Long Movie Clip Classification with State-Space Video Models

Md Mohaiminul Islam, Gedas Bertasius

UNC Chapel Hill

Abstract. Most modern video recognition models are designed to operate on short video clips (e.g., 5-10s in length). Thus, it is challenging to apply such models to long movie understanding tasks, which typically require sophisticated long-range temporal reasoning. The recently introduced video transformers partially address this issue by using long-range temporal self-attention. However, due to the quadratic cost of self-attention, such models are often costly and impractical to use. Instead, we propose ViS4mer, an efficient long-range video model that combines the strengths of self-attention and the recently introduced structured state-space sequence (S4) layer. Our model uses a standard Transformer encoder for short-range spatiotemporal feature extraction, and a multi-scale temporal S4 decoder for subsequent long-range temporal reasoning. By progressively reducing the spatiotemporal feature resolution and channel dimension at each decoder layer, ViS4mer learns complex long-range spatiotemporal dependencies in a video. Furthermore, ViS4mer is 2.63× faster and requires 8× less GPU memory than the corresponding pure self-attention-based model. Additionally, ViS4mer achieves state-of-the-art results in 6 out of 9 long-form movie video classification tasks on the Long Video Understanding (LVU) benchmark. Furthermore, we show that our approach successfully generalizes to other domains, achieving competitive results on the Breakfast and the COIN procedural activity datasets. The code is publicly available. [1]

1 Introduction

Suppose we ask someone to describe the relationship between the characters from the 'Interstellar' movie illustrated in Figure 1. It might be difficult for them to do so just by looking at a short video clip of several seconds. However, this is a much easier task if a person watches the whole movie. Thus, in this work, we pose the question of whether we can develop a computer vision model that can leverage long-range temporal cues to answer complex questions such as 'What is the genre of the movie?', 'What is the relationship among the characters?', 'Who is the director of the movie?', etc.

The majority of modern video recognition models [2, 5, 8, 14–16, 25, 36, 39, 43, 48] are unfortunately not equipped to solve these tasks as they are designed for short-range videos (e.g., 5-10 seconds in duration). Furthermore, extending these

[1] https://github.com/md-mohaiminul/ViS4mer

Fig. 1: Most traditional video models are designed for local prediction tasks (e.g., atomic action recognition, object detection, etc.) in short video clips (e.g., 5 seconds in length). In contrast, we aim to understand complex video understanding tasks in long movie videos (e.g., 200 seconds long), such as classifying the relationships among movie characters, predicting the writer of the story, categorizing the genre of the movie, etc.

models to the long-range video setting by stacking more input video frames is impractical due to excessive computational cost and GPU memory consumption.

Recently, several Transformer models [5,53] have been shown to perform well on long-range video understanding tasks. However, due to the quadratic cost of standard self-attention, these models are either very computationally costly [5], or they have to operate on pre-extracted CNN features [53], which discard fine-grained spatiotemporal information, thus, limiting the expressivity of the final video model. The latter characteristic is particularly important for long-range movie clip analysis, since fine-grained spatiotemporal cues may be indicative of the relationships between different movie characters, the genre of a movie, etc.

To address the efficiency-related issues of standard self-attention, recent work in Natural Language Processing (NLP) has proposed a structured state-space sequence model (S4) [19] for long-range sequence analysis. Unlike self-attention, the S4 layer has linear memory and computation cost with respect to the input length. As a result, it can handle much longer input sequences.

Combining the strengths of the self-attention and the S4 layer, we propose ViS4mer, a long-range **Vi**deo classification model composed of a standard transformer encoder and a multi-scale temporal **S4** decoder. The transformer encoder is used for spatial short-range video feature extraction whereas the S4 decoder performs long-range temporal reasoning, which is necessary for complex movie clip classification tasks. We build our temporal S4 decoder using a multi-scale architecture progressively reducing the number of tokens and the channel dimension with every layer. This allows our model to learn hierarchical spatiotemporal video representation while also reducing the computational cost associated with operating on a large number of video tokens.

We validate ViS4mer on the recently introduced Long Video Understanding (LVU) benchmark [53], which consists of nine diverse movie understanding tasks. We show that ViS4mer outperforms previous approaches in 6 out of 9 long-range video classification tasks. Moreover, compared to its self-attention counterpart, ViS4mer is 2.63× faster and requires 8× less GPU memory. Lastly, ViS4mer

generalizes to other domains, achieving competitive results on the Breakfast [30] and COIN [45] long-range procedural activity datasets.

2 Related Work

Modeling Long Sequences. Long sequence modeling is a fundamental task in Natural Language Processing (NLP). Previously, Bahdanau *et al.* [3] proposed a recurrent attention scheme for machine translation. Improving upon this work, Vaswani *at el.* [49] introduced a self-attention operation for the same machine translation task. Subsequently, a plethora of transformer-based architectures has been proposed for various NLP tasks [6, 10, 12, 34, 41, 55]. However, one major drawback of the transformer architecture is the quadratic complexity of standard self-attention. Various efficient self-attention schemes have been proposed for reducing the computation cost when modeling long sequences [9, 26, 29, 56]. Most relevant to our work, is the method of Gu *at el.* [17, 19, 20] that proposes an efficient structured state-space sequence (S4) layer for long sequence modeling. Inspired by this work, in this paper, we design a video architecture that incorporates the ideas from [19, 20] for long-range movie understanding tasks.

Video Recognition. Most existing video recognition methods are built using 2D and 3D Convolutional Neural Networks [8,15,16,25,31,40,43,48,52,58,60]. Due to the local nature of 2D and 3D convolutions, most of these models typically operate on short video clips of a few seconds. Inspired by the success of Transformer models in natural language processing (NLP), recently the transformer-based models have been successfully used for video recognition tasks [2, 5, 14, 36, 39]. However, due to the quadratic cost of the self-attention operation, these models are very computationally costly and, thus, only applied to short-range video segments. Our work aims to address this issue by proposing a novel efficient ViS4mer model for long movie clip understanding tasks.

Understanding Long-form Movie Videos. Movie understanding is a popular area of video understanding with many prior methods designed for movie-based tasks. Tapaswi *et al.* [47] introduce a movie question answering dataset. Bain *et al.* [4] and Zellers *at al.* [57] propose text-to-video retrieval and question answering benchmarks. However, these multi-modal benchmarks are often biased towards the language domain and are not ideal for evaluating video-only models. Several prior works introduced movie understanding datasets [22, 50, 54], which are not publicly accessible for copyright issues. Recently, Wu *et al.* [53] introduced a Long-form Video Understanding (LVU) benchmark that uses publicly available MovieClips [1]. The proposed LVU benchmark contains nine diverse tasks covering a wide range of aspects of long-form video understanding, which makes it suitable for evaluating our work as well. The current state-of-the-art Object Transformer method [53] applied on this benchmark, uses a Transformer architecture, and a variety of external modules (e.g., short-term video feature extractor [16,51], object detection, and tracking modules [21,32,42], and self-supervised pretraining) Instead, in this work, we propose ViS4mer, a simple and efficient long-range video recognition model.

Table 1: Theoretical runtime and memory requirement of state-space and self-attention operations w.r.t sequence length (L), batch size (B), and hidden dimension (H). Tildes denote log factors [19]. The runtime and memory cost of the state-space layer is linear w.r.t the input sequence length as opposed to the quadratic cost of self-attention.

	Self-attention	State-space
Parameters	H^2	H^2
Memory	$B(L^2 + HL)$	BLH
Training	$B(L^2H + LH^2)$	$BH(\tilde{H} + \tilde{L}) + B\tilde{L}H$
Inference	$L^2H + LH^2$	H^2

3 Background: Structured State-Space Sequence Model

Before describing our method, we first review some background information on Structured State-Spaces Sequence layers [19], which is one of the key components of our ViS4mer architecture. We start from the fundamental State-Space Model (SSM), which is defined by the simple equation (1). It maps a 1-dimensional input signal $u(t)$ to an N-dimensional latent space $x(t)$, then projects the hidden state $x(t)$ to a 1-dimensional output signal $y(t)$.

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

Here, A, B, C, and D are parameters learned using gradient descent. Unfortunately, the standard implementation of SSM can be very costly because computing the hidden state requires L successive multiplications with the matrix A. This results in $O(N^2L)$ operations and $O(NL)$ space for state dimension N and sequence length L. Moreover, this operation suffers from the vanishing/exploding gradients problem. To address this issue, the recent work [19] leverages HiPPO theory [18], which requires the A matrix to be defined as:

$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)1/2 & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases} \tag{2}$$

This provides theoretical guarantees allowing SSMs to capture long-range dependences in the sequential data. Building on this work, the method in [19] develops a structured state-space sequence (S4) layer, which significantly reduces the computation cost of a basic SSM.

In Table 1, we compare the theoretical time and space complexity of the self-attention and structured state-space sequence layers. We observe that self-attention has quadratic complexity w.r.t input sequence length L for training time, inference time, and memory requirement. In contrast, the state-space operation has linear time and space dependency w.r.t the input sequence length L. We refer the reader to the original paper [19] for further details.

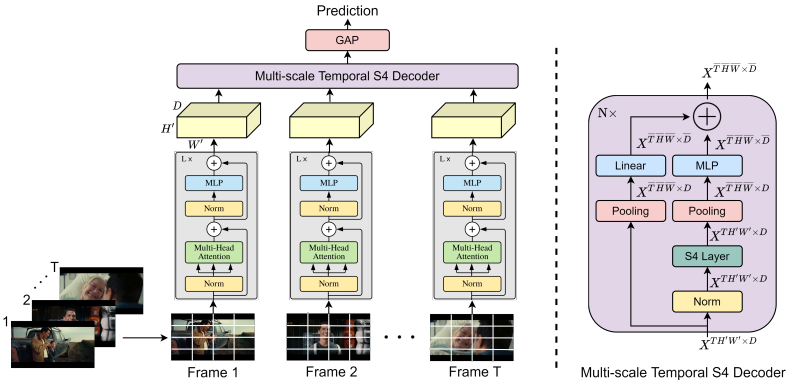


Fig. 2: Overview of the ViS4mer model. First, we split the video frames into fixed-size patches and use a short-range transformer encoder to extract contextual patch-level features for each video frame. Next, we feed the resulting spatiotemporal patch tokens from the whole video to a novel multi-scale temporal S4 decoder for modeling long-range temporal interactions in movie clips. Each S4 decoder block reduces the spatiotemporal resolution and the channel dimension using a Pooling and an MLP layer for learning multi-scale features. Afterward, the outputs from the S4 decoder are aggregated using global average pooling (GAP), and fed into the classification layer for the final downstream task prediction.

4 The ViS4mer Model

Our goal is to design a model for long-range movie clip analysis. To this end, we propose ViS4mer, a long-range video recognition model comprised of a transformer encoder and a multi-scale temporal S4 decoder. Following Vision Transformer [13], we first divide input video frames into smaller patches. We then apply a standard transformer encoder to extract fine-grained patch-level features from each video frame. Afterward, we use our proposed multi-scale temporal S4 decoder for long-range temporal reasoning over the patch-level features. Since the decoder has access to fine-grained spatiotemporal video patch information, it can effectively recognize the complex global properties of a long-range video. In Figure 2, we illustrate the overall architecture of our proposed ViS4mer model. Below, we also discuss each of these components in more detail.

4.1 Transformer Encoder

Let us assume, we have a video $V \in \mathbb{R}^{T \times H \times W \times 3}$ of T frames denoted by $(f_1, ..., f_i, ..., f_T)$. Each frame has a spatial resolution of $H \times W \times 3$, where H is the height, and W is the width of the frame. The transformer encoder $\mathscr{E}$ is then applied to each frame independently.

Following ViT [13], we divide each frame into N non-overlapping patches of size $P \times P$, where $N = HW/P$. Then a linear layer is applied to project each patch to a latent dimension of D, and a positional embedding $E \in \mathbb{R}^{N \times D}$ is added to each patch embedding. We can think of these embeddings as a sequence of N tokens $(z_1, ..., z_i, ..., z_N)$, where $z_i \in \mathbb{R}^D$.

The resulting sequence is passed through the transformer encoder $\mathscr{E}$ which is a stack of L transformer blocks. Each transformer block contains a multi-headed attention (MHA) and a multi-layer perceptron (MLP) block. Layer normalization (LN) is applied before each block, and a skip connection layer is added after each block. These operations can be expressed as:

$$\begin{aligned} \boldsymbol{z}' &= \text{MHA}(\text{LN}(\boldsymbol{z}_{in})) + \boldsymbol{z}_{in} \\ \boldsymbol{z}_{out} &= \text{MLP}(\text{LN}(\boldsymbol{z}')) + \boldsymbol{z}' \end{aligned} \tag{3}$$

The spatiotemporal token outputs of the transformer encoder can then be denoted as $(h_1, ..., h_i, ..., h_T)$, where $h_i = \mathscr{E}(f_i) \in \mathbb{R}^{H' \times W' \times D}$, $H' = H/P$, $W' = W/P$. All of these outputs are stacked into a single matrix $X \in \mathbb{R}^{T \times H' \times W' \times D}$.

4.2 Multi-scale Temporal S4 Decoder

Next, we describe our framework for modeling complex long-range temporal dependencies among the spatiotemporal tokens produced by the transformer encoder. One obvious choice is to use another transformer model to process such a sequence of tokens. However, this can be challenging due to (i) a large number of spatiotemporal tokens and (ii) the quadratic complexity of the self-attention operation. To illustrate this point, we note that processing a video of 60 frames of 224×224 spatial resolution using a patch size of 16×16 yields a total of $14 \times 14 \times 60 = 11,760$ output tokens. Using a standard self-attention operator on such a large number of tokens requires $\sim$138 million pairwise comparisons which is extremely costly. Another solution would be to only consider CLS token outputs for each frame. However, doing so removes fine-grained spatiotemporal information, thus, degrading the performance of long-range movie understanding tasks (as shown in our experimental evaluation).

To overcome this challenge we design a temporal multi-scale S4 decoder architecture for complex long-range reasoning. Instead of using self-attention, our decoder model utilizes the recently introduced S4 layer (described in Section 3). Since the S4 layer has a linear computation and memory dependency with respect to the sequence length, this significantly reduces the computational cost of processing such long sequences.

To effectively adapt the S4 layer to the visual domain, we design a multi-scale S4 decoder architecture. Our temporal multi-scale S4 decoder consists of multiple blocks where each block operates on different spatial resolution and channel dimensions. Starting from a high spatial resolution and high channel dimension, the proposed model gradually decreases spatiotemporal resolution and channel dimension at each block. Our multi-scale architecture is inspired by several successful multi-scale models in the visual domain such as Feature

Pyramid Networks [32], and Swin transformer [35]. Because of this multi-scale strategy, different blocks can effectively learn features at different scales, which helps the model to learn complex spatiotemporal dependencies over long videos. Furthermore, operating on shorter input sequences of smaller channel dimensions in the deeper blocks helps us to reduce overfitting on a relatively small LVU benchmark. Overall, as will be shown in our experimental section, in addition to producing better performance, our multi-scale strategy further reduces the computational cost and GPU memory requirements.

Figure 2 (right) shows the architecture of the multi-scale temporal S4 decoder. The decoder network $\mathscr{D}$ consists of N blocks, which are defined below:

$$\begin{aligned} \boldsymbol{x}_{s4} &= \text{S4}(\text{LN}(\boldsymbol{x}_{in})) \\ \boldsymbol{x}_{mlp} &= \text{MLP}(\text{Pooling}(\boldsymbol{x}_{s4})) \\ \boldsymbol{x}_{skip} &= \text{Linear}(\text{Pooling}(\boldsymbol{x}_{in})) \\ \boldsymbol{x}_{out} &= \boldsymbol{x}_{mlp} + \boldsymbol{x}_{skip} \end{aligned} \tag{4}$$

S4 Layer. We flatten the input tensor X to a sequence of L vectors $x_{in} = (x_1, ..., x_i, ..., x_L)$, where $L = T \times H' \times W'$, and $x_i \in \mathbb{R}^D$. We then pass this sequence to the S4 layer, which outputs the feature tensor $x_{s4} \in \mathbb{R}^{L \times D}$.

Spatiotemporal Resolution Reduction. We reduce the space-time resolution of our input tensor by a factor of $s_T \times s_H \times s_W$ using a max-pooling layer where $s_T \times s_H \times s_W$ is the stride along each axis of the input tensor. The resulting tensor has dimensionality of $\overline{T} \times \overline{H} \times \overline{W}$. This allows our model to learn multi-scale spatiotemporal representations while also reducing the computational cost of operating on long sequences.

Channel Reduction. After the pooling layer, we apply an MLP, which reduces the channel dimension of the input tensor. In addition to decreasing the computational cost, this also reduces overfitting on the LVU benchmark.

Skip Connections. We use skip connections from the input tensor to the final output of a decoder block. Due to the mismatch of feature dimensionalities, we apply an additional pooling layer to the input tensor. Furthermore, to handle the channel dimension mismatch, we use an additional linear layer.

4.3 Loss Functions

Following [53], we use a cross-entropy loss for the classification tasks, and the mean squared error (MSE) for the regressions tasks.

$$L_{ce}(\mathcal{F}_{\mathcal{C}}(\theta)) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{K} y_j^i \log(\mathcal{F}_{\mathcal{C}}(\theta; x^i)_j) \tag{5}$$

$$L_{mse}(\mathcal{F}_{\mathcal{R}}(\theta)) = -\frac{1}{B}\sum_{i=1}^{B}(y^i - \mathcal{F}_{\mathcal{R}}(\theta; x^i))^2 \tag{6}$$

Here, $\mathcal{F}_{\mathcal{C}}(\theta)$ and $\mathcal{F}_{\mathcal{R}}(\theta)$ are our classification and regression models respectively, B is the batch size, K is the number of classes (for the classification task), y is the label, x is the input, and θ are the learnable model parameters.

4.4 Implementation Details

We resize each video frame to the spatial resolution of 224×224 and use a patch size of 16×16. For the transformer encoder, we use a 24-block transformer with hidden dimension 1024 pretrained on ImageNet [11]. For our multi-scale temporal S4 decoder, we use a 3-block architecture. Each block has a pooling layer with a kernel of $1 \times 2 \times 2$, stride of $1 \times 2 \times 2$, and padding of $1 \times 1 \times 1$. As discussed above, each block also has an MLP layer, which reduces the feature dimension by a factor of $2\times$. For all of our experiments, we use Adam optimizer [28] with a learning rate of 10^{-3}, and with a weight decay of 0.01. We train our models using NVIDIA RTX A6000 GPU with a batch size of 16.

5 Experiments

We evaluate ViS4mer on the recently proposed Long-form Video Understanding (LVU) benchmark [53] which contains nine diverse tasks related to long-form movie understanding. Moreover, we also perform thorough ablation studies (i) comparing our S4-based model to an equivalent self-attention baseline, (ii) studying the efficiency of ViS4mer, (iii) comparing our method with other efficient attention schemes, (iv) analyzing the design choices of our model, (v) validating the robustness to different short-range encoders, and (vi) lastly investigating our model's long-range modeling capability. Additionally, to demonstrate the generalization of our approach, we also validate ViS4mer on two long-range procedural activity datasets, COIN [45, 46] and Breakfast [30].

5.1 Main Results on the LVU benchmark

The long-form video understanding benchmark (LVU) [53] is constructed using the publicly available MovieClip dataset [1], which contains ∼30K videos from ∼3K movies. Each video is typically one to three minutes long. The benchmark contains nine tasks covering a wide range of long-form video understanding tasks. These 9 tasks fall into three main categories: (i) **content understanding**, which consists of (*'relationship', 'speaking style', 'scene/place'*) prediction, (ii) **metadata prediction**, which includes (*'director', 'genre', 'writer', and 'movie release year'*) classification, and (iii) **user engagement**, which requires predicting (*'YouTube like ratio', and 'YouTube popularity'*).

The content understanding and the metadata prediction tasks are evaluated using the standard top-1 accuracy metric, whereas the user engagement prediction tasks are evaluated using mean-squared error (MSE). Following [53], we use standard splits and train our model using video clips of 60 seconds.

Table 2: Comparison to prior works on the LVU dataset. Compared to previous long-range video models (VideoBERT, and Object Transformer), ViS4mer achieves significantly better accuracy in most tasks. Furthermore, ViS4mer also outperforms our implemented Long Sequence Transformer baseline, which uses the same design as our model, except for the S4 layers, which are replaced with the self-attention layers.

	Sequence	Content (↑)			Metadata (↑)				User (↓)	
	Model	Relation	Speak	Scene	Director	Genre	Writer	Year	Like	Views
SlowFast+NL [16, 51]	non-local	52.40	35.80	54.70	44.90	53.00	36.30	**52.50**	0.38	3.77
VideoBERT [44]	self-attention	52.80	37.90	54.90	47.30	51.90	38.50	36.10	0.32	4.46
Obj. Transformer [53]	self-attention	53.10	39.40	56.90	51.20	54.60	34.50	39.10	**0.23**	**3.55**
Long Seq. Transformer	self-attention	52.38	37.31	62.79	56.07	52.70	42.26	39.16	0.31	3.83
ViS4mer	state-space	**57.14**	**40.79**	**67.44**	**62.61**	**54.71**	**48.8**	44.75	0.26	3.63

We compare our proposed ViS4mer model with the previous methods validated on this benchmark. In particular, we use the same baselines as in [53]. Additionally, we implement our own Long Sequence Transformer (**LST**) baseline, which follows exactly the same design as our ViS4mer model except for the S4 layers, which are replaced with standard self-attention layers.

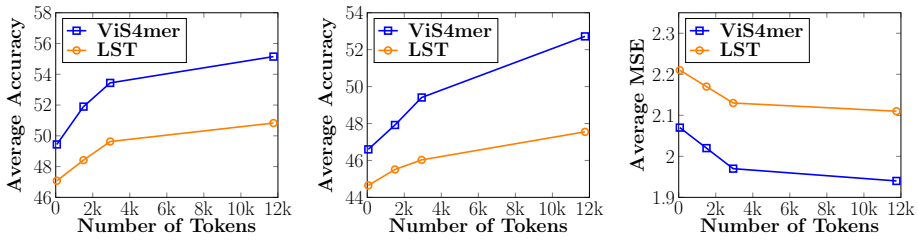
We present our results in Table 2 where we show that ViS4mer achieves state-of-the-art performance in most tasks. Specifically, ViS4mer outperforms both long-range video models (VideoBERT, and Object Transformer) in the content understanding and metadata prediction tasks by a significant margin and achieves comparable performance in the user engagement tasks. Furthermore, we also demonstrate that ViS4mer outperforms our Long Sequence Transformer baseline, which suggests the superiority of S4 layers over the standard self-attention layers for these tasks.

5.2 Ablation Studies

Detailed Comparison with Self-attention. Since most previous methods predominantly use self-attention for long sequence video modeling, we compare our state-space (i.e., S4) based design with the equivalent self-attention-based approaches. For these comparisons, we use the same Long Sequence Transformer baseline (described in the previous subsection), which replaces all state-space layers of the ViS4mer model with the self-attention layers.

Specifically, we compare the performance of ViS4mer and Long Sequence Transformer by varying the number of input tokens for both models. We use video clips of 60 seconds and a frame per second rate of 1. We vary the number of input tokens to $60, 1500, 2940$, and 11760 by applying spatial max-pooling with a varying stride individually on each frame before feeding the tokens into the model. Note that the 60 token baseline corresponds to a model that operates only on the frame-level CLS tokens.

We present our results for this study in Figure 3, where we plot (a) the average accuracy of three content understanding tasks, (b) the average accuracy of four metadata prediction tasks, and (c) the average MSE of two user engagement prediction tasks as a function of the number of input tokens. Based on



(a) Content Understanding ↑ (b) Metadata Prediction ↑ (c) User Engagement Prediction ↓

Fig. 3: We compare the performance of our ViS4mer and Long Sequence Transformer (LST) as a function of the number of input tokens on **(a)** the content understanding (using top-1 acc.), **(b)** the metadata prediction (using top-1 acc.), and **(c)** the user engagement prediction (using MSE) tasks. ViS4mer performs better for all number of tokens in all tasks.

Table 3: The GPU memory requirements (in GB) and the training speed (sample/seconds) of our state-space-based ViS4mer and the self-attention-based Long Sequence Transformer (LST). We compare both of these approaches while varying the number of input tokens. As we increase the number of tokens, the GPU memory and computation requirement of self-attention grows more rapidly for the LST baseline than for ViS4mer. Overall, ViS4mer requires 8× less GPU memory and is 2.63× times faster than the self-attention baseline while operating on very long video sequences (i.e., 11,760 spatiotemporal tokens).

# of Tokens	Samples/s (↑)		GPU Memory (GB)(↓)	
	ViS4mer	LST	ViS4mer	LST
60	**12.46**	8.85	**2.23**	2.45
1,500	**8.27**	6.31	**3.61**	3.99
2,940	**6.25**	4.47	**3.67**	5.43
11,760	**4.95**	1.88	**5.15**	41.38

these results, we observe that increasing the number of input tokens increases the performance of both ViS4mer and Long Sequence Transformer. We also note that ViS4mer achieves better performance than LST in all cases, which suggests that the state-space layers are superior to self-attention layers in this setting. Furthermore, our results indicate that the performance gap between the two methods increases as we increase the number of tokens. This observation suggests that the proposed ViS4mer architecture is more effective at incorporating information from very long video sequences.

Computational Cost Analysis. Additionally, in Table 3, we investigate the GPU memory requirements (in GB) and the training speed (sample/second) of ViS4mer and Long Sequence Transformer while varying the number of input tokens in the same way as was done in our previous analysis. These results indicate that in addition to being more accurate, ViS4mer is also significantly more memory-efficient and faster in all settings. It is worth mentioning, that

Table 4: Comparison with other efficient attention schemes. ViS4mer outperforms Performer [9] and Orthoformer [39] on the LVU benchmark while requiring similar memory and computation cost.

	Content (↑)			Metadata (↑)				User (↓)			
	Relation	Speak	Scene	Director	Genre	Writer	Year	Like	Views	Sam./s (↑)	Mem (↓)
Self-attention	52.38	37.31	62.79	56.07	52.70	42.26	39.16	0.31	3.83	1.88	41.38
Performer	50.00	38.80	60.46	58.87	49.45	48.21	41.25	0.31	3.93	4.67	5.93
Orthoformer	50.00	39.30	66.27	55.14	**55.79**	47.02	43.35	0.29	3.86	4.85	5.56
State-space	**57.14**	**40.79**	**67.44**	**62.61**	54.71	**48.8**	**44.75**	**0.26**	**3.63**	**4.95**	**5.15**

when increasing the number of tokens to a large number (11,760) the memory requirement of the self-attention-based model grows very rapidly, requiring 41.38GB of GPU memory. In contrast, ViS4mer is much more memory-efficient, requiring only 5.15GB of GPU memory. Moreover, based on these results, we observe that ViS4mer is 2.63× faster compared to Long Sequence Transformer when operating on sequences consisting of 11,760 spatiotemporal tokens.

Comparison with Other Efficient Attention Schemes. We also compare our method with other efficient self-attention schemes (e.g., Performer [9] and Orthoformer [39]) that do not require quadratic complexity with respect to the input length. We construct such models by replacing the state-space layers of the ViS4mer with the corresponding efficient self-attention layers and keeping all other settings the same as for our model. Table 4 shows the results of these comparisons. We can observe that ViS4mer achieves the best performance in most LVU benchmark tasks while requiring similar memory and computation cost as other efficient attention schemes (e.g., Performer and Orthoformer).

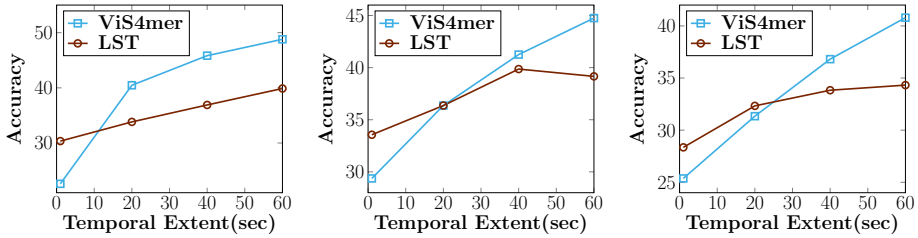
Table 5: Ablation on the ViS4mer architecture design. We observe that both (i) multi-scale feature learning (enabled by pooling) and (ii) progressive channel dimension reduction are critical for the best performance on the LVU benchmark. ViS4mer achieves substantially better performance compared to the vanilla S4 model while being 1.41× faster and requiring 2.2× less GPU memory.

Pooling	Scaling	Content(↑)	Metadata(↑)	User(↓)	Samples/s(↑)	Memory(GB)(↓)
✗	✗	49.53	49.26	2.30	2.25	7.27
✓	✗	48.96	49.77	2.10	3.98	5.96
✗	✓	52.25	48.79	2.09	4.12	5.95
✓	✓	**55.12**	**52.72**	**1.94**	**4.95**	**5.15**

ViS4mer Architecture Analysis. In Table 5, we analyze the significance of (i) multi-scale feature learning, which is enabled by the pooling layers, and (ii) the channel dimensionality reduction, which improves efficiency and reduces overfitting. These results indicate that both of these architecture design choices contribute not only to better performance on the LVU benchmark but also to the higher efficiency of ViS4mer. Specifically, compared to the vanilla S4 model, our final ViS4mer achieves 5.5%, and 3.5% better performance on the content

Table 6: Short-range encoder ablation. ViS4mer outperforms Object Transformer in 6 out of the 9 tasks while using the same short-range model (SlowFast [16]). Moreover, we observe that using ViT as our short-range encoder produced the best results in 6 out of 9 LVU benchmark tasks.

		Content (↑)			Metadata (↑)				User (↓)	
Model	Encoder	Relation	Speak	Scene	Director	Genre	Writer	Year	Like	Views
Obj. Trans. [53]	SlowFast [16]	53.10	39.40	56.90	51.20	54.60	34.50	39.10	**0.23**	**3.55**
	ViT [13]	54.76	33.17	52.94	47.66	52.74	36.30	37.76	0.30	3.68
ViS4mer	SlowFast [16]	**59.52**	40.29	60.46	53.27	52.74	42.85	39.86	0.27	3.70
	ConvNeXt [37]	**59.52**	38.30	62.79	57.00	54.40	45.83	42.65	0.30	3.74
	Swin [35]	54.76	37.31	61.62	56.07	49.45	47.61	39.86	0.31	3.56
	ViT [13]	57.14	**40.79**	**67.44**	**62.61**	**54.71**	**48.8**	**44.75**	0.26	3.63



(a) Writer Prediction. (b) Year Prediction. (c) Speaking Style Prediction.

Fig. 4: Performance on the **(a)** Writer Prediction, **(b)** Year Prediction, and **(c)** Speaking Style Prediction tasks as a function of the input video duration. Based on these results, we observe that the Long Sequence Transformer (LST) performs better on very short clips. However, ViS4mer excels on much longer clips indicating its effectiveness at modeling long video sequences.

understanding and metadata prediction tasks respectively, and 0.36 lower MSE on the user engagement tasks. Furthermore, ViS4mer has 2.2× faster run-time and 1.41× smaller GPU memory usage than the vanilla S4 model.

Short-range Encoder Ablation. To validate the robustness of our model, we also conduct experiments with different short-range encoder models. Specifically, we experiment with four popular short-range models, which includes both CNN and Transformer-based encoders: (i) SlowFast [16], which was used by the previous Object Transformer method [53], (ii) ConvNeXt [37], (iii) Swin Transformer [35], and (iv) ViT [13]. We report our results of this analysis in Table 6. Based on these results, we first note that ViS4mer outperforms Object Transformer in 6 tasks while using SlowFast [16], and 9 tasks while using ViT [13]. Furthermore, we observe that using ViT as our short-range encoder leads to the best performance in 6 out of 9 LVU benchmark tasks.

Temporal Extent Ablation. Additionally, we compare the long-range temporal reasoning abilities of our state-space-based ViS4mer and an equivalent self-attention-based Long Sequence Transformer. In particular, we train both of these models using video inputs spanning $1, 20, 40$, and 60 seconds. In Fig-

Table 7: Evaluation on two long-range procedural activity classification datasets, *i.e.*, COIN [45, 46], and Breakfast [30]. ViS4mer achieves comparable performance as the best performing, Distant Supervision [33] framework, while using significantly less pretraining data. This indicates ViS4mer's ability to generalize to other domains.

(a) Long-range procedural activity classification on the Breakfast [30] dataset.

Model	Pretraining Dataset	Pretraining Samples	Accuracy(↑)
VideoGraph [24]	Kinetics-400	306K	69.50
Timeception [23]	Kinetics-400	306K	71.30
GHRM [59]	Kinetics-400	306K	75.50
Distant Supervision [33]	HowTo100M	**136M**	**89.90**
ViS4mer	Kinetics-600	495K	88.17

(b) Long-range procedural activity classification on the COIN [45] dataset.

Model	Pretraining Dataset	Pretraining Samples	Accuracy(↑)
TSN [46]	Kinetics-400	306K	73.40
Distant Supervision [33]	HowTo100M	**136M**	**90.00**
ViS4mer	Kinetics-600	495K	88.41

ure 4, we illustrate our results on three LVU tasks (*i.e.*, Writer prediction, Year prediction, and Speaking style prediction). Our results suggest that while the self-attention-based approach performs better when applied on clips that span short temporal extent (*i.e.*, 1s in duration), the state-space model achieves much better performance when video inputs span long temporal extents (i.e., 40s and more). These results suggest that compared to self-attention, the state-space layers enable more effective long-range temporal reasoning.

5.3 Evaluation on Other Datasets

Lastly, to evaluate ViS4mer's ability to generalize to other domains, we conduct experiments on two long-range procedural activity classification datasets: Breakfast [30] and COIN [45, 46]. The Breakfast dataset contains 1,712 videos of 10 complex cooking activities. The average duration of the videos is 2.32 minutes. The COIN dataset consists of 11,827 videos capturing 180 diverse procedural tasks. The average length of a video is 2.36 minutes.

Given the long duration of the videos in both datasets, we believe that these datasets are well suited to test our model's ability for long-range activity understanding. For all of our experiments, we use standard splits [24, 46] and measure performance in terms of activity classification accuracy.

We report our results on these two datasets in Table 7. Based on these results, we observe that ViS4mer outperforms previous approaches pretrained on Kinetics [7, 27]. Moreover, ViS4mer achieves competitive performance as the state-of-the-art Distant Supervision method [33], which uses several orders of magnitude more pretraining data (i.e., HowTo100M [38]). Thus, we believe that these results provide sufficient evidence of ViS4mer's generalization ability.

5.4 Qualitative Results

In Figure 5, we also illustrate some qualitative results on the LVU dataset. In particular, we demonstrate several instances of the correct and incorrect predictions of our ViS4mer model on the relationship and genre prediction tasks. These results indicate that our method can effectively identify the relationships among the characters (Figure 5(a)) and the genre of the movie (Figure 5(c)). Furthermore, these qualitative examples highlight some movie instances which are difficult to classify even for a human (Figures 5(b), (d)), thus, illustrating the challenging nature of long-range movie understanding tasks.

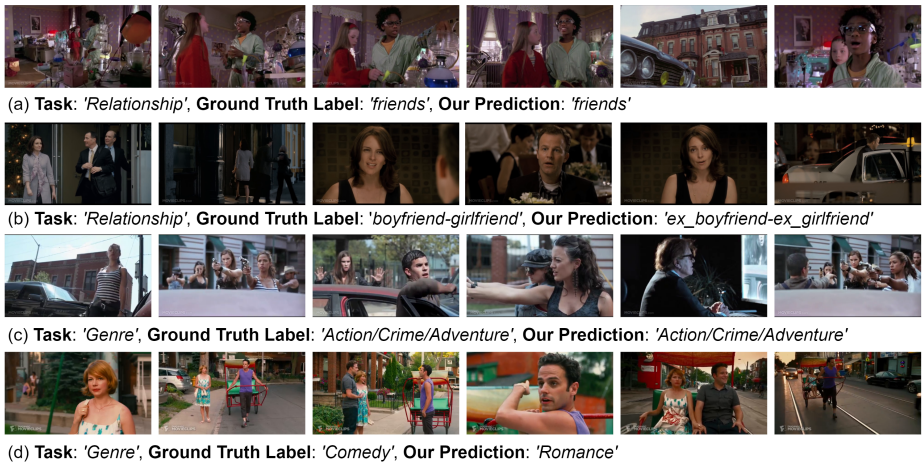


Fig. 5: Our qualitative results on the LVU dataset. ViS4mer can effectively identify **(a)** the relationship among the characters and **(c)** the genre of the movie. Furthermore, the complex examples shown in the rows **(b) (d)** illustrate the difficulties of long-range movie understanding tasks.

6 Conclusion

Combining the strength of self-attention and structured state-space sequence models, we introduce ViS4mer, an efficient framework for long-range movie video classification. Our method (i) is conceptually simple, (ii) achieves state-of-the-art results on several complex movie understanding tasks, (iii) has low memory requirement and computation cost, and (iv) successfully generalizes to other domains such as procedural activity classification. In the future, we plan to extend our work to other long-range video understanding tasks such as video summarization, question answering, and video grounding.

References

1. Movieclips. https://www.movieclips.com/.
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. arXiv preprint arXiv:2103.15691 (2021)
3. Bahdanau, D., Cho, K., et al.: Neural machine translation by jointly learning to align and translate. arxiv preprint arxiv: 1409.0473 (2014)
4. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. In: Proceedings of the Asian Conference on Computer Vision (2020)
5. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Proceedings of the International Conference on Machine Learning (ICML) (July 2021)
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
7. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018)
8. Carreira, J., Zisserman, A.: 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)
9. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al.: Rethinking attention with performers. arXiv preprint arXiv:2009.14794 (2020)
10. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. arXiv preprint arXiv:2104.11227 (2021)
15. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 203–213 (2020)
16. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)
17. Goel, K., Gu, A., Donahue, C., Ré, C.: It's raw! audio generation with state-space models. arXiv preprint arXiv:2202.09729 (2022)
18. Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C.: Hippo: Recurrent memory with optimal polynomial projections. Advances in Neural Information Processing Systems **33**, 1474–1487 (2020)

19. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)
20. Gu, A., Johnson, I., Goel, K., Saab, K.K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
22. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding. In: European Conference on Computer Vision. pp. 709–727. Springer (2020)
23. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 254–263 (2019)
24. Hussein, N., Gavves, E., Smeulders, A.W.: Videograph: Recognizing minutes-long human activities in videos. arXiv preprint arXiv:1905.05143 (2019)
25. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
26. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: International Conference on Machine Learning. pp. 5156–5165. PMLR (2020)
27. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Kitaev, N., Kaiser, Ł., Levskaya, A.: Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451 (2020)
30. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 780–787 (2014)
31. Li, X., Wang, Y., Zhou, Z., Qiao, Y.: Smallbignet: Integrating core and contextual views for video classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1092–1101 (2020)
32. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
33. Lin, X., Petroni, F., Bertasius, G., Rohrbach, M., Chang, S.F., Torresani, L.: Learning to recognize procedural activities with distant supervision. arXiv preprint arXiv:2201.10990 (2022)
34. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
36. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021)

37. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
38. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2630–2640 (2019)
39. Patrick, M., Campbell, D., Asano, Y.M., Metze, I.M.F., Feichtenhofer, C., Vedaldi, A., Henriques, J., et al.: Keeping your eye on the ball: Trajectory attention in video transformers. arXiv preprint arXiv:2106.05392 (2021)
40. Peng, Y., Zhao, Y., Zhang, J.: Two-stream collaborative learning with spatial-temporal attention for video classification. IEEE Transactions on Circuits and Systems for Video Technology **29**(3), 773–786 (2018)
41. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
42. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
43. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199 (2014)
44. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7464–7473 (2019)
45. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1207–1216 (2019)
46. Tang, Y., Lu, J., Zhou, J.: Comprehensive instructional video analysis: The coin dataset and performance evaluation. IEEE transactions on pattern analysis and machine intelligence **43**(9), 3138–3153 (2020)
47. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4631–4640 (2016)
48. Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5552–5561 (2019)
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
50. Vicol, P., Tapaswi, M., Castrejon, L., Fidler, S.: Moviegraphs: Towards understanding human-centric situations from videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8581–8590 (2018)
51. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
52. Wu, C.Y., Girshick, R., He, K., Feichtenhofer, C., Krahenbuhl, P.: A multigrid method for efficiently training video models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 153–162 (2020)

53. Wu, C.Y., Krahenbuhl, P.: Towards long-form video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1884–1894 (2021)
54. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A graph-based framework to bridge movies and synopses. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4592–4601 (2019)
55. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems **32** (2019)
56. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems **33**, 17283–17297 (2020)
57. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6720–6731 (2019)
58. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European conference on computer vision (ECCV). pp. 803–818 (2018)
59. Zhou, J., Lin, K.Y., Li, H., Zheng, W.S.: Graph-based high-order relation modeling for long-term action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8984–8993 (2021)
60. Zolfaghari, M., Singh, K., Brox, T.: Eco: Efficient convolutional network for online video understanding. In: Proceedings of the European conference on computer vision (ECCV). pp. 695–712 (2018)