, 2020.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2000–2009, 2019.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 909–918, 2020.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7083–7093, 2019.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12056–12065, 2019.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *arXiv preprint arXiv:2103.13915*, 2021.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5552–5561, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 352–361, 2020a.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pp. 20–36. Springer, 2016.

Xiaofang Wang, Xuehan Xiong, Maxim Neumann, AJ Piergiovanni, Michael S Ryoo, Anelia Angelova, Kris M Kitani, and Wei Hua. Attentionnas: Spatiotemporal attention cell search for video classification. In *European Conference on Computer Vision*, pp. 449–465. Springer, 2020b.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Yujing Wang, Yaming Yang, Jiangang Bai, Mingliang Zhang, Jing Bai, Jing Yu, Ce Zhang, Gao Huang, and Yunhai Tong. Evolving attention with residual convolutions. *arXiv preprint arXiv:2102.12895*, 2021.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

A APPENDIX

A.1 CROSS-STAGE SELF-ATTENTION

In this section, we further clarify the principles of proposed cross-stage self-attention. Mainsteam multi-head self-attention is proposed in Vaswani et al. (2017), which has been adopted in transformer network. We can formulate the process as follows.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O, \tag{10}$$

where $head_j = Attention(QW_j^Q, KW_j^K, VW_j^V)(1 \leq j \leq h)$. h is the number of total heads. Q, K and V mean query, key and value matrices respectively. W^O is the linear projection for concatenation of multiple heads' outputs. W_j^Q, W_j^K, W_j^V are the linear projections of key, query and value matrices for jth head. Then attention function can be written as:

$$Attention(\hat{Q}, \hat{K}, \hat{V}) = Softmax(\frac{\hat{Q}\hat{K}^{\mathrm{T}}}{\sqrt{d_k}})\hat{V}, \tag{11}$$

in which $\hat{Q}$, $\hat{K}$ and $\hat{V}$ mean converted query, key, value matrices by linear projection. d_k denotes dimension of input $\hat{Q}$ and $\hat{K}$ matrices. The attention weight $\frac{\hat{Q}\hat{K}^{\mathrm{T}}}{\sqrt{d_k}}$ is the pairwise similarity between query and key matrices, which has been forwarded progressively in proposed CSTransformer structure.

$$Cross\_MultiHead(Q, K, V) = Concat(c\_head_1, ..., c\_head_h)W^O. \tag{12}$$

In equation (12), $c\_head_j = Cross\_Attention(QW_j^Q, KW_j^K, VW_j^V)$. Cross-stage self-attention of ith $(1 \leq i \leq n)$ transformer block is formulated in equation (13) and (14). n denotes total number of transformer blocks.

$$Cross\_Attention(\hat{Q}_i, \hat{K}_i, \hat{V}_i) = Softmax(A_i + \alpha_i * A_{i-1})\hat{V}_i, \tag{13}$$

$$A_i = \frac{\hat{Q}_i \hat{K}_i^{\mathrm{T}}}{\sqrt{d_k}}, \tag{14}$$

where $\hat{Q}_i$, $\hat{K}_i$, $\hat{V}_i$ are linearly projected query, key and value matrices of ith transformer block. $*$ means element-wise dot operation. a_i represents a learnable ratio of ith block. We adopt multi-head cross-stage self-attention, namely $Cross\_MultiHead(Q, K, V)$, for self-attention output. Note that A_i should have the same shape with A_{i-1}, otherwise, we will use $MultiHead(Q, K, V)$ as output. $A_0 = 0$.

A.2 CSTRANSFORMER STRUCTURE

To be more clear, we exlain details of CSTransformer structure. We adopt "ViT-Base" (Dosovitskiy et al. (2020)) as our baseline. Detailed settings of "CSTransformer-V1", "CSTransformer-V2" and "CSTransformer-V3" are shown in table 5. The embedding dimension is 768; The head number is 12; The MLP sizes of STB and TTB are 3072 and 768 respectively.

Model	# of STB	# of TTB	Embedding size	MLP$_{STB}$	MLP$_{TTB}$	Heads
CSTransformer-V1	10	10	768	3072	768	12
CSTransformer-V2	12	6	768	3072	768	12
CSTransformer-V3	12	8	768	3072	768	12

Table 5: Details of CSTransformer model variants.

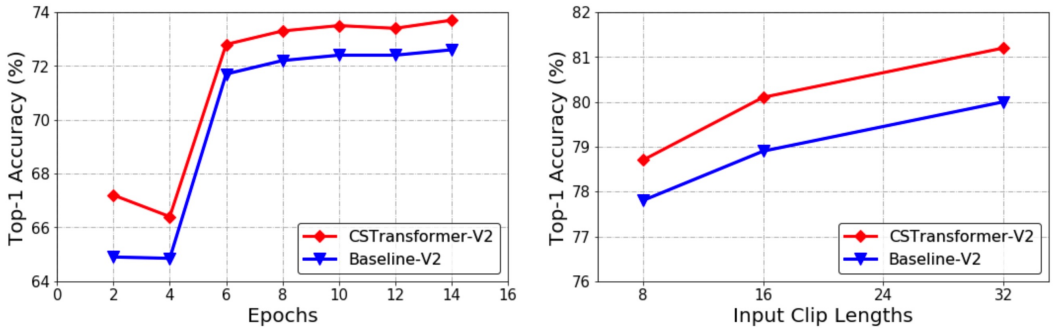

Figure 5: **Comparison results between Baseline-V2 and CSTransformer-V2.** The left represents top-1 accuracy of different epochs when input clip length is 8. The right denotes top-1 accuracy of different input clip lengths.

A.3 More Experimental Analysis

Here, we provide more experimental analysis and insights. The default dataset is Kinetics-400 (Kay et al. (2017)).

In order to further analyze the influence caused by proposed cross-stage self-attention and features. We also show the comparison results between baseline and CSTransformer, which can be seen in figure 5. "Baseline-V2" has the same structure of "CSTransformer-V2", except that it doesn't adopt cross-stage self-attention and features. Note that in left figure, we report top-1 accuracy on validation dataset in training process, and we only sample one clip for inference in different epochs. In right figure, we test the models with the view of (1×3) after training. As we can see, CSTransformer structure can consistently achieve higher performance than baseline when training epoch increases. Furthermore, even with different input clip lengths, CSTransformer structure also performs better than baseline model.

A.4 More Visualization Results

In this section, we provide more self-attention maps for visualization. Sampled frame clips are all from Kinetics-400 dataset.

Visualization for comparison. The choosed models are "Baseline-V2" and "CSTransformer-V2". As shown in Figure 6 and 7. The $1st$ row represents original frame clips. The $2nd$ and $3rd$ rows mean self-attention maps of "Baseline-V2" and "CSTransformer-V2". Note that brighter areas mean that more attention has been focused on. We can clearly observe that self-attention maps from CSTransformer structure can more focus on important objects and motion areas. However, self-attention maps from baseline may focus on some irrelevant regions.

Self-attention maps of CSTransformer. We show original video clips and their self-attention maps from proposed CSTransformer-V2 in figure 8, 9, and 10. The $1st$ and $2nd$ rows of each figure are original frame clips and self-attention maps of "CSTransformer-V2" respectively.

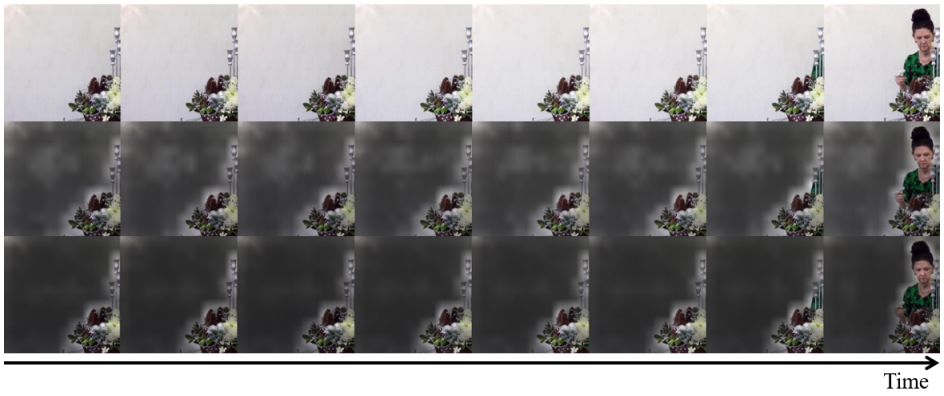

Figure 6: **A video clip, namely "A woman was arranging flowers".**

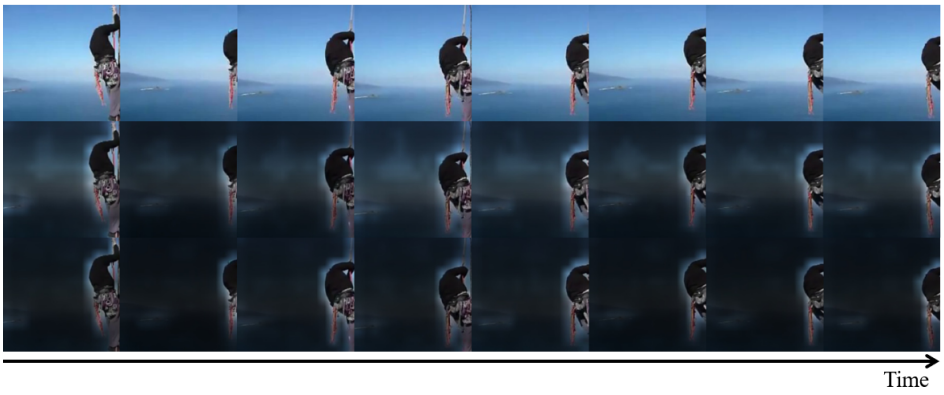

Figure 7: **A video clip, namely "A man was abseiling".**

Figure 8: **A video clip, namely "A person was biking through snow".**

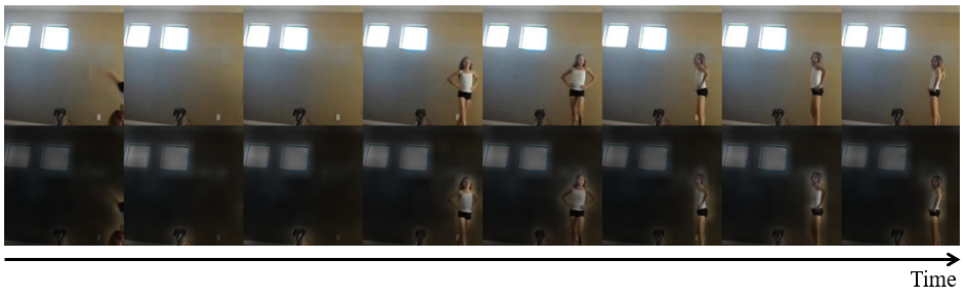

Figure 9: **A video clip, namely "A girl was bending back".**

Figure 10: **A video clip, namely "A woman was answering questions".**