Efficient Movie Scene Detection using State-Space Transformers

Md Mohaiminul Islam[1*]   Mahmudul Hasan[2]   Kishan Shamsundar Athrey[2]

Tony Braskich[2]   Gedas Bertasius[1]

[1]UNC Chapel Hill   [2]Comcast Labs

Abstract

The ability to distinguish between different movie scenes is critical for understanding the storyline of a movie. However, accurately detecting movie scenes is often challenging as it requires the ability to reason over very long movie segments. This contrasts with most existing video recognition models, which are typically designed for short-range video analysis. This work proposes a State-Space Transformer model that can efficiently capture dependencies in long movie videos for accurate movie scene detection. Our model, called TranS4mer, is built using a novel S4A building block, combining the strengths of structured state-space sequence (S4) and self-attention (A) layers. Given a sequence of frames divided into movie shots (uninterrupted periods where the camera position does not change), the S4A block first applies self-attention to capture short-range intra-shot dependencies. Afterward, the state-space operation in the S4A block aggregates long-range inter-shot cues. The final TranS4mer model, which can be trained end-to-end, is obtained by stacking the S4A blocks one after the other multiple times. Our proposed TranS4mer outperforms all prior methods in three movie scene detection datasets, including MovieNet, BBC, and OVSD, while being $2\times$ faster and requiring $3\times$ less GPU memory than standard Transformer models. We will release our code and models.

1. Introduction

Imagine watching *The Godfather* movie as illustrated in Figure 1. The first scene starts in the office room of mafia boss Don Vito Corleone, where he is discussing family business affairs with his eldest son Sonny Corleone and the family lawyer Tom Hagen. Then in the second scene, the family meets Virgil Sollozzo to hear about his business proposition. Afterward, the movie transitions into a third scene, where Don Vito chides his son Sonny for interfering. All of these movie scenes are inherently connected and create a complex storyline. Thus, the ability to recognize such

movie scenes is essential for understanding the plot of a movie. Moreover, identifying movie scenes enables broader applications, such as content-driven video search, preview generation, and minimally disruptive ad insertion.

Unlike standard video recognition tasks (e.g., action recognition), which require a short-range analysis of video clips lasting only 5–10 seconds, movie scene detection requires short- and long-range understanding, where videos might last several minutes. For instance, to successfully recognize the scenes in Figure 1, we need to identify the visual concepts in each frame, such as characters, objects, and backgrounds (short-range modeling), while also reasoning about complex temporal events and activities (long-range modeling). For example, if we only look at the boundary frames of the first and second scenes of Figure 1 and identify the local properties (e.g., characters, objects, and backgrounds), it is quite difficult to determine the transition between the two scenes. However, considering longer temporal windows with frames from both scenes, we can recognize global events and identify their boundaries.

We also note that models that only use short-range temporal context typically produce many false positives due to abrupt low-level changes in the scene dynamics (e.g., camera switching between characters). For example, if we look at the middle two frames of scene 2 in Figure 1, we might mistake them for a scene boundary since the frames contain two characters. However, looking at all the frames in scene 2 makes it clear that they belong to the same scene. Therefore, a successful movie scene detection model should identify the short-range visual elements and also reason about their long-range temporal dependencies in a movie segment.

However, most existing movie scene detection methods [11, 35] are built using convolutional neural networks (CNNs), which are inherently designed using local operators (i.e., convolution kernels) for short-range video analysis. Thus, such models often fail to capture the long-range dependencies crucial for accurate movie scene detection. Recently, transformers have been shown to effectively model short-range and long-range dependencies in various Natural Language Processing (NLP) tasks [1, 7, 14, 15, 32, 37, 47, 49]. Inspired by the success of transform-

---

*Research done while MI was an intern at Comcast Labs.

Figure 1. Three scenes from the movie *The Godfather*. In the first scene, The Godfather (Don Vito Corleone) discusses family business affairs with his eldest son Sony and family lawyer Tom Hagen. Then the movie transitions into the second scene, where the mafia family meets Virgil Sollozzo to hear his business proposition. Afterward, in the third scene, Don Vito chides his son Sonny for interfering in the middle of his talk. Generally, movies are composed of well-defined scenes, and detecting those scenes is essential to high-level movie understanding. Therefore, this work aims to detect high-level scenes from a long-range movie video.

ers in NLP, subsequent works have shown the remarkable success of using transformers in several short-range vision tasks [8, 16]. However, applying Transformers to long video sequences remains challenging due to the quadratic cost of self-attention. To overcome these issues, recent work proposed a structured state-space sequence (S4) operator [18, 19, 33] for efficient long-range sequence analysis.

Motivated by the developments in transformer and S4-based architectures, we present TranS4mer, an efficient end-to-end model for movie scene boundary detection, which combines the strengths of transformers for short-range modeling and S4 operators for long-range modeling. Specifically, we propose a novel state-space self-attention (S4A) block and build our TranS4mer model by stacking multiple S4A blocks on top of each other. Our model takes a sequence of frames divided into movie shots as input, where a shot is defined as a series of frames captured by the same camera over an uninterrupted period [46]. The S4A block first captures the short-range intra-shot dependencies by applying self-attention over the frames in each shot. Subsequently, it aggregates the inter-shot interactions over long movie segments by employing the state-space operator over all shots in a given movie video.

One key property of our model is that it independently applies self-attention to each shot, significantly reducing the cost of standard self-attention. For instance, if we have a video of 25 shots and each shot contains 3 frames, applying a standard vision transformer [16] with a patch size of 32×32 will result in 3,675 tokens and a single self-attention operation will require $\sim$13.5M pairwise comparisons. In contrast, applying self-attention within each shot parallelly reduces the number of comparisons $25\times$ times. Simultaneously, TranS4mer attains long-range modeling ability through the efficient state-space layers that do not require costly pairwise comparisons and operate over the entire input video. As a result, the proposed TranS4mer model can be efficiently applied to long movie videos.

We experiment with three movie scene detection datasets: MovieNet [25], BBC [3], and OVSD [40], and report that our method outperforms prior approaches by large margins ($+\mathbf{3.38}\%$ AP, $+\mathbf{4.66}\%$ and $+\mathbf{7.36}\%$ respectively). TranS4mer is also $\mathbf{2}\times$ faster and requires $\mathbf{3}\times$ less GPU memory than the pure Transformer-based models. Moreover, to evaluate the generalizability of our model,

we experiment with several long-range video understanding tasks, such as movie clip classification and procedural activity recognition. TranS4mer performs the best in 5 out of 7 movie clip classification tasks on LVU [48], the best in procedural activity classification on Breakfast [30], and the second-best in procedural activity classification on COIN [43] datasets.

2. Related works

Long Sequence Modeling. Many modern approaches to long sequence modeling started in the field of Natural Language Processing (NLP). Bahdanau *et al.* [1] introduced an RNN architecture with an attention mechanism for machine translation. Afterward, Vaswani *et al.* [47] proposed the transformer model for capturing long-range dependencies in text sequences. Following this work, various improved transformer architectures were proposed [7, 14, 15, 32, 37, 49]. However, despite these advances, applying transformers to very long sequences is still challenging because of the quadratic cost of the self-attention operation. Consequently, many recent methods have developed efficient variations of standard self-attention [12, 26, 29, 36, 50]. Furthermore, the recent work of Gu *et al.* [17–19] proposed a structured state-space sequence (S4) model for long sequence analysis. The S4 models have achieved impressive results in many NLP tasks [17–20, 33], while having only linear memory and computation costs (w.r.t the sequence length). Building on this work, Islam *et al.* [34] proposed a model that applies S4 on pre-extracted video features for long-range video classification. In contrast, we propose an end-to-end trainable movie scene detection model that combines S4 and self-attention into a novel S4A block, which we use to build our TranS4mer model.

Movie Scene Detection. The early approaches for movie scene detection consisted predominantly of unsupervised clustering-based methods [2, 4, 10, 39, 40, 42, 52] built using hand-crafted features. Recently, Huang *et al.* [25] introduced a large-scale dataset for movie scene detection called MovieNet, which became the most popular benchmark for this problem. Subsequently, Huang *et al.* [25] and Rao *et al.* [38] proposed an LSTM-based model for movie scene detection, which is applied on top of pre-extracted CNN features. Most recently, Chen *et al.* [11] and Mun *et al.* [35] introduced several end-to-end movie
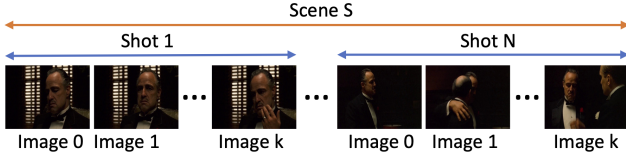
Figure 2. The hierarchical structure of long movie videos. First, multiple consecutive frames captured from the same camera position depict a shot. Next, multiple neighboring shots constitute a movie scene. Finally, a movie is composed of a collection of scenes. Here, we show one scene that contains N shots with K frames in each shot.

scene detection approaches built using 2D/3D CNN architectures. However, these CNN-based models are limited by the smaller receptive fields of the local convolution operations. In contrast, our TranS4mer model is constructed using global operators (state-space and self-attention), which are more effective for long-range modeling.

3. Background

The State-Space Model (SSM) is a fundamental model used in many scientific fields, such as control theory and computational neuroscience. We can define the SSM in continuous time using the following equations:

$$h'(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t) + Dx(t) \tag{1}$$

Here, $x(t)$ is the input signal, $h(t)$ is a latent state, $h'(t)$ is the first order derivative of $h(t)$ wrt t, i.e., $h'(t) = \frac{dh}{dt}$, $y(t)$ is the output signal, and A is the state matrix. Equation (1) can be discretized by sampling the underlying input signal $u(t)$ with a step size Δ.

$$h_k = \overline{A}h_{k-1} + \overline{B}x_k$$
$$y_k = \overline{C}h_k + \overline{D}x_k \tag{2}$$

Here, $\overline{A}$, $\overline{B}$, $\overline{C}$, and $\overline{D}$ are the discrete versions of A, B, C and D. Furthermore, the equation (2) can be expressed as a convolution operation ($*$):

$$y = \overline{K} * x \qquad \overline{K} = (\overline{CB}, \overline{CAB}, \dots, \overline{CA}^{L-1}\overline{B}), \tag{3}$$

where $\overline{K}$ is a kernel, and L is the sequence length.

S4 Model. The Structured State Space (S4) model is a particular class of SSM that requires the state matrix A to be *diagonal and low-rank*. Because of the specific structure of the state matrix A, we can derive a closed-form formula for the convolutional kernel $\overline{K}$ in equation (3) [18], which then facilitates an efficient computation of $\overline{K}$ without having to multiply the matrix A, $L-1$ times.

Gated S4 Model. Borrowing the idea from Gated Attention [24], Metha *at el.* [33] introduced a Gated S4 model, which first uses linear layers to compute two latent representations (u and v) from the input sequences x, and then

applies an S4 layer to obtain a contextualized representation $\hat{u}$. Finally, the tokens from $\hat{u}$ are used to gate v:

$$u = \phi(W_u x) \qquad v = \phi(W_v x)$$
$$h = S4(u) \qquad \hat{u} = W_h x \tag{4}$$
$$y = W_y(\hat{u} \odot v)$$

Here, W_u, W_v, W_h, W_y are the weight matrices of linear layers, ϕ denotes GELU nonlinearity [23], $\odot$ is elementwise multiplication, and S4 denotes an S4 layer described above.

4. Technical Approach

We aim to develop an efficient scene detection method for long movie videos. Movie videos have a specific hierarchical structure consisting of three semantic levels: (i) frames, (ii) shots, and (iii) scenes (See Figure 2). First, multiple consecutive frames constitute a shot, where the frames are captured from the same camera position over an uninterrupted period [46]. Second, multiple consecutive shots construct a scene, where a scene is a semantically cohesive unit of the storyline [27, 41]. Finally, a movie is composed of a collection of scenes. Shots can be easily recognized by detecting the changes in low-level visual cues [9, 13, 42]. However, identifying the scene boundary is more challenging as it requires high-level semantic understanding.

Problem Overview. First, we formally define the problem of movie scene detection. We are given a movie video V divided into T non-overlapping automatically-detected shots, $V = \{s_1, \dots, s_i, \dots, s_T\}$. We can then uniformly sample K frames from each shot. Therefore, a shot is represented as $s_i = \{f_1, \dots, f_j, \dots, f_K\}$, where $f_j \in \mathbb{R}^{3 \times H \times W}$ depicts an RGB frame, H is the height, and W is the width of each frame. Following these definitions, we can then formulate the movie scene detection problem as a prediction task, where we need to produce a binary output $y_i \in \{0, 1\}$ for each video shot s_i (i.e., whether a given shot is at a movie scene boundary or not).

Approach Overview. Our movie scene detection model, called the State-Space Transformer (TranS4mer), is built using our proposed S4A blocks that combine the strengths of structured state-space sequence (S4) and self-attention (A) layers. The S4A block utilizes the self-attention layer to encode short-range dependencies between frames in each shot and the S4 layer for capturing long-range dependencies between shots far away from each other. A detailed illustration of our model is presented in Figure 3. Below, we describe each component of our model in more detail.

4.1. The TranS4mer Model

Following previous methods [11, 35, 38], our model takes as input $N = 2m + 1$ temporally adjacent shots, $\mathcal{V}_i = \{s_{i-m}, \dots, s_i, \dots, s_{i+m}\}$, and outputs a binary prediction $y_i \in \{0, 1\}$ for the shot s_i. The neighboring shots provide
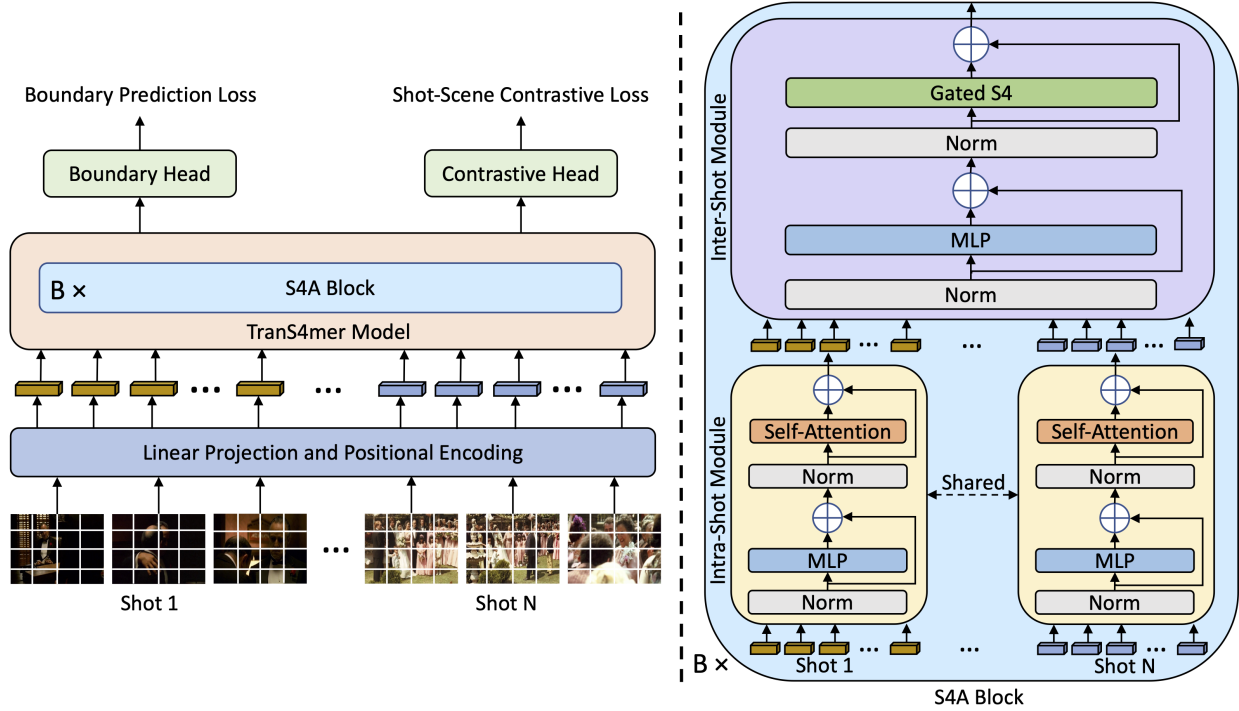
Figure 3. Overview of the TranS4mer model. **Left:** Our model takes a sequence of frames divided into automatically-detected shots. Following the ViT [16], the frames are decomposed into non-overlapping patches and passed through a linear projection layer. Afterward, we feed these patch-level embeddings through our TranS4mer model, which consists of B state-space self-attention (S4A) blocks stacked on top of each other. Finally, the CLS tokens of all shots are passed to two linear layers, i.e., the boundary prediction head (which outputs a binary label indicating the boundary shot) and the contrastive head (which maximizes intra-shot similarity). **Right:** Each of our proposed S4A blocks contains a self-attention-based intra-shot module that models the interaction among frames in the same shot and a Gated S4-based inter-shot module that aggregates long-range temporal cues from shots far away from each other.

long-range contextual cues for predicting the boundary at shot s_i. Here, $\mathcal{V}_i \in \mathbb{R}^{N \times K \times 3 \times H \times W}$ is composed of K frames from each of N consecutive shots, where H is the height, and W is the width of each frame.

Following ViT [16], we divide each frame into P non-overlapping patches of size $p \times p$, where $P = HW/p^2$. Then we use a linear layer to project each patch into a hidden dimension D, and a positional embedding is added to each patch token. As a result, we obtain a tensor of shape $\mathbf{V}_i \in \mathbb{R}^{N \times K \times P \times D}$, which is fed to our TranS4mer model.

The S4A Block. Our proposed S4A block enables us to compute a strong short-range feature representation and model long-range temporal dependencies among shots. It consists of two modules: a short-range intra-shot module and a long-range inter-shot module.

Intra-Shot Module. The intra-shot module takes the input tensor $\mathbf{V}_i \in \mathbb{R}^{N \times K \times P \times D}$, which we can think of as a collection of N shots, where a sequence of tokens represents each shot $S_i = x_1, \ldots, x_L$, where $x_i \in \mathbb{R}^D$, $L = K \times P$. The intra-shot module models the sequence of tokens from each shot independently. To this end, it first applies a multi-layer perceptron (MLP) layer and a residual connection. Afterward, it applies a Multiheaded Self Attention (MSA) layer and another residual connection. Layer Normalization (LN) is applied before the MLP

and the MSA layers. The module can be defined as follows:

$$\mathbf{x}' = \text{MHA}(\text{LN}(\mathbf{x}_{in})) + \mathbf{x}_{in}$$
$$\mathbf{x}_{out} = \text{MLP}(\text{LN}(\mathbf{x}')) + \mathbf{x}' \tag{5}$$

The output of the intra-shot module $\mathbf{V}'_i \in \mathbb{R}^{N \times K \times P \times D}$, is then passed to the inter-shot module.

Note that applying self-attention to tokens from each shot separately reduces the computation cost to $O(NL^2)$. In contrast, applying self-attention to the entire sequence of tokens in $\mathbf{V_i}$ would cost $O(N^2L^2)$.

Inter-Shot Module. Our previously defined intra-shot module only models short-range dependencies and, thus, lacks the ability for long-range temporal understanding, which is crucial for the movie scene detection task. To address this limitation, we propose an efficient inter-shot module based on the Gated S4 operator that aggregates long-range inter-shot cues.

First, we unroll the input tensor $\mathbf{V}'_i$ into a sequence $z_1, \ldots, z_{L'}$, where $z_i \in \mathbb{R}^D$, $L' = N \times K \times P$. Then, we pass the sequence to a multi-layer perceptron (MLP) layer and a residual connection. Afterward, we apply a Gated S4 (GS4) layer and another residual connection. Layer Normalization (LN) is applied before MLP and GS4 layers:

$$\mathbf{z}' = \text{GS4}(\text{LN}(\mathbf{z}_{in})) + \mathbf{z}_{in}$$
$$\mathbf{z}_{out} = \text{MLP}(\text{LN}(\mathbf{z}')) + \mathbf{z}' \qquad (6)$$

The output of this operation is then passed into the next S4A block. Unlike the intra-shot module, which operates on a short sequence of length $L = K \times P$, the inter-shot module operates on much longer sequences of $L' = N \times K \times P$. Our model can efficiently process such long sequences due to the Gated S4 layers in the inter-shot module.

4.2. Training and Loss Functions

Most prior works in move scene detection use self-supervised pretraining [11, 35]. Thus, we also adopt the self-supervised pretraining strategy of Mun *at el.* [35], which uses pseudo boundaries of move scenes from the Dynamic Time Warping algorithm as a supervisory signal. Afterward, we finetune our pretrained model using the labeled subset of the data. Note that we do not claim technical novelty over the pretraining strategy. Instead, our main goal is to develop an efficient state-space transformer backbone for movie scene detection. Below, we provide more details related to the pretraining and fine-tuning protocols.

Pretraining. Leveraging the pseudo movie scene boundaries, we optimize our model using the following two losses: (i) a shot-scene contrastive loss and (ii) a pseudo-boundary prediction loss. The contrastive loss maximizes the similarity between the representations of a shot and its corresponding pseudo ground truth scene while also minimizing the similarity of a shot with all the other scenes. We represent each shot with its respective CLS token, whereas each scene is represented by mean-pooling the CLS tokens of all shots in that scene. The DTW algorithm divides the input video $\mathcal{V}_i = \{s_{i-m}, \ldots, s_i, \ldots, s_{i+m}\}$ into two pseudo-scenes $\mathcal{V}_L = \{s_{i-m}, \ldots, s_{i*}\}$ and $\mathcal{V}_R = s_{i*+1}, \ldots, s_{i+m}\}$, here s_{i*} is the pseudo-boundary shot. We can then define the shot-scene contrastive loss as:

$$l_C(\mathbf{r}, \bar{\mathbf{r}}) = -\log \frac{\mathcal{S}(\mathbf{r}, \bar{\mathbf{r}})}{\mathcal{S}(\mathbf{r}, \bar{\mathbf{r}}) + \sum_{\mathbf{r_n}} \mathcal{S}(\mathbf{r}_n, \bar{\mathbf{r}}) + \sum_{\bar{\mathbf{r}}_n} \mathcal{S}(\mathbf{r}, \bar{\mathbf{r}}_n)} \quad (7)$$

Here, $\mathbf{r}$ is a shot representation obtained by applying a linear layer on the CLS token of that shot, $\bar{\mathbf{r}}_L$ and $\bar{\mathbf{r}}_R$ are the scene-level representations of pseudo-scenes $\mathcal{V}_L$ and $\mathcal{V}_R$, $\mathbf{r}_n$ and $\bar{\mathbf{r}}_n$ are the negative shots and scenes from the mini-batch. Similarity between two representations is calculated as: $\mathcal{S}(\mathbf{x}, \mathbf{y}) = \exp((\mathbf{x}^\top \mathbf{y})/(\|\mathbf{x}\|\|\mathbf{y}\|))$. The first loss term can then be written as: $\mathcal{L}_C = l_C(\mathbf{r}_{i-m}, \bar{\mathbf{r}}_L) + l_C(\mathbf{r}_{i+m}, \bar{\mathbf{r}}_R)$.

The second loss function of pseudo-boundary prediction is a simple objective that guides the model to predict a pseudo-boundary from a given shot representation. We can define this loss as a binary cross-entropy loss:

$$\mathcal{L}_B = -\log(\rho_b(\mathbf{r}_{i*})) - \log(1 - \rho_b(\mathbf{r}_{\bar{b}})) \qquad (8)$$

Here, $\mathbf{r}_{i*}$ is the CLS representation of the pseudo-boundary shot s_{i*}, $\mathbf{r}_{\bar{b}}$ is the representation of a randomly selected non-boundary shot $s_{\bar{b}}$, ρ_b is a projection layer that outputs the probability of a shot being the pseudo-boundary.

The overall pretraining loss is the sum of the shot-scene contrastive loss and the pseudo-boundary loss.

Finetuning. After the self-supervised pretraining, we finetune our model using the subset of the dataset with ground truth annotations. We use a binary cross-entropy loss as in [35] to finetune our model.

4.3. Implementation Details

Following the ViT [16], we resize video frames to a spatial dimension of 224×224 and divide using a patch size of 32×32. Our model contains 12 S4A blocks and uses a hidden dimension of 384. It takes 25 neighboring shots as input, where each shot contains 3 frames. We use four data augmentation techniques: random crop, flip, color jitter, and Gaussian blur. We use a batch size of 256 and pretrain our model for 10 epochs before finetuning for 20 epochs. During pretraining, we use a base learning rate of 0.3, which we reduce using a cosine schedule after a linear warmup strategy of 1 epoch. During finetuning, we use a learning rate of 10^{-6}. We use the Adam [28] optimizer with a momentum of 0.9 and weight decay of 10^{-6}. We train our models using 8 NVIDIA RTX A6000 GPUs.

5. Experimental Setup

5.1. Datasets

MovieNet [25] is a large-scale dataset containing 1100 movies with 1.6 million shots, and 318 of these movies have ground-truth scene boundary annotation. The annotated subset of the dataset is divided into 190, 64, and 64 movies for training, validation, and testing. Following the previous approaches, we pretrain our model using all the movies, finetune using the standard training split of the annotated subset, and report the result on the test split.

BBC [3] contains 11 episodes from the BBC TV series *Planet Earth* [5]. The average duration of the videos is 50 minutes. The dataset contains 670 scenes and 4.9K shots.

OVSD [40] contains 21 short films with an average duration of 30 minutes. It has a total of 10K shots and 300 scenes extracted from movie scripts.

5.2. Evaluation Metrics

AP measures the weighted average of precisions at different thresholds.

mIoU is the average intersection over union of predicted scene boundaries with the closest ground truth boundaries.

AUC-ROC is calculated as the True Positive Rate with respect to the False Positive Rate at different thresholds.

F1-Score is the harmonic mean of precision and recall.

	Method	AP (↑)	mIoU (↑)	AUC-ROC (↑)	F1 (↑)	Memory (GB) (↓)	Samples/s (↑)
Unsupervised	GraphCut [39]	14.10	29.70	-	-	-	-
	SCSA [10]	14.70	30.50	-	-	-	-
	DP [21]	15.50	32.00	-	-	-	-
	Story Graph [45]	25.10	35.70	-	-	-	-
	Grouping [40]	33.60	37.20	-	-	-	-
	BaSSL [35]	31.55	39.36	71.67	32.55	34.28	0.96
	Transformer [16]	32.16	37.24	70.23	31.24	30.28	1.27
	TimeSformer [6]	32.47	37.76	70.95	31.45	28.12	1.47
	Vanilla S4 [18]	33.34	38.12	71.86	32.21	15.62	1.83
	TranS4mer	**34.45**	**39.60**	**73.25**	**33.41**	**10.13**	**2.57**
Supervised	Siamese [3]	35.80	39.60	-	-	-	-
	MS-LSTM [25]	46.50	46.20	-	-	-	-
	LGSS [38]	47.10	48.80	-	-	-	-
	ViS4mer [34]	55.13	48.27	88.74	46.15		
	ShotCoL [11]	53.40	-	-	-	34.28	0.96
	BaSSL [35]	57.40	50.69	90.54	47.02	34.28	0.96
	Transformer [16]	58.81	51.21	90.84	47.88	30.28	1.27
	TimeSformer [6]	59.62	50.75	90.66	48.02	28.12	1.47
	Vanilla S4 [18]	59.71	51.32	90.96	47.85	15.62	1.83
	TranS4mer	**60.78**	**51.91**	**91.89**	**48.36**	**10.13**	**2.57**

Table 1. **Results on MovieNet [25].** Horizontal lines in both Unsupervised and Supervised rows separate prior works from our implemented baselines. First, our method achieves state-of-the-art performance, outperforming the previous best method (BaSSL) by a significant margin of **3.38**% AP. Second, compared to prior unsupervised approaches, the unsupervised TranS4mer model achieves better performance in all metrics. Finally, TranS4mer is **2.68**× faster and requires **3.38**× less GPU memory than the previous SOTA methods (ShotCol and BaSSL), while also being faster and more memory efficient than the transformer and state-space baselines.

5.3. Baselines

Prior Methods. In our comparisons, we include several unsupervised approaches such as GraphCut [39], SCSA [10], DP [21], Story Graph [45], and Grouping [40]. These methods use various clustering techniques using pre-extracted features. Moreover, we compare our approach with supervised approaches, such as Siamese [3], MS-LSTM [25], and LGSS [38], which also use pre-extracted CNN features. We also compare against the long-range video model ViS4mer [34], which uses S4 layers as a late-fusion strategy. Finally, we include two recent end-to-end trainable models that use self-supervised pretraining, namely, ShotCoL [11] and BaSSL [35]. Both models use CNN models based on ResNet-50 [22].

Our Implemented Baselines. To make our comparisons more complete, we implement three additional baselines: *1) Transformer:* We construct this baseline by following the standard Vision Transformer [16]. In particular, we replace the Gated S4 layers of the TranS4mer model with self-attention layers and keep everything else unchanged. *2) TimeSformer:* We construct this baseline by replacing the Gated S4 layers of our model with the divided space-time attention proposed by Bertasius *at el.* [6]. *3) Vanilla S4:* We develop this baseline by replacing the Gated S4 layers of the TranS4mer model with the S4 layers [18].

6. Results and Analysis

6.1. Main Results on the MovieNet Dataset

Comparison to the State-of-the-Art. We present the results on MovieNet [25] in Table 1. We first compare our method with the unsupervised approaches, including GruphCut [39], SCSA [10], DP [21], Story Graph [45], Grouping [40], and BaSSL (unsupervised) [35]. For a fair comparison, we train an unsupervised variant of our TranS4mer model using pseudo-boundary loss without ground-truth annotations and report the result without supervised finetuning. Our results suggest that our unsupervised TranS4mer model achieves the best performance according to all metrics and outperforms the prior best method Grouping by **+0.85**% AP. Next, we compare the supervised variant of our model (i.e., finetuned with ground truth annotations) with the state-of-the-art supervised methods (ShotCol [11] and BaSSL [35])). We also compare with the state-space-based long-range video classification model ViS4mer [34]. We observe that TranS4mer achieves the best performance in all metrics and outperforms the previous best method (BaSSL) by a large margin (**+3.38**% AP).

Comparison with Transformer and State-space Baselines. In Table 1, we also compare our TranS4mer model with the Transformer and State-space baselines: Trans-

former, TimeSformer, and Vanilla S4 (See subection 5.3). Note that all three baselines use intra-shot and inter-shot modules; however, instead of a Gated S4 layer, Transformer uses standard self-attention [16], TimeSformer uses divided space-time attention [18], and Vanilla S4 uses standard S4 [18] at the inter-shot module. Moreover, similar to the TranS4mer-Unsupervised method, we train unsupervised variants of these baselines without any ground-truth annotation. First, we observe that all three baselines outperform the previous state-of-the-art method (BaSSL [35]) in both supervised and unsupervised settings, indicating the effectiveness of our proposed intra-shot and inter-shot block designs. Second, TranS4mer performs better than all three of these baselines, outperforming Transformer by **+1.97**% and **+2.29**%, TimeSformer by **+1.16**% and **+1.98**%, and Vanilla S4 by **+1.07**% and **+1.11**% in supervised and unsupervised approaches, respectively. These experiments indicate that TranS4mer is better suited for movie scene detection than the standard self-attention and state-space models.

Computational Cost Analysis. We compare the GPU memory consumption (in gigabytes) and the training speed (samples/s) of our model with the previous best methods as well as our implemented baselines in Table 1. First, we observe that, in addition to achieving the best performance, our method also requires **3.38**× less memory and is **2.68**× faster than the previous state-of-the-art method BaSSL [35]. Second, compared to the standard Transformer baseline, TranS4mer is **2**× faster and requires **3**× less GPU memory. Finally, our model is significantly cheaper than efficient self-attention and state-space models such as TimeSformer and Vanilla S4. Whereas TimeSformer and Vanilla S4 require **28.12** GB and **15.62** GB of memory, our TranS4mer requires only **10.13** GB of memory. Moreover, the training speed of TranS4mer is **2.57** samples/second compared to the **1.47** and **1.82** samples/second speed of TimeSformer and Vanilla S4, respectively. These observations suggest that TranS4mer is much more efficient than prior methods and standard self-attention and state-space-based models.

Comparison Using the Same Backbone Model. For a fair comparison, we also experiment with a previous-best BaSSL [35] method that uses the same backbone as our model (i.e., ViT-S/32). We report that our TranS4mer outperforms this BaSSL variant by **+2.77**% AP.

6.2. Scene Detection on Other Datasets

We experiment with two additional movie scene detection datasets, BBC [3] and OVSD [40]. We present these results in Table 2, which shows that TranS4mer achieves the best result among all methods, outperforming the previous best BaSSL [35] method by 4.66% AP on the BBC and 7.36% AP on the OVSD datasets. These results demonstrate our method's ability to generalize to other datasets.

Method	AP (↑)		Method	AP (↑)
BaSSL [35]	39.98		BaSSL [35]	28.68
Transformer [16]	41.86		Transformer [16]	33.12
TimeSformer [6]	42.23		TimeSformer [6]	33.87
Vanilla S4 [18]	42.56		Vanilla S4 [18]	34.21
TranS4mer	**43.64**		TranS4mer	**36.04**
(a) Performance on BBC [3].			(b) Performance on OVSD [40].	

Table 2. **Scene Detection on additional datasets.** TranS4mer outperforms the prior method (BaSSL) by a large margin of 4.66% AP on BBC [3] and 7.36% AP on OVSD [40] datasets.

Method	Relation	Speak	Scene	Director	Genre	Writer	Year
ObjTrans. [48]	53.10	39.40	56.90	51.20	54.60	34.50	39.10
ViS4mer [34]	57.14	**40.79**	67.44	62.61	54.71	**48.80**	44.75
TranS4mer	**59.52**	39.21	**70.93**	**63.86**	**55.85**	46.93	**45.45**
(a) Performance on LVU [48]							

Model	#Data(↓)	Acc.(↑)	Model	#Data(↓)	Acc.(↑)
GHRM [51]	306K	75.50	TSN [44]	306K	73.40
Dist.Sup. [31]	**136M**	89.90	Dist.Sup. [31]	**136M**	**90.00**
ViS4mer [34]	495K	88.17	ViS4mer [34]	495K	88.41
TranS4mer	495K	**90.27**	**TranS4mer**	495K	89.23
(b) Performance on Breakfast [30]			(c) Performance on COIN [43]		

Table 3. **Long-range video classification.** (a) TranS4mer achieves best performance in 5 out of 7 classification tasks in LVU benchmark [48]. Moreover, (b) it performs best on the Breakfast [30] and (c) second-best on the COIN [43] datasets, while using significantly less pretraining data (middle column).

6.3. Long-range Video Classification

In addition to move scene detection, we also experiment with other tasks including long-form movie clip classification in the LVU [48] dataset, and long-range procedural activity classification on Breakfast [30] and COIN [43] datasets. LVU contains $30K$ videos from $3K$ movies, Breakfast has 1,712 videos of 10 cooking activities, and COIN contains 11,827 videos of 180 different procedural activities. Videos of each dataset are $2-3$ minutes long. We present the results in Table 3. On LVU, TranS4mer achieves the best performance in 5 out of 7 tasks (Table 3(a)). Moreover, TranS4mer also achieves the best performance on Breakfast (Table 3(b)) and the second-best performance on COIN (Table 3(c)) while using significantly fewer pretraining samples than the state-of-the-art Distant Supervision [31] approach. These results indicate that TranS4mer can be effectively applied to other domains and tasks.

6.4. Ablation Studies

Temporal Extent Ablation. To verify our hypothesis that long-range temporal context is critical for accurate movie scene detection, we experiment with video inputs of different lengths (i.e., video inputs containing 9, 17, 25, and 33 contextual shots) while predicting the boundary for the middle shot. Note that videos with more input shots denote longer input videos. We present these results in Figure 4a. First, we notice that a short temporal window of 9 contex-

(a) **Performance** ↑ TranS4mer achieves much better performance for longer videos.

(b) **Training Speed** ↑ TranS4mer is significantly faster than the Transformer baseline.

(c) **Memory Requirements** ↓ TranS4mer requires significantly less GPU memory.
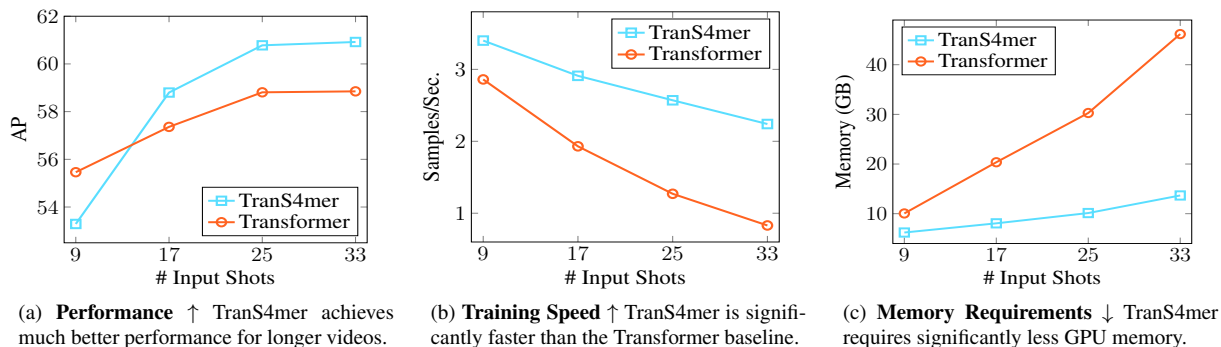
Figure 4. (a) Performance (AP), (b) Training Speed (Samples/Second), and (c) Memory Requirements (GigaBytes), and as a function of the number of input shots. Note that videos with more input shots denote longer input videos.

Intra-Shot	Inter-Shot	Ap (↑)	S4 layers	AP (↑)	S4 layers	AP (↑)	S4 layers	AP (↑)
✓	✗	57.80	1-6	58.31	every 2nd	59.82	S4	59.71
✗	✓	55.59	7-12	58.82	every 4th	58.01	DS4	60.13
✓	✓	**60.78**	1-12	**60.78**	all	**60.78**	GS4	**60.78**
(a) TranS4mer modules.			(b) S4 in different layers.		(c) S4 in every k^{th} layer.		(d) Different S4 variants.	

Table 4. Ablation studies with (a) TranS4mer modules, (b) (c) positioning, and (d) variants of S4 layers. Intra-shot and inter-shot modules are complementary, and TranS4mer with Gated S4 layers at every S4A block yields the best performance.

tual shots produces a suboptimal performance of **53.29%** AP. Increasing the temporal extent to 25 contextual shots increases the performance to **60.78%** AP (**+7.49%** boost). Adding more contextual shots does not yield any significant improvements. These results verify the importance of long-range temporal modeling for movie scene detection.

Detailed Comparison with Transformer. We next present a detailed comparison of our model with the Transformer baseline [16]. In particular, we experiment with videos that contain a different number of input shots and plot the accuracy in AP as a function of the number of input shots (see Figure 4). Although the Transformer baseline performs better for shorter video inputs, our TranS4mer model achieves much better performance using longer videos. Specifically, TranS4mer yields **+2.07%** larger AP than the transformer model when using the longest video inputs. This suggests that TranS4mer is more effective at capturing long-range dependencies for the movie scene detection task. Moreover, we also illustrate the training speed (Samples/Second) and the memory requirement (GigaBytes) as a function of the number of input shots (see Figures 4b and 4c). We observe that as we increase the video length, the training speed of the Transformer baseline drops drastically, and the memory requirement grows rapidly. Overall, our proposed TranS4mer model is **2.7×** faster and requires **3.38×** less GPU memory than the Transformer baseline while using the longest video inputs.

Ablation with Intra-Shot and Inter-Shot Modules. In Table 4a, we analyze the significance of our (i) self-attention-based intra-shot module and (ii) state-space-based inter-shot module. We observe that removing the inter-shot

module drops the performance by **2.98%** AP, whereas discarding the intra-shot module decreases the performance by **5.19%** AP. Therefore, based on these results, we can conclude that both the intra-shot and inter-shot modules are essential for good movie scene detection performance. We also note that they are complementary to each other, which aligns with our intuition of the intra-shot module being responsible for short-range representations while the inter-shot module is dedicated to long-range temporal reasoning.

Ablation with S4 Layers. We experiment with injecting our proposed S4A block into different encoder layers (Table 4b). We also evaluate a sparser TranS4mer variant with our S4A block injected into every k^{th} layer of the network (Table 4c). Lastly, we experiment with different S4 variants (i.e., Diagonal S4 (DS4) [20], Gated S4 (GS4) [33]) (Table 4d). Our results suggest that using S4A with Gated S4 in every layer of the network leads to the best accuracy.

7. Conclusion

In this work, we propose an efficient movie scene segmentation method which combines the strengths of self-attention for short-range modeling and state-space models for long-range modeling. Our method, named TranS4mer, achieves state-of-the-art results in three different movie scene segmentation datasets (MovieNet, BBC, and OVSD). Moreover, our model is significantly faster and requires less GPU memory than previous approaches and the standard vision transformer models. In future, we will extend our model to other long-range video understanding and multi-modal tasks such as movie summarization, movie trailer generation, and natural language grounding in long videos.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, et al. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv: 1409.0473*, 2014. 1, 2

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Analysis and re-use of videos in educational digital libraries with automatic scene detection. In *Italian Research Conference on Digital Libraries*, pages 155–164. Springer, 2015. 2

[3] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1199–1202, 2015. 2, 5, 6, 7

[4] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Shot and scene detection via hierarchical clustering for re-using broadcast video. In *International conference on computer analysis of images and patterns*, pages 801–811. Springer, 2015. 2

[5] BBC. Planet earth. https://www.bbc.co.uk/programmes/b006mywy. 5

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 6, 7

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[8] 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*, pages 213–229. Springer, 2020. 2

[9] Brandon Castellano. Pyscenedetect: Intelligent scene cut detection and video splitting tool, 2018. 3

[10] Vasileios T Chasanis, Aristidis C Likas, and Nikolaos P Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *IEEE transactions on multimedia*, 11(1):89–100, 2008. 2, 6

[11] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9796–9805, 2021. 1, 2, 3, 5, 6

[12] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 2

[13] Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. Video shot detection and condensed representation. a review. *IEEE signal processing magazine*, 23(2):28–37, 2006. 3

[14] 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. 1, 2

[15] 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. 1, 2

[16] 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. 2, 4, 5, 6, 7, 8

[17] Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. *International Conference on Machine Learning (ICML)*, 2022. 2

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2, 3, 6, 7

[19] Albert Gu, Isys Johnson, Karan Goel, Khaled Kamal Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2

[20] Ankit Gupta. Diagonal state spaces are as effective as structured state spaces. *arXiv preprint arXiv:2203.14343*, 2022. 2, 8

[21] Bo Han and Weiguo Wu. Video scene segmentation using a novel boundary evaluation criterion and dynamic programming. In *2011 IEEE International conference on multimedia and expo*, pages 1–6. IEEE, 2011. 6

[22] 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*, pages 770–778, 2016. 6

[23] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR, abs/1606.08415*, 3, 2016. 3

[24] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *International Conference on Machine Learning*, pages 9099–9117. PMLR, 2022. 3

[25] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*, pages 709–727. Springer, 2020. 2, 5, 6

[26] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 2

[27] Ephraim Katz. *The film encyclopedia*. Thomas Y. Crowell, 1979. 3

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[29] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020. 2

[30] Hilde Kuehne, Ali Arslan, and Thomas Serre. 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*, pages 780–787, 2014. 2, 7

[31] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. *arXiv preprint arXiv:2201.10990*, 2022. 7

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 2

[33] Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022. 2, 3, 8

[34] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. *European Conference on Computer Vision (ECCV)*, 2022. 2, 6, 7

[35] Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Boundary-aware self-supervised learning for video scene segmentation. *arXiv preprint arXiv:2201.05277*, 2022. 1, 2, 3, 5, 6, 7

[36] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 2

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1, 2

[38] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10146–10155, 2020. 2, 3, 6

[39] Zeeshan Rasheed and Mubarak Shah. Detection and representation of scenes in videos. *IEEE transactions on Multimedia*, 7(6):1097–1105, 2005. 2, 6

[40] Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. In *2016 IEEE international symposium on multimedia (ISM)*, pages 275–280. IEEE, 2016. 2, 5, 6, 7

[41] Yong Rui, Thomas S Huang, and Sharad Mehrotra. Constructing table-of-content for videos. *Multimedia systems*, 7(5):359–368, 1999. 3

[42] Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, 2011. 2, 3

[43] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2, 7

[44] Yansong Tang, Jiwen Lu, and Jie Zhou. Comprehensive instructional video analysis: The coin dataset and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3138–3153, 2020. 7

[45] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Storygraphs: visualizing character interactions as a timeline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 827–834, 2014. 6

[46] Paul-Louis Thirard and Lorenzo Codelli. Robert sklar. film, an international history of the medium, 1993;; kristin thompson, david bordwell. film history, an introduction, 1994. *1895, revue d'histoire du cinéma*, 17(1):170–170, 1994. 2, 3

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[48] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021. 2, 7

[49] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019. 1, 2

[50] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020. 2

[51] Jiaming Zhou, Kun-Yu Lin, Haoxin Li, and Wei-Shi Zheng. Graph-based high-order relation modeling for long-term action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8984–8993, 2021. 7

[52] Xiang Sean Zhou, Yong Rui, and Thomas S Huang. Constructing table-of-content for videos. In *Exploration of Visual Data*, pages 53–73. Springer, 2003. 2