Expert Systems With Applications 323 (2026) 132422

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Video plot segmentation

Xichen Tan [a,1], Yuanjing Luo [b,1], Yunfan Ye [c], Guang Yu [d], Chengyu Wang [a], Fang Liu [c,*], Zhiping Cai [a]

[a] College of Computer Science and Technology, National University of Defense Technology, Changsha, Hunan, 410073, China
[b] College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha, Hunan, 410004, China
[c] School of Design, Hunan University, Changsha, Hunan, 410082, China
[d] College of Meteorology and Oceanography, National University of Defense Technology, Changsha, Hunan, 410073, China

ARTICLE INFO

Keywords:
Video plot segmentation
Video understanding
Spatio-temporal attention
Vision transformer

ABSTRACT

We present Video Plot Segmentation, a novel and fundamental task in video understanding that aims to automatically extract semantically coherent segments from videos, each corresponding to distinct narrative or thematic units. This task requires in-depth analysis and comprehension of video content and is motivated by the need for higher-level structural representations in video understanding. To tackle this challenging task, we introduce MangNet, the first benchmark for Video Plot Segmentation annotated by professional video editors. MangNet is designed to assess models' ability to capture narrative coherence and to objectively evaluate plot segmentation performance. Furthermore, we propose the Cluster Vision Transformer (CiT), a model tailored for plot segmentation by modeling the consistency of narrative elements (i.e., characters, objects, and backgrounds) across different scenes while integrating spatio-temporal information. CiT operates primarily through Cluster-Attention blocks consisting of two stages: a *Cluster Stage*, which captures correlations among characters, objects, and backgrounds by clustering spatial video tokens with dynamically updated cluster centers during training, and a *Separate Attention Stage*, which emphasizes plot discontinuities by applying self-attention over cluster tokens in spatio-temporal dimensions. Experimental results demonstrate that CiT achieves consistent improvements on the Video Plot Segmentation task and effectively delineates plot boundaries.

1. Introduction

The exponential growth of video content across domains–from streaming platforms and social media to surveillance and educational archives–has intensified the demand for efficient video understanding systems. Video Plot Segmentation (VPS) addresses a critical challenge in video understanding: extracting meaningful semantic units from continuous video contents. Unlike traditional video segmentation tasks that focus on low-level visual features or temporal discontinuities, VPS aims to decompose videos into coherent narrative segments that capture the intrinsic semantic structure of the content.

The necessity of VPS arises from the limitations of treating long-form videos as undifferentiated continuous data. In scenarios such as retrieval, summarization, and content analysis, users and systems often need semantically meaningful access at an intermediate granularity. For feature-length documentaries or multi-episode series, scanning the full video or using coarse divisions can miss the narrative structure. VPS segments videos into coherent, narrative-driven units that can serve as an intermediate representation for organizing and navigating long-form content.

VPS provides several key advantages:

(1) Semantic Granularity: By identifying distinct plot segments, VPS transforms videos from linear, time-based streams into structured, semantically interpretable units. (2) Enhanced Retrieval Precision: In video search scenarios, users often seek specific narrative moments rather than entire videos. VPS enables precise, semantics-driven video querying that goes beyond keyword or metadata-based search. (3) Intelligent Content Understanding: Segmenting videos into meaningful plots can facilitate higher-level understanding by providing semantically coherent units for downstream processing (e.g., browsing, indexing, and summarization).

By segmenting videos into meaningful plots, VPS bridges the gap between original image feature processing and semantic understanding, offering a more intelligent, context-aware approach to video analysis.

* Corresponding author.
E-mail addresses: tanxc23@nudt.edu.cn (X. Tan), luoyuanjing@csuft.edu.cn (Y. Luo), yeyunfan@hnu.edu.cn (Y. Ye), guangyu@nudt.edu.cn (G. Yu), chengyu@nudt.edu.cn (C. Wang), fangl@hnu.edu.cn (F. Liu), zpcai@nudt.edu.cn (Z. Cai).
[1] Both authors contribute equally to this work.

https://doi.org/10.1016/j.eswa.2026.132422
Received 4 June 2025; Received in revised form 21 January 2026; Accepted 8 April 2026
Available online 9 April 2026
0957-4174/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

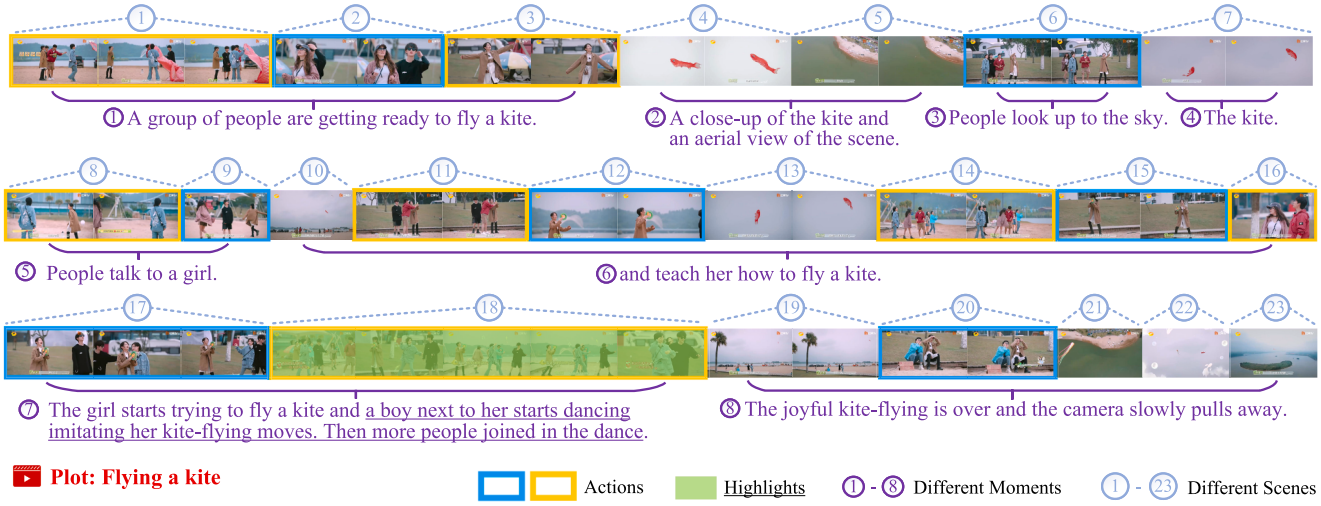


Fig. 1. Illustration of different temporal video analysis tasks within a plot: localized units (e.g., shots/actions/moments/scenes/highlights) versus narrative-coherent plot segments that span multiple such units. Each sampled video frame can be viewed as a representative shot.

Many video analysis tasks are closely related to VPS, including Temporal Action Localization (Cao et al., 2023; Hu et al., 2023; Lee et al., 2023; Liu et al., 2025, 2024; Moniruzzaman & Yin, 2023; Shou et al., 2016; Sun et al., 2022; Zhang et al., 2022), Video Highlight Detection (Chen et al., 2021b; Lei et al., 2021; Liu et al., 2022; Sul et al., 2023), Moment Retrieval (Anne Hendricks et al., 2017; Fang et al., 2025; Gao & Xu, 2021; Han et al., 2023; Huo et al., 2025; Sun et al., 2023; Zhang et al., 2019, 2020, 2023), as well as Scene Segmentation (Baraldi et al., 2015; Islam et al., 2023; Rao et al., 2020; Sidiropoulos et al., 2011; Xu et al., 2024; Yang et al., 2023; Zabih et al., 1995) and Shot Segmentation (Hassanien et al., 2017; Kar et al., 2024; Tang et al., 2018; Xu et al., 2016; Zhu et al., 2023). However, each of these tasks serves a distinct purpose and operates at a different semantic level. Temporal Action Localization targets atomic action instances, Video Highlight Detection selects salient short clips, and Moment Retrieval retrieves query-dependent temporal spans. Shot/scene segmentation instead follows structural cues (e.g., camera cuts) or setting consistency to produce low-level units. In contrast, VPS partitions an entire video into narrative-complete plot segments by modeling long-range storyline coherence that often spans multiple shots and scenes (see Fig. 2 and Table 1).

Further elaboration on these differences is provided in Figs. 1, 2, and Table 1. Fig. 1 shows a concrete kite-flying example, where one plot segment covers multiple actions/scenes/shots (e.g., preparation, interaction, flying, and ending). It also includes query-related moments and short highlights, illustrating that such localized events do not necessarily define plot boundaries. Fig. 2 provides an abstract, hierarchical comparison between VPS and related tasks (moment retrieval, temporal action localization, video highlight detection, and shot/scene segmentation), highlighting their differences in objective, temporal granularity, and boundary cues. In particular, VPS targets narrative-complete segments that often span multiple lower-level units, whereas other tasks focus on localized, structural, or query-dependent spans. Table 1 summarizes these distinctions across representative benchmarks, average ground-truth duration (Avg. GT Dur.), boundary cues, narrative-continuity requirements, and semantic impact levels. It shows that MangNet has the longest Avg. GT Dur., consistent with the denser semantics of plot segments, and that VPS places higher importance on character/object/background information, reflecting the need to jointly model these elements and their interactions for accurate plot segmentation.

Although previous tasks have been well-researched with mature solutions, the new task of VPS is more complex and requires specially

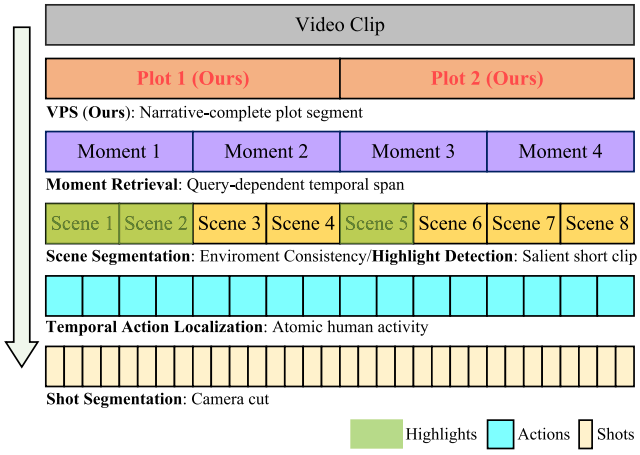


Fig. 2. Hierarchical comparison of VPS and related temporal video analysis tasks on a shared timeline: VPS spans longer ranges and emphasizes narrative continuity.

tailored methods. It faces two main challenges: (1) Plot is a complex and long-lasting concept, involving various types of information. (2) Existing video benchmarks aren't entirely suitable for VPS.

To address these challenges, we introduce MangNet, containing 1,399 five-minute video clips spanning various genres, each including 1 to 4 complete plots. Leveraging this benchmark, we further propose the Cluster Vision Transformer (CiT), a variation of the Vision Transformer (ViT) that clusters patches from video frames. This approach enables the model to focus more on the dependencies of key parts within the video in a spatio-temporal context, allowing for a better grasp of plot developments.

In summary, this work makes three main contributions:

- We formulate Video Plot Segmentation as a new video understanding task that targets the detection of temporally coherent plots with complete narrative structures.
- MangNet is introduced as a large-scale, carefully annotated benchmark covering 1,399 video clips from seven genres, designed to support systematic evaluation of plot segmentation.

Table 1
Comparison between VPS and related temporal video analysis tasks in representative benchmarks, average ground-truth duration (Avg. GT Dur.), boundary cues, narrative-continuity requirement, and semantic impact. VPS is distinguished by narrative coherence, requiring stronger continuity and greater reliance on character/object/background information, which leads to longer ground-truth segments in MangNet.

Tasks	Representative Benchmark		Boundary Cue	Narrative Continuity Required	Impact Level		
	Name	Avg. GT Dur.			Character	Object	Background
Temporal Action Localization	ActivityNet1.3 (Caba Heilbron et al., 2015)/THUMOS14 (Jiang et al., 2014)	0.80/0.07 min	Action boundary	✗	High	Low	Low
Video Highlight Detection	QVHighlights (Lei et al., 2021)	0.41 min	Saliency score	✗	High	Medium	Medium
Moment Retrieval	Charades-STA (Gao et al., 2017)	0.50 min	Query relevance	✗	Medium	Medium	Medium
Shot Segmentation	ClipShots (Tang et al., 2018)	0.26 min	Visual discontinuity	✗	Low	Low	Low
Scene Segmentation	MovieScenes (Rao et al., 2020)	0.83 min	Environmental consistency	✗	Medium	Medium	High
Video Plot Segmentation	MangNet (our proposal)	1.72 min	Narrative coherence	✓	High	High	High

- We propose CiT, a novel ViT-based architecture with cluster-aware attention, which achieves state-of-the-art performance on MangNet.

2. Related work

2.1. Video temporal segmentation

Temporal action localization:. Temporal Action Localization (TAL) aims to identify the start and end timestamps of action instances within videos. Early approaches primarily followed a two-stage framework, where temporal proposals were first generated and then classified. Pioneering works such as S-CNN (Shou et al., 2016) introduced this pipeline, which was later enhanced by R-C3D (Xu et al., 2017), integrating proposal generation and action classification into a unified end-to-end architecture, inspired by object detection methods. The SSN (Zhao et al., 2017) further refined this framework by modeling the internal structure of action instances, leading to improved localization accuracy.

With a growing emphasis on precise action boundaries, boundary-sensitive methods gained prominence. BSN (Lin et al., 2018) explicitly focused on detecting temporal boundaries, while BMN (Lin et al., 2019) introduced a boundary-matching mechanism to refine densely distributed proposals, significantly improving localization precision. G-TAD (Xu et al., 2020) further leveraged Graph Convolutional Networks (GCNs) to model temporal contextual relationships, capturing dependencies between action segments. More recently, Transformer-based architectures have demonstrated remarkable performance in TAL. ActionFormer (Zhang et al., 2022) adapted transformers to process videos of varying lengths while effectively localizing actions across different temporal scales. TriDet (Shi et al., 2023) introduced Trident-head temporal modeling combined with a sparse sampling strategy, achieving a balance between accuracy and efficiency.

However, most existing approaches primarily focus on coarse-grained, human-centric features, often overlooking the subtleties of objects, backgrounds, and their interactions, which are crucial for comprehensive video understanding.

Video highlight detection:. Video Highlight Detection focuses on identifying the most interesting or representative segments within videos, a task critical for content summarization and user engagement. Traditional approaches relied on hand-crafted audiovisual features and rule-based systems to identify important segments in videos, particularly in structured domains like sports (Ma et al., 2002; Rui et al., 2000). With the advent of machine learning, methods shifted toward supervised and weakly supervised learning frameworks that could better capture the complex relationship between visual content and highlight worthiness (Sun et al., 2014; Yang et al., 2015). Recent research has focused on self-supervised learning to reduce dependency on labeled data (Lei et al., 2021), multimodal fusion to leverage visual, audio, and textual information simultaneously (Badamdorj et al., 2021; Liu et al., 2022).

However, this task differs from ours as it merely pinpoints prominent video segments, without considering the ordinary yet narrative-contributing segments.

Moment retrieval:. Moment Retrieval aims to locate specific temporal segments within videos based on natural language queries or descriptions. Early pioneering works, such as TALL (Gao et al., 2017) and MCN (Anne Hendricks et al., 2017), established foundational frameworks by introducing cross-modal matching between video features and textual query embeddings. These approaches typically employed sliding window mechanisms to generate candidate moments and ranked them based on semantic similarity scores with the query. With advancements in deep learning, subsequent research has focused on enhancing cross-modal interaction and temporal reasoning. The 2D-TAN (Zhang et al., 2020) model introduced a two-dimensional temporal map to effectively capture temporal adjacency relationships, significantly improving localization precision. More recent approaches have incorporated Transformer architectures to capture long-range dependencies. For instance, ACRM (Tang et al., 2021) leveraged Transformer to establish frame-level correspondences between video content and language queries.

While Moment Retrieval enhances the efficiency of retrieving specific video segments based on textual input, it primarily addresses the challenge of cross-modal alignment between video and text. In contrast, Video Plot Segmentation requires models to automatically extract coherent narrative segments from videos without any textual input, focusing on understanding and delineating the storyline inherent within the visual content.

Shot and scene segmentation:. Shot segmentation identifies boundaries between camera shots, typically triggered by visual discontinuities such as hard cuts or gradual transitions (e.g., fades and dissolves), and it is commonly treated as a low-level structural parsing problem with mature benchmarks and evaluation protocols (Mühling et al., 2007; Tang et al., 2018; Xu et al., 2016). Scene segmentation further groups consecutive shots into higher-level scenes by exploiting contextual/setting consistency (e.g., location, environment, and persistent visual context), and has been studied with both classical and learning-based formulations (Baraldi et al., 2015; Islam et al., 2023; Rao et al., 2020; Sidiropoulos et al., 2011; Xu et al., 2024; Yang et al., 2023; Zabih et al., 1995),



Fig. 3. Illustration of positive and negative examples for plot annotation guidance. Solid boxes denote correct plot segments (complete performance plots), while blue dashed boxes mark common errors where brief audience reaction shots are treated as plot boundaries. *Since these shots do not break narrative continuity or performer consistency, they should not be used to split a complete performance into multiple plots.*

using datasets such as OVSD (Rotman et al., 2017) and MovieScenes (Rao et al., 2020). While shot/scene segmentation primarily defines boundaries via structural or contextual changes, VPS targets narrative-coherent plot units that may span multiple shots and scenes, where boundaries are determined by storyline continuity (e.g., a coherent activity involving consistent characters/objects/background) rather than camera or setting changes alone.

Story and chapter segmentation:. Story/chapter segmentation aims to divide videos into larger thematic or narrative units, often at a coarser granularity than scenes, but practical formulations and supervision vary across domains (e.g., news programs, documentaries, episodic videos). Existing approaches typically rely on additional structural signals (e.g., anchors, titles, program structure) or multi-modal cues when available. Representative works such as (Chen et al., 2021a; Zhai et al., 2005) focus on news broadcasts and leverage textual captions to separate stories, which is effective in that domain but differs from VPS. In contrast, VPS operates in a visual-only setting and segments videos into coherent plot units without requiring textual inputs, emphasizing narrative continuity that can persist across multiple scenes and shots.

Narrative event boundary detection:. Narrative event boundary detection has often been studied with textual narratives, e.g., Wang et al. (2022), where event boundaries are inferred from text/context. In videos, Generic Event Boundary Detection (GEBD) targets event transitions by modeling local temporal changes in visual content, with representative offline and online methods such as DDM-Net (Tang et al., 2022) and ESTimator (Jung et al., 2025). Concretely, DDM-Net computes multi-scale frame differences from multi-level features to form dense difference maps (DDM) that emphasize motion/transition cues, while ESTimator performs online detection by capturing subtle frame-to-frame discrepancies (via a Consistent Event Anticipator) and adaptively adjusting the decision threshold (via an Online Boundary Discriminator). Compared with GEBD, which primarily emphasizes local transition cues, VPS focuses on plot-level segmentation that requires modeling long-range narrative coherence across multiple events.

2.2. Transformer-based approaches

Cluster-based transformer:. Incorporating clustering into transformer variants has proven effective for enhancing feature utilization. For instance, Yu et al. (2022a,b) treat object queries as cluster centers to group pixels in panoptic segmentation. Li et al. (2021) propose a Clustered Spatio-Temporal Transformer (CSTT) for Group Activity Recognition, dynamically clustering individuals to enhance semantic representations. This suggests that using clustering principles can effectively aggregate diverse semantic information in Video Plot Segmentation.

Spatio-temporal transformer:. Capturing both spatial and temporal information is crucial for video understanding, typically addressed by spatio-temporal transformers. Li et al. (2021) innovatively substitute fully connected attention with cluster-based attention, introducing intra-group attention and inter-group attention. Bertasius et al. (2021) discover that divided space-time attention is more effective, providing superior performance. Arnab et al. (2021) present ViViT, employing self-attention at consistent frame locations to reduce long-token sequence dependencies in videos. Since our plot segmentation task requires deep consideration of features, an advanced model beyond existing ones is demanded.

3. Video plot segmentation

3.1. Definition of plot

According to the definition in the book *Writing Blockbuster Plots* (Alderson, 2016), the plot is described as the way in which external dramatic actions shape the emotional development of characters through cause and effect over time, providing narrative structure and direction. In this research, we define a plot as a segment with complete narrative and logical consistency, characterized by distinct story beginning and ending time points. A single video may contain multiple such plots, each with its independent story content.

3.2. Task formulation

In Video Plot Segmentation, we sample T frames from the original video to create the input $X \in \mathbb{R}^{T\times H\times W\times C}$. This input is encoded by a neural network, producing M proposed plot segments, each formatted as $Y = \{s_i, e_i\}_{i=1}^{M}$, where s_i and e_i are the start and end times of each plot segment. Since M often exceeds the actual number of plot segments, we utilize the Hungarian algorithm to find the most fitting predicted result for each actual plot segment. Finally, these predicted results and their corresponding ground truths are used to calculate the loss function.

4. MangNet benchmark

4.1. Benchmark reliability

Understanding the plot is essential for our task. Existing works on video temporal tasks differ fundamentally from plot segmentation, and there are no dedicated benchmarks for Video Plot Segmentation. To create a reliable benchmark, we download numerous videos from the renowned platform Mango TV, where permissible.

In accordance with the existing videos and the plot definition outlined in Section 3.1, we first annotated 10 typical positive and negative plot examples to establish consistent annotation criteria. Fig. 3 shows a representative clip illustrating both correct plot segmentation and common incorrect cases, which we used to guide annotators during labeling. Based on these examples, we then conducted discussions and training sessions with recruited volunteer video editors to calibrate their understanding before large-scale annotation.

Subsequently, each remaining video was assigned to up to three video experts for plot annotation through the following steps:

1. **Independent annotation**: Two professional video editors independently selected plot segments containing complete story content,

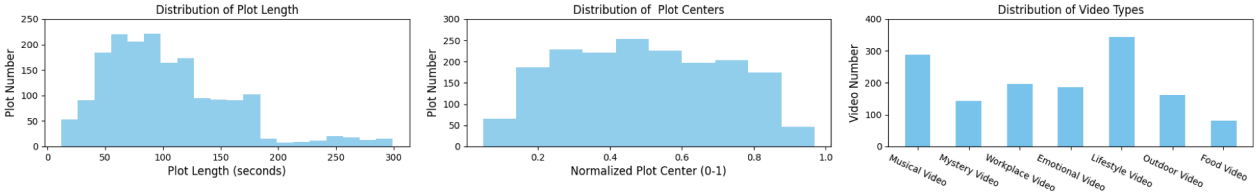


Fig. 4. Distribution of plot length (left), plot centers normalized by video duration (middle), and video types (right).

labeled as $A = \{s_1, e_1\}$ and $B = \{s_2, e_2\}$, based on the plot definition in Section 3.1 and the provided plot samples. The Intersection over Union (IoU) of the two time windows A and B was calculated as:

$$IoU = \frac{(A \cap B)}{(A \cup B)}. \tag{1}$$

2. **Consistency assessment**:
   - High Consistency: If $IoU \geq 0.9$, the earliest start time $min(s_1, s_2)$ and the latest end time $max(e_1, e_2)$ were taken as the final start and end times of the plot.
   - Significant Discrepancy: If $IoU < 0.7$, it was considered a significant divergence, and Step 1) was repeated.
   - Moderate Discrepancy: If $0.7 \leq IoU < 0.9$, a third editor was consulted to make a comprehensive judgment and provide the final plot start and end times.
3. **Majority voting**: The annotations from the three editors (if applicable) were subjected to majority voting. If at least two editors selected the same plot interval, that interval was designated as the final plot segment. Otherwise, Step 1) was repeated.
4. **Conflict resolution**: If a video segment could not reach a consensus after three rounds of annotation, it was deemed a contentious segment. A collective discussion was then held to determine the final annotation, or the video was directly excluded.

4.2. Benchmark statistics

We have collected 1399 5-min video clips across various genres, including Musical, Mystery, Workplace, Emotional, Lifestyle, Outdoor, and Food Videos. Each clip contains 1 to 4 complete plot segments, totaling 1801 plots, with an average duration of 103 s per plot. Fig. 4 shows the distribution of plot length, plot centers, and video types. Most plots are 0-200s, following a near-normal distribution. Plot centers, normalized from 0 to 1, show an even distribution across the video duration, indicating MangNet's consistency and reduced susceptibility to temporal bias - where GTs tend to occur nearer to the beginning of videos. The diversity of video types highlights a rich array of narratives. We split the benchmark into Train, Validation, and Test sets in an 8:1:1 ratio, forming the MangNet benchmark.

5. Design of CiT

5.1. Motivation and overview

Motivated by the observation that *plot continuity is reflected in the consistency of characters, objects, and background across scenes*, our model emphasizes such spatio-temporal coherence for storyline-aware segmentation. To leverage the transformer's advanced feature-capturing ability in video temporal tasks, we explore foundational applications of transformers in computer vision (i.e., ViT (Dosovitskiy et al., 2020) and TimeSformer (Bertasius et al., 2021)), and designed our CiT model based on these insights.

Fig. 5 demonstrates the CiT framework. Given an input video, we sample T frames with dimensions $H(Height) \times W(Width) \times C(Channels)$, and divide each frame into N square patches of size $L \times L$. Inspired by Xu et al. (2022), we then initialize K learnable cluster tokens to zero and concatenate them with the patches, forming the input tensor $X \in \mathbb{R}^{T \times (K+N) \times D}$, where D denotes the hidden dimensionality. In *Transformer Layers*, we perform self-attention on each frame's cluster tokens and patches spatially, and on same-position patches across frames temporally, capturing initial spatial data and forming basic spatio-temporal connections among patches. We then use a series of *Cluster-Attention (CA) blocks* to further grasp spatio-temporal information. During the Cluster Stage, we merge patches using cluster tokens based on similarity and update the cluster tokens with the merged patches. In the Separate Attention Stage, we use the updated cluster tokens to represent frame features, computing self-attention for them in both spatial and temporal dimensions (refer to Section 5.2 for details). Finally, we average the last CA block's output spatially, inputting it into the *decoder* and *span&classification heads*, yielding the prediction results.

Algorithm 1 Cluster assignment.

```
#B: batch size; K: numer of cluster tokens;
#N: number of patche tokens; D: hidden dimensionality;
#H: number of heads
function init():
  q1, k1, q2, k2, v = nn.Linear(D, D, bias = True)
  S = 1/sqrt(D)
function forward(C, P, A'):
  B, K, D = C.shape
  B, N, D = P.shape
  Q1 = q1(C).reshape(B, K, H, D/H)
  Q1 = Q1.permute(0, 2, 1, 3)
  K1 = k1(P).reshape(B, N, H, D/H)
  K1 = K1.permute(0, 2, 1, 3)
  Q2 = q2(C).reshape(B, K, H, D/H)
  Q2 = Q2.permute(0, 2, 1, 3)
  K2 = k2(P).reshape(B, N, H, D/H)
  K2 = K2.permute(0, 2, 1, 3)
  V = v(P).reshape(B, N, H, D/H)
  V = V.permute(0, 2, 1, 3)
  M1 = matmul(Q1, K1.transpose(-2, -1)) × S
  M2 = matmul(Q2, K2.transpose(-2, -1)) × S
  if A' is not None then
      M1 = M1 + A'
  end if
  A1 = gumbel_softmax(M1, hard = True, dim = -2)
  A2 = softmax(M2, dim = -1)
  A = A1 + A2
  new_C = matmul(A, V).transpose(1, 2)
  new_C = new_C.reshape(B, K, D)
    return new_C
```

5.2. Cluster-attention blocks

In our approach, Cluster-Attention blocks are pivotal for capturing the spatio-temporal features of video, including two critical stages:

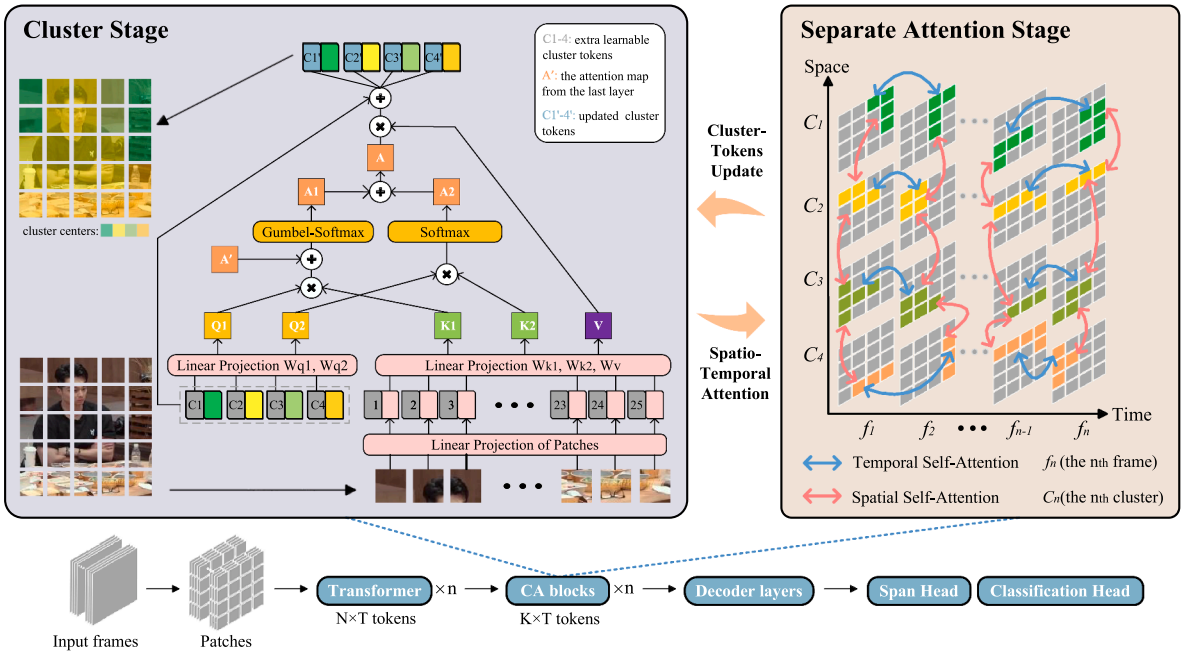


Fig. 5. The framework of CiT. It processes video frames by segmenting them into patches. Through Transformer Layers and Cluster-Attention (CA) blocks, the model captures and refines spatio-temporal relationships among patches, leveraging both spatial and temporal self-attention mechanisms. The processed data is then fed into the Decoder and Span&Classification heads, resulting in the final result.

Cluster stage:. Inspired by Visual Transformer (Wu et al., 2020) and CMT-Deeplab (Yu et al., 2022a), we have devised a similarity-based tokenizer for clustering, as depicted in Cluster Stage of Fig. 5. Given the input tensor $X \in \mathbb{R}^{T\times(K+N)\times D}$, cluster tokens are defined as $C = X[:,:K] \in \mathbb{R}^{T\times K\times D}$ and patch tokens as $P = X[:,K:] \in \mathbb{R}^{T\times N\times D}$, where $C_{ij}\{i = 1,2,...,T; j = 1,2,...,K\}$ represents the j^{th} cluster token of the i^{th} frame, and $P^{ij}\{i = 1,2,...,T; j = 1,2,...,N\}$ represents the j^{th} patch of the i^{th} frame. To perform a similarity-based tokenizer, we initially compute two attention maps as

$$A_1 = \underset{K}{gumbel_softmax}(W_{q1}C \times (W_{k1}P)^T + A'), \tag{2}$$

$$A_2 = \underset{N}{softmax}(W_{q2}C \times (W_{k2}P)^T), \tag{3}$$

where $W_{q1}, W_{k1}, W_{q2}, W_{k2}$ represent the linear projection weights, A' denotes the attention map from the last layer, and A_1 is the attention map assigning each patch to a specific cluster center. To perform the assignment, we need to compute cluster-wise argmax, which is a non-differentiable operation (Yu et al., 2022b). Therefore, we employ Gumbel-Softmax (Jang et al., 2016) for hard assignments while ensuring the diffrentiability. A_2 represents the conventional cross-attention between cluster tokens and patch tokens. Following the approach in CMT-Deeplab (Yu et al., 2022a), to mitigate the high sparsity issue in matrix A_2, we simply add A_1 and A_2 together to obtain the final attention map A: $A = A_1 + A_2$.

To understand how cluster tokens focus on high-level semantics better, we have visualized the matrix A in Fig. 9. Once we have the final attention map A, we can then update the cluster tokens using the original patch tokens, yielding a new set of cluster tokens C':

$$C' = C + A \times W_v P, \tag{4}$$

where W_v is the linear projection weight and $C' \in \mathbb{R}^{T\times K\times D}$. In Alg. 1, we show in pseudo-code how to compute the cross-attention between cluster tokens and patch tokens in the Pytorch framework.

Separate attention stage:. Inspired by ViT, we employ K cluster tokens instead of a single classification token within each frame to represent the image and its high-level semantic features. Therefore, instead of performing self-attention on each patch of the frame, we conduct self-attention solely on cluster tokens, significantly reducing the computational complexity compared to the original TimeSformer architecture. Fig. 6 illustrates how we model the cluster tokens across temporal and spatial scales, refining feature representations for decoder input. Specifically, we apply self-attention on the cluster tokens along both the T (Temporal) and S (Spatial) dimensions:

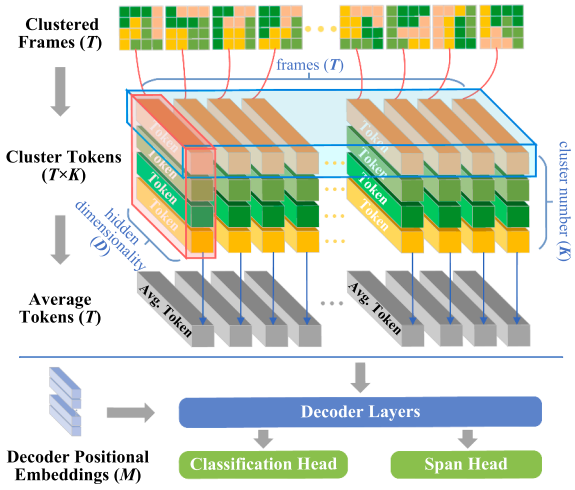


Fig. 6. After mapping T clustered frames into $T \times K$ cluster tokens, self-attention is computed for tokens of different types within the same frame (pink mask), as well as for tokens of the same type across different frames (blue mask). After that, the tokens are averaged for accurate feature representations. Finally, adding decoder positional embedding, T averaged tokens are input sequentially into the decoder and classification/span head for prediction results.

$$C_{t'}^{l} = \underset{T}{softmax}(W_{qt}C^{l} \times (W_{kt}C^{l})^T) \times W_{vt}C^{l}, \tag{5}$$

$$C_{t}^{l} = C_{t'}^{l} + C^{l}. \tag{6}$$

We first apply self-attention across cluster tokens of the same type over the temporal dimension, incorporating residual connections (He et al., 2016) between $C_{t'}^{l}$ and the input cluster tokens C^{l}, where l denotes the l^{th} layer. Next, we calculate the self-attention between cluster tokens of different types over the spatial dimension:

$$C_s^l = \underset{K}{softmax}(W_{qs}C_t^l \times (W_{ks}C_t^l)^T) \times W_{vs}C_t^l, \tag{7}$$

$$C_{new}^l = C_t^l + C_s^l. \tag{8}$$

$W_{qt}, W_{kt}, W_{vt}, W_{qs}, W_{ks}, W_{vs}$ in Eqs. (5) and (7) represent the linear projection weights. We take C_t^l as the input for spatial self-attention, introducing a residual connection between C_t^l and C_s^l to obtain the updated cluster tokens C_{new}^l. Since multiple CA blocks are stacked together, we need to restore the cluster tokens to their original dimensions for subsequent CA blocks. This is achieved by applying Eq. (9) to integrate information within the cluster tokens and perform dimension transformation:

$$P^{l+1} = Sigmoid(MLP(LN(P^l))) \times Linear(C_{new}^l), \tag{9}$$

$P^l, P^{l+1} \in \mathbb{R}^{N\times D}$ represent the patches for the $\{l\}^{th}$ and $\{l+1\}^{th}$ layers, with LN and $Linear$ denote the layer normalization and linear layer, respectively. Following ViT, after the last CA block, we average the K cluster tokens of each frame to obtain a global feature and feed them into the decoder layers for subsequent prediction tasks:

$$C_{avg} = \frac{\sum_{i=1}^{K} C_{ij}^{last}}{K}\{j = 1, 2, ..., T\} \in \mathbb{R}^{T\times D}. \tag{10}$$

5.3. Plot predictions

Utilizing Cluster-Attention (CA) blocks, we significantly augment the model's proficiency in detecting plot transitions within videos. This enhancement is achieved by clustering tokens within individual frames and subsequently computing attention on these cluster tokens across both temporal and spatial dimensions. Consequently, when predicting the plot, we can achieve good performance by directly utilizing the Transformer decoder as described in Vaswani et al. (2017). The standard decoder calculates the cross-attention between C_{avg} and the decoder positional embeddings $Q_{emb} \in \mathbb{R}^{M\times D}$, yielding the decoder output $D_{out} \in \mathbb{R}^{M\times D}$, where M represents the number of the model outputs in our configuration. Given the D_{out}, we employ both the span head (a 3-hidden-layer MLP with sigmoid activation) and the classification head (a simple linear layer) to generate predictions for the plots and corresponding classification labels for each plot, respectively:

$$\hat{y} = sigmoid(MLP(D_{out})) \in \mathbb{R}^{M\times 2}, \tag{11}$$

$$\hat{l} = Linear(D_{out}) \in \mathbb{R}^{M\times 2}. \tag{12}$$

Let $\hat{y}$ represent the predicted plots, given by $\hat{y}_i = \{\hat{c}_i, t\hat{w}_i\}_{i=1}^{M}$, where $\hat{c}_i, \hat{w}_i \in [0, 1]$ respectively represent the normalized center coordinate and width of the i^{th} plot in a video. The classification label is denoted by $\hat{l}$, where $\hat{l}_i = \{\hat{p}_i^{tp}, \hat{p}_i^{tn}\}_{i=1}^{M}$. Here, $\hat{p}_i^{tp}$ and $\hat{p}_i^{tn}$ represent the probabilities of the i^{th} plot being a true positive and a true negative, respectively.

5.4. Computational complexity

Given that the computational complexity of transformers mainly arises from the self-attention mechanism, we do not consider the linear transformation and subsequent MLP module complexities when analyzing the model complexity. For a video input denoted as $X \in \mathbb{R}^{T\times N\times D}$, in the case of conventional spatio-temporal ViT models like TimeSformer, the spatial self-attention complexity is $\mathcal{O}(TN^2D)$, the temporal self-attention complexity is $\mathcal{O}(T^2ND)$, resulting in the overall time complexity of:

$$\mathcal{O}(TimeSformer) = \mathcal{O}(TN^2D + T^2ND). \tag{13}$$

For CA block, the Cluster Stage involves computing the cross-attention between K cluster tokens and N patch tokens, with a complexity of $\mathcal{O}(TKND)$. Subsequently, in the Separate Attention Stage, self-attention is calculated individually for these K cluster tokens in both the temporal and spatial dimensions, with complexities of $\mathcal{O}(T^2KD)$ and $\mathcal{O}(TK^2D)$, respectively. Therefore, the overall attention complexity is:

$$\mathcal{O}(CA\ block) = \mathcal{O}(TKND + T^2KD + TK^2D). \tag{14}$$

In most cases, $T \le N$ and $K \ll N$. Therefore, we can obtain:

$$\mathcal{O}(TimeSformer) \sim \mathcal{O}(TN^2D), \tag{15}$$

$$\mathcal{O}(CA\ block) \sim \mathcal{O}(TKND). \tag{16}$$

Clearly, CA blocks of CiT significantly reduce computational complexity compared to TimeSformer, and subsequent experiments have demonstrated that it also achieves better predictive performance.

5.5. Training

Our objective is to draw inspiration from prior successful studies (Lei et al., 2021) and design effective loss functions to enhance the training efficiency of CiT.

Plot loss. We use plot loss $\mathcal{L}_{plot}$ to assess the disparity between predicted plot $\hat{y}_i$ and ground truth y_i. L_1 loss and L_{iou} loss are employed to respectively account for the absolute and relative errors between $\hat{y}_i$ and y_i:

$$\mathcal{L}_{plot}(y_i, \hat{y}_{\sigma(i)}) = \lambda_{L1}||y_i - \hat{y}_{\hat{\sigma}(i)}|| + \lambda_{iou}\mathcal{L}_{iou}(y_i, \hat{y}_{\hat{\sigma}(i)}), \tag{17}$$

where λ_{L1}, $\lambda_{iou} \in \mathbb{R}$ are hyperparameters, and $\hat{y}_{\sigma(i)}$ is the i^{th} predicted plot under a certain permutation $\sigma \in \mathfrak{S}_M$.

Bipartite matching. As the number of predicted results M is significantly larger than the number of ground truth instances, we employ the Hungarian algorithm to optimally match each ground truth moment with a corresponding prediction. The matching cost is defined similarly to that in Carion et al. (2020):

$$\mathcal{L}_{match} = -\hat{p}_{\sigma(i)}^{tp} + \mathcal{L}_{plot}(y_i, \hat{y}_{\sigma(i)}). \tag{18}$$

Using this matching cost, we apply the Hungarian algorithm to achieve optimal bipartite matching:

$$\hat{\sigma} = \underset{\sigma\in\mathfrak{S}_M}{argmin} \sum_{i=1}^{M} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)}). \tag{19}$$

Final loss. Given the optimal bipartite matching $\hat{\sigma}$, the selected predictions are employed to calculate the final loss function:

$$\mathcal{L} = \sum_{i=1}^{M}[-\lambda_{cls} log\hat{p}_{\hat{\sigma}(i)}^{tp} + \mathcal{L}_{plot}(y_i, \hat{y}_{\sigma(i)})], \tag{20}$$

where λ_{cls} is the hyperparameter for classification loss $log\hat{p}_{\hat{\sigma}(i)}^{tp}$. The final loss is applied to every decoder layer.

6. Experiments

6.1. Implementation details

In this paper, we set CiT with the input frames size $T \cdot H \cdot W = 150 \times 224 \times 224$, patch number $N = 196$, cluster tokens number per frame $K = 16$, and hidden dimensionality $D = 384$. Both the transformer and CA block layers have 3 layers, while the decoder has 2 layers with $M = 32$ positional embeddings. CiT is trained with AdamW (Loshchilov & Hutter, 2017) (learning rate $= 1e-4$, weight decay $= 1e-4$) and Cosine Annealing scheduler (Loshchilov & Hutter, 2016) (warmup $= 5$ epoches) on `train` split. Hyperparameters λ_{cls}, λ_{L1} and λ_{iou} in final loss are 4, 10 and 1. The training batch size $= 4$ and the training epochs $=$ 200.

Referring to the settings in Lei et al. (2021), we report mean average precision (mAP) results with IoU thresholds of 0.5, 0.6, 0.7, 0.8, 0.9, and compute the average mAP for IoU thresholds ranging from [0.5 : 0.05 : 0.95]. All comparative experiments are conducted on `val` and `test` splits of MangNet with an NVIDIA A800.

Table 2
Comparison with the state-of-the-art models on MangNet *val* split. The best scores are in **bold**, while the second-best underlined.

Methods	Backbone	Pretrain	@0.5	@0.6	@0.7	@0.8	@0.9	Avg.	95% CI Interval (Avg.)
BMN (Lin et al., 2019)	✓	✗	40.17 ± 0.29	29.25 ± 0.82	19.14 ± 0.62	10.96 ± 1.13	6.05 ± 1.54	18.78 ± 0.42	[18.21, 19.36]
Moment-DETR (Lei et al., 2021)	✓	✗	35.12 ± 0.74	26.76 ± 1.63	22.77 ± 0.47	14.26 ± 2.39	6.18 ± 0.12	19.02 ± 0.39	[18.48, 19.55]
G-TAD (Xu et al., 2020)	✓	✗	51.58 ± 0.54	36.06 ± 0.02	26.44 ± 0.74	16.00 ± 0.24	7.73 ± 2.32	24.53 ± 0.93	[23.24, 25.82]
TimeSformer (Bertasius et al., 2021)	✗	✓	56.61 ± 1.63	40.21 ± 0.92	26.90 ± 0.32	17.12 ± 1.53	6.75 ± 1.13	26.65 ± 0.46	[26.13, 27.17]
TimeSformer-HR (Bertasius et al., 2021)	✗	✓	57.45 ± 1.44	43.14 ± 0.16	26.97 ± 2.72	16.05 ± 1.27	7.35 ± 0.04	26.96 ± 0.78	[25.88, 28.05]
RTD-Net (Tan et al., 2021)	✓	✗	53.61 ± 1.82	40.22 ± 2.15	26.77 ± 1.50	14.81 ± 1.63	6.34 ± 1.89	25.29 ± 1.24	[23.88, 26.69]
TCANet (Qing et al., 2021)	✓	✗	50.90 ± 2.57	40.44 ± 0.25	28.48 ± 1.43	18.60 ± 2.64	6.30 ± 0.71	26.20 ± 1.20	[24.53, 27.87]
ActionFormer (Zhang et al., 2022)	✓	✗	57.95 ± 0.81	43.28 ± 1.48	27.13 ± 2.07	14.82 ± 2.51	3.75 ± 1.05	26.49 ± 0.77	[25.62, 27.37]
UniVTG (Lin et al., 2023)	✓	✗	39.64 ± 2.02	31.79 ± 1.35	16.55 ± 0.17	8.04 ± 0.76	3.40 ± 1.27	18.10 ± 0.74	[17.08, 19.13]
TriDet (Shi et al., 2023)	✓	✗	54.13 ± 2.06	43.11 ± 1.20	30.52 ± 0.03	17.50 ± 0.88	5.86 ± 0.81	27.16 ± 0.54	[26.42, 27.90]
CG-DETR (Moon et al., 2023)	✓	✗	58.39 ± 0.42	41.54 ± 3.65	28.83 ± 3.47	17.14 ± 0.89	6.31 ± 0.34	27.62 ± 1.34	[25.77, 29.48]
DoraemonGPT (Yang et al., 2024)	✗	✗	19.03 ± 1.78	11.50 ± 1.96	6.84 ± 0.36	4.14 ± 1.28	2.13 ± 0.75	7.65 ± 0.44	[7.15, 8.15]
LLaVa-Video-7B (Zhang et al., 2024)	✗	✗	6.73 ± 0.63	2.45 ± 0.23	0.61 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	1.53 ± 0.15	[1.36, 1.70]
VideoXL-7B (Shu et al., 2025)	✗	✗	16.14 ± 0.26	8.62 ± 0.34	5.12 ± 0.10	1.62 ± 0.09	0.40 ± 0.03	5.44 ± 0.06	[5.37, 5.50]
CiT (Ours)	✗	✗	**58.48** ± 0.54	**44.98** ± 0.74	**30.80** ± 0.36	**22.37** ± 0.44	**11.81** ± 1.47	**29.77** ± 0.47	[**29.12**, **30.42**]

Table 3
Comparison with the state-of-the-art models on MangNet *test* split. The best scores are in **bold**, while the second-best underlined.

Methods	Backbone	Pretrain	@0.5	@0.6	@0.7	@0.8	@0.9	Avg.	95% CI Interval (Avg.)
BMN (Lin et al., 2019)	✓	✗	37.70 ± 0.42	27.88 ± 0.29	16.58 ± 2.32	8.40 ± 2.02	3.80 ± 0.90	16.86 ± 0.72	[16.05, 17.67]
Moment-DETR (Lei et al., 2021)	✓	✗	39.14 ± 1.39	29.27 ± 0.80	18.57 ± 0.75	11.53 ± 0.30	4.43 ± 0.50	18.69 ± 0.29	[18.28, 19.09]
G-TAD (Xu et al., 2020)	✓	✗	54.88 ± 0.74	39.44 ± 1.99	25.53 ± 3.71	11.96 ± 1.12	4.96 ± 1.04	24.31 ± 1.04	[22.87, 25.76]
TimeSformer (Bertasius et al., 2021)	✗	✓	52.12 ± 2.97	36.96 ± 0.21	22.54 ± 1.43	10.86 ± 0.64	5.09 ± 0.47	22.79 ± 0.30	[22.38, 23.20]
TimeSformer-HR (Bertasius et al., 2021)	✗	✓	55.82 ± 2.76	42.85 ± 1.09	26.61 ± 0.18	12.47 ± 0.42	5.28 ± 1.05	25.89 ± 0.04	[25.83, 25.95]
RTD-Net (Tan et al., 2021)	✓	✗	**60.16** ± 0.56	38.13 ± 0.72	21.27 ± 0.24	12.64 ± 1.46	2.17 ± 0.13	23.68 ± 0.38	[23.15, 24.21]
TCANet (Qing et al., 2021)	✓	✗	58.94 ± 0.99	45.02 ± 2.74	27.55 ± 1.85	15.18 ± 1.31	5.17 ± 1.29	27.01 ± 0.83	[25.86, 28.15]
ActionFormer (Zhang et al., 2022)	✓	✗	53.42 ± 0.31	37.94 ± 2.81	27.26 ± 0.58	14.73 ± 1.56	6.23 ± 0.45	25.02 ± 0.92	[23.75, 26.29]
UniVTG (Lin et al., 2023)	✓	✗	36.67 ± 0.17	25.84 ± 0.67	14.29 ± 0.50	8.57 ± 0.51	2.67 ± 0.76	15.59 ± 0.09	[15.46, 15.71]
TriDet (Shi et al., 2023)	✓	✗	54.16 ± 1.52	42.12 ± 0.18	25.66 ± 0.10	12.29 ± 1.61	3.87 ± 1.08	24.89 ± 0.08	[24.79, 25.00]
CG-DETR (Moon et al., 2023)	✓	✗	54.27 ± 2.65	40.88 ± 3.38	28.88 ± 0.10	15.56 ± 1.05	4.77 ± 2.84	26.01 ± 1.29	[24.34, 27.68]
DoraemonGPT (Yang et al., 2024)	✗	✗	24.88 ± 3.37	10.22 ± 2.08	4.45 ± 0.78	2.55 ± 0.07	0.00 ± 0.00	7.46 ± 0.83	[6.32, 8.61]
LLaVa-Video-7B (Zhang et al., 2024)	✗	✗	4.50 ± 0.59	2.78 ± 0.42	0.66 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	1.38 ± 0.14	[1.22, 1.54]
VideoXL-7B (Shu et al., 2025)	✗	✗	14.37 ± 0.66	7.80 ± 0.29	3.57 ± 0.37	1.76 ± 0.30	0.00 ± 0.00	4.73 ± 0.18	[4.53, 4.93]
CiT (Ours)	✗	✗	59.68 ± 0.79	**47.27** ± 1.61	**29.09** ± 0.71	**16.37** ± 0.10	**6.95** ± 0.79	**28.80** ± 0.23	[**28.49**, **29.11**]

Table 4
Ablation studies on model hyperparameters.

# Encoder layers	@0.5	@0.75	Avg.
2	54.64	20.61	26.81
6	**58.86**	**25.89**	**30.10**
12	53.43	19.13	25.50
# Input Frames	@0.5	@0.75	Avg.
50	55.69	23.76	27.02
75	55.93	22.98	27.14
150	**58.86**	**25.89**	**30.10**
300	51.46	23.58	26.89
# Cluster Tokens	@0.5	@0.75	Avg.
8	54.03	21.35	26.42
16	**58.86**	**25.89**	**30.10**
32	57.48	21.84	27.20
64	56.86	24.38	28.44
# Proposed Plots	@0.5	@0.75	Avg.
16	55.67	21.00	26.25
32	**58.86**	**25.89**	**30.10**
48	53.10	19.44	25.26

Table 5
Ablation studies of the cluster stage (C-Stage) and separate attention stage (SA-Stage) within the CA block.

C-Stage	SA-Stage	@0.5	@0.75	Avg.
✗	✗	56.10	23.05	27.46
✓	✗	56.33	20.79	27.79
✗	✓	52.3	23.85	27.45
✓	✓	**58.86**	**25.89**	**30.10**

6.2. Baselines

Given our task's novelty, existing models are not directly suitable for segmenting video plots. To fairly evaluate our proposed CiT, we adapt several state-of-the-art models from related temporal video analysis tasks as baselines. Specifically, for these adapted baselines, we retain their original feature extraction modules in the encoder and standardize the decoders to match the VPS output format, so that all baselines produce frame-level plot boundaries under the same training objective. For consistency across *adapted baselines*, we use SlowFast (Feichtenhofer et al., 2019) as the shared feature backbone and feed the extracted features into each baseline encoder; for TimeSformer and TimeSformer-HR, we initialize them with pre-trained weights from Kinetics-600 (K600). In contrast, CiT does not use SlowFast or any feature pre-extraction backbone: it directly processes raw video frames and learns spatio-temporal representations end-to-end for plot boundary prediction.

In addition, we include representative Video-LLM and agentic video understanding baselines, namely Doraemon-GPT (Yang et al., 2024), LLaVA-Video-7B (Zhang et al., 2024), and VideoXL-7B (Shu et al., 2025). Since these models do not natively output plot boundaries, we adopt a unified prompting protocol (Fig. 7) that instructs each model to produce plot segments in a strict JSON format, i.e., a list of start, end intervals in integer seconds. For evaluation, we match each predicted interval to

```
You are an expert video editor. Your task is to perform Video Plot
Segmentation (VPS) on the given video.

Definition (Plot Segment):
A plot segment is a temporally coherent semantic unit with consistent
main participants, key objects, and overall context. A boundary occurs
when these cues change in a sustained way. Brief cutaways or reaction
shots should NOT be treated as boundaries if the surrounding content
remains continuous.

Video:
- Duration: 300 seconds.
- You must output plot segments with START and END
  times in INTEGER seconds.
- A video may contain 1, or multiple plot segments.
  Determine the number of segments by yourself.

Output requirements (STRICT):
1) Output ONLY valid JSON (no markdown, no extra text).
2) The JSON must follow this schema exactly:
    {
     "video_duration_sec": 300,
     "segments": [
      {"start": <int>, "end": <int>},
      ...
     ]
    }
3) Constraints:
- 0 <= start < end <= 300
- start and end must be integers (seconds).
- Segments must be non-overlapping and sorted by start ascending.

Procedure (do internally, but do not output reasoning):
- Watch the full video.
- Identify major plot units based on narrative/thematic coherence.
- Decide boundaries at integer seconds.

Now, watch the video carefully and return the JSON result.
```

Fig. 7. Unified prompt for video-LLM plot segmentation (JSON Output).

the ground-truth interval with the highest temporal overlap and compute mAP under different IoU thresholds as well as the averaged mAP score, following the same VPS evaluation protocol used for all methods.

6.3. Results and analysis

We conduct an extensive comparison with state-of-the-art temporal detection models on MangNet. Table 2 reports results on the *val* split. CiT achieves the best overall performance, obtaining the highest Avg mAP and consistently strong mAP across IoU thresholds. The advantage is more pronounced at higher IoUs, indicating that CiT yields more accurate plot boundaries rather than only coarse localization. We attribute this improvement to our plot-oriented design, which encourages consistent modeling of long-range spatio-temporal cues (e.g., main participants, key objects, and overall context) while preserving temporal dependencies for boundary discrimination. To improve evaluation completeness, we additionally report results on the MangNet *test* split in Table 3. The same trends hold on the test split, suggesting that the gains are not specific to the validation set.

To quantify robustness, we conduct five independent trials for all methods and report mean±std for each metric, together with a 95% confidence interval (CI) for Avg mAP estimated under a normality assumption (reported as $[L, U]$). The resulting statistics show that the relative ranking is consistent across runs, and the uncertainty ranges are sufficiently small to support stable comparisons.

We also apply interpolation to align the pre-trained model parameters with a standardized setup, such as transforming the original classification token into K cluster tokens. These adjustments may have contributed to the suboptimal performance of the pre-trained TimeSformer models.

Simultaneously, we compare the computational complexity of each layer in TimeSformer and CA Blocks, measured in GFLOPS. For the standard TimeSformer with 8 input frames and the TimeSformer-HR with 16 input frames, the complexities are 18.07 and 135.76 GFLOPS, respectively. In contrast, for the CA Block, the complexities for the same input frame counts are 1.32 and 2.77 GFLOPS, respectively. It is evident that by reducing N patch tokens to K cluster tokens (where $K \ll N$), CiT significantly reduces complexity compared to the TimeSformer variants, while also achieving better performance.

6.4. Ablation study

We conduct ablation study focusing on model hyperparameters and main stages, measuring mAP at IoU thresholds of 0.5 and 0.75, and the average mAP across $[0.5 : 0.05 : 0.95]$, in line with (Lei et al., 2021).

Model hyperparameters:. Our analysis, detailed in Table 4, focuses on the effects of the number of encoder layers, sampled frames, cluster tokens, and proposed plots. We tweak selected hyperparameters while keeping others constant.

For the effect of encoder layers, we set the number of layers to 2, 6, and 12 respectively, with each having half Transformer layers and half CA Blocks. Table 4 reveals that CiT performs optimally when the number of encoder layers is 6, with more layers leading to expected overfitting.

To analyze the impact of different sampled frame counts, we set their numbers to 50, 75, 150, and 300 respectively. The model's performance improves with an increase in sampled frames from 50 to 150. Nevertheless, at 300 frames, there's no additional improvement. We speculate that processing more frames may require different parameter settings and a more complex network structure.

Regarding the number of candidate proposed plots M mentioned in the Section 3.2, we set M to 16, 32, and 48 respectively. Results show that model performance improves when M increased from 16 to 32, but declines when M is set to 48. This indicates that blindly increasing the number of plot segments output by the model in an attempt to cover all actual plot intervals is not advisable. Such an approach may lead to over-optimization on the training set, resulting in reduced generalization capability on the test set.

Finally, we compare how different numbers of cluster tokens affect the model's accuracy, noting that the model performs best with 16 cluster tokens. This indicates that a certain number of cluster tokens is sufficient to capture high-level semantic information.

Cluster stage and separate attention stage:. We study the impact of different combinations of two stages in the CA block on model's predictive performance, with the results presented in Table 5. When both stages are omitted, we retain only the basic layer normalization and MLP from Eq. (9) in each CA block's output. We observe that using only one of the two stages yields results comparable to not using either stage. This behavior reflects the complementary roles of the two stages. Using only the Clustering Stage reduces tokens per frame but does not sufficiently model spatio-temporal dependencies of the aggregated representations. In contrast, using only the Separate Attention Stage keeps all patch tokens, which disperses attention and weakens the focus on salient subjects. Therefore, the model performs best when these two stages work together.

Table 6
Comparison with the state-of-the-art models on TVSum.

Methods	F_1 score
SUM-GAN-sl (Apostolidis et al., 2019b)	58.0
SUM-GAN-AAE (Apostolidis et al., 2019a)	58.3
HMT (Zhao et al., 2022)	60.1
VASNet (Fajtl et al., 2018)	61.4
Trailer (Wang et al., 2020)	62.8
MMAN (Han et al., 2024)	62.4
Human	63.7
CiT (Ours)	61.3

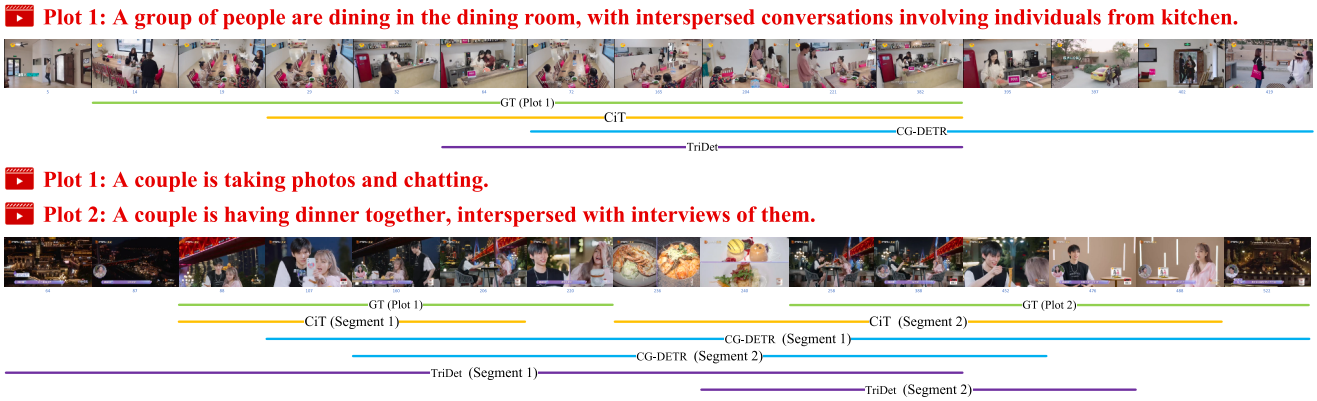


Fig. 8. Prediction results visualization. Colored underlines represent ground truth (GT) and model predictions.

6.5. Generalization study

To provide additional evidence of generalization beyond MangNet, we evaluate whether the representations learned by CiT can transfer to a representative video summarization benchmark, TVSum (Song et al., 2015). Following the standard setting, we replace the VPS decoder with a lightweight linear head that predicts frame-level importance scores, and report the F_1 score for summarization evaluation. This design keeps the backbone and the core spatio-temporal modeling unchanged, and only adapts the prediction head to match the summarization label format.

As shown in Table 6, CiT achieves an F_1 score of 61.3 on TVSum, which is competitive with representative summarization baselines and is close to the human performance reported in VASNet (Fajtl et al., 2018). This result indicates that the learned spatio-temporal representations and continuity-aware modeling principles generalize beyond MangNet, and can be leveraged for related temporal understanding objectives with only a lightweight head replacement. More broadly, it provides supporting evidence that VPS-style semantic structuring can serve as a useful intermediate representation for downstream video analysis tasks.

6.6. User study on VPS utility

We also conduct a lightweight user study to assess whether plot-level segments are perceived as practical and coherent units for human navigation. The study design aligns with the key task attributes summarized in Table 1–boundary cues, narrative continuity, and semantic impact–so that the evaluation reflects the intended distinction between VPS and existing temporal video analysis settings. Before participation, each participant has been informed of the study purpose and procedure. We recruit 15 lab volunteers; each participant spends approximately 10-15 min watching five short videos (2-3 min each), and each video contains at least one plot segment detected by VPS. After watching each full video, participants complete a forced-choice selection task: they are presented with four candidate segments and are asked to choose the one that best corresponds to a coherent plot segment. Only one option is the VPS prediction; the remaining three segments are sampled from other temporal segmentation schemes.

After finishing the selection task, participants completed a short post-study questionnaire and answered three brief interview questions. The questionnaire captured overall perceived usefulness, reduced search effort (i.e., less time and fewer interactions needed to locate relevant content), and confidence in the selected segment. It also explicitly probed perceptions aligned with Table 1: (i) whether plot boundaries are intuitive when they are not solely determined by low-level visual discontinuities (e.g., camera cuts), (ii) whether the resulting segments preserve narrative continuity and therefore support coherent browsing, and (iii) whether plot-level segments facilitate localization of query-relevant content that depends on higher-level semantics, such as consistent characters, objects, and background across a storyline. The interviews further asked participants to explain why plot segments were helpful (or not) and to describe confusing boundary cases (e.g., brief audience reaction shots, camera cuts, or scene changes).

Participants' selections showed substantial agreement with VPS outputs: 4 participants matched VPS in all 5 videos, 8 matched in 4 videos, 2 matched in 3 videos, and 1 matched in 2 videos (average agreement: 4/5 per participant, i.e., 60/75 selections). Notably, no participant consistently avoided the VPS option across the five videos. Together with the questionnaire and interview feedback, these results provide preliminary evidence that humans tend to prefer storyline-level segments that respect narrative continuity rather than boundaries driven solely by short, localized events. This trend is broadly consistent with the distinctions summarized in Table 1 and offers initial support that VPS captures storyline-level segmentation based on narrative continuity and higher-level semantic cues. As a representative comment, participant #9 noted that "*Plot segments feel like complete story units, which makes it easier to jump to the relevant part without scanning many short clips.*"

6.7. Visualization

Predicted clips:. Fig. 8 illustrates a comparison of the ground truth (GT) with the predictions generated by CiT and the other two top-performing models. The visualization displays the plot content across two representative video clips. The results demonstrate that CiT's predictions consistently align more closely with the ground truth, regardless of whether the video contains a single plot or multiple plots. While the competing

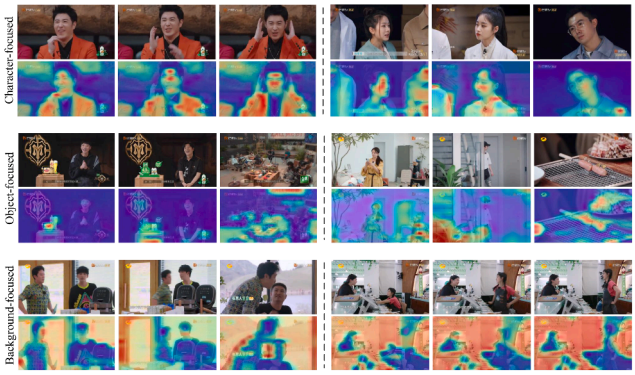


Fig. 9. Heatmap samples reveal that within the continuous contextual frames of disparate videos, CiT consistently focuses on the primary components of the video, including Character (row#1, 2), Object (row#3, 4), and Background (row#5, 6).

models show notable deviations from the GT, CiT's advantage becomes particularly pronounced in the more challenging scenario of multiple plot segments, where it exhibits significantly better plot segmentation accuracy.

Cluster attention:. To better demonstrate how the model captures high-level semantic information, we visualized the final attention map A between the cluster tokens and patch tokens in Cluster Stage from Fig. 5. The experimental results are shown in Fig. 9, where we can see that the model can actively track relevant parts across different frames.

7. Conclusion

This work introduces a novel and complex task, Video Plot Segmentation, creating the MangNet benchmark and CiT model grounded in a deep understanding of the plot narrative. Numerous experiments on MangNet demonstrate the superiority of CiT. Yet, this work also has some limitations: 1) MangNet is annotated under the strict supervision of professionals. Although some definitions can help us to clarify the structure of the plots, the definitions themselves are subjective, leading to reliance on manual annotation for benchmark expansion, which is both costly and inefficient; 2) Its lightweight and memory-limited structure hinders the direct processing of high-resolution, hour-length videos. Addressing these limitations constitutes an important direction for future research.

CRediT authorship contribution statement

Xichen Tan: Conceived and designed the study, Performed the experiments, Wrote the paper, Reviewed and edited the manuscript, Read and approved the manuscript; **Yuanjing Luo:** Wrote the paper, Reviewed and edited the manuscript, Read and approved the manuscript; **Yunfan Ye:** Reviewed and edited the manuscript, Read and approved the manuscript; **Guang Yu:** Reviewed and edited the manuscript, Read and approved the manuscript; **Chengyu Wang:** Reviewed and edited the manuscript, Read and approved the manuscript. Fang Liu: Reviewed and edited the manuscript, Read and approved the manuscript; **Zhiping Cai:** Reviewed and edited the manuscript, Read and approved the manuscript.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the Key Program of NSFC Hunan (2026JJ30028), the National Natural Science Foundation of China (62402171, 62472434, 62506374), the Department of Education of Hunan province (25B0310), and the Postgraduate Research and Innovation Project of Hunan Province (XJJC2025012).

References

Alderson, M. (2016). Writing blockbuster plots: A step-by-step guide to mastering plot, structure, and scene. Writer's Digest Books.

Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., & Russell, B. (2017). Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision* (pp. 5803–5812).

Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2019a). Unsupervised video summarization via attention-driven adversarial learning. In *International conference on multimedia modeling* (pp. 492–504). Springer.

Apostolidis, E., Metsai, A. I., Adamantidou, E., Mezaris, V., & Patras, I. (2019b). A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization. In *Proceedings of the 1st international workshop on AI for smart TV content production, access and delivery* (pp. 17–25).

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6836–6846).

Badamdorj, T., Rochan, M., Wang, Y., & Cheng, L. (2021). Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8127–8137).

Baraldi, L., Grana, C., & Cucchiara, R. (2015). Shot and scene detection via hierarchical clustering for re-using broadcast video. In *International conference on computer analysis of images and patterns* (pp. 801–811). Springer.

Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML* (p. 4). (*vol. 2*).

Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J. (2015). ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 961–970).

Cao, C., Wang, Y., Zhang, Y., Lu, Y., Zhang, X., & Zhang, Y. (2023). Co-occurrence matters: Learning action relation for temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(5), 3327–3339.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Chen, H., Wang, Z., Pei, Y., Huang, B., & Tu, W. (2021a). Story segmentation for news broadcast based on primary caption. In *Proceedings of the 2nd ACM international conference on multimedia in asia* (pp. 1–6).

Chen, J., Wang, J., Wang, X., Wang, X., Feng, Z., Liu, R., & Song, M. (2021b). CoEvo-Net: Coevolution network for video highlight detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(6), 3788–3797.

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

Fajtl, J., Sokeh, H. S., Argyriou, V., Monekosso, D., & Remagnino, P. (2018). Summarizing videos with attention. In *Asian conference on computer vision* (pp. 39–54). Springer.

Fang, D., Xu, H., Wei, W., Guizani, M., & Gao, H. (2025). Multi-modal integrated proposal generation network for weakly supervised video moment retrieval. *Expert Systems with Applications*, *269*, 126497. https://doi.org/10.1016/j.eswa.2025.126497

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6202–6211).

Gao, J., Sun, C., Yang, Z., & Nevatia, R. (2017). TALL: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision* (pp. 5267–5275).

Gao, J., & Xu, C. (2021). Learning video moment retrieval without a single annotated video. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(3), 1646–1657.

Han, D., Cheng, X., Guo, N., Ye, X., Rainer, B., & Priller, P. (2023). Momentum cross-modal contrastive learning for video moment retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(7), 5977–5994.

Han, T., Zhou, Q., Yu, J., Yu, Z., Zhang, J., & Zhao, S. (2024). Effective video summarization by extracting parameter-free motion attention. *ACM Transactions on Multimedia Computing, Communications and Applications*, *20*(7), 1–20.

Hassanien, A., Elgharib, M., Selim, A., Bae, S.-H., Hefeeda, M., & Matusik, W. (2017). Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. arXiv preprint arXiv:1705.03281, .

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hu, Y., Fu, J., Chen, M., Gao, J., Dong, J., Fan, B., & Liu, H. (2023). Learning proposal-aware re-ranking for weakly-supervised temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(1), 207–220.

Huo, S., Zhou, Y., Chen, K., & Xiang, W. (2025). Skim-and-scan transformer: A new transformer-inspired architecture for video-query based video moment retrieval. *Expert Systems with Applications*, *270*, 126525. https://doi.org/10.1016/j.eswa.2025.126525

Islam, M. M., Hasan, M., Athrey, K. S., Braskich, T., & Bertasius, G. (2023). Efficient movie scene detection using state-space transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18749–18758).

Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144, .

Jiang, Y.-G., Liu, J., Zamir, A. R., Toderici, G., Laptev, I., Shah, M., & Sukthankar, R. (2014). Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Jung, H., Kim, D., Lim, S., Son, J., & Choi, J. (2025). Online generic event boundary detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13741–13750).

Kar, T., Kanungo, P., Mohanty, S. N., Groppe, S., & Groppe, J. (2024). Video shot-boundary detection: issues, challenges and solutions. *Artificial Intelligence Review*, *57*(4), 104.

Lee, P., Kim, T., Shim, M., Wee, D., & Byun, H. (2023). Decomposed cross-modal distillation for RGB-based temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2373–2383).

Lei, J., Berg, T. L., & Bansal, M. (2021). Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, *34*, 11846–11858.

Li, S., Cao, Q., Liu, L., Yang, K., Liu, S., Hou, J., & Yi, S. (2021). GroupFormer: Group activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13668–13677).

Lin, K. Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A. J., Yan, R., & Shou, M. Z. (2023). UniVTG: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2794–2804).

Lin, T., Liu, X., Li, X., Ding, E., & Wen, S. (2019). BMN: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3889–3898).

Lin, T., Zhao, X., Su, H., Wang, C., & Yang, M. (2018). BSN: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the european conference on computer vision (ECCV)* (pp. 3–19).

Liu, H., Li, X., Fan, B., & Xu, J. (2025). BRTAL: Boundary refinement temporal action localization via offset-driven diffusion models. *IEEE Transactions on Circuits and Systems for Video Technology*, *35*(8), 8174–8186. https://doi.org/10.1109/TCSVT.2025.3548515

Liu, Y., Li, S., Wu, Y., Chen, C.-W., Shan, Y., & Qie, X. (2022). UMT: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3042–3051).

Liu, Y., Zhu, H., Ren, H., Shi, J., & Wang, D. (2024). Fusion detection network with discriminative enhancement for weakly-supervised temporal action localization. *Expert Systems with Applications*, *238*, 122000. https://doi.org/10.1016/j.eswa.2023.122000

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, .

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, .

Ma, Y.-F., Lu, L., Zhang, H.-J., & Li, M. (2002). A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on multimedia* (pp. 533–542).

Moniruzzaman, M., & Yin, Z. (2023). Collaborative foreground, background, and action modeling network for weakly supervised temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(11), 6939–6951.

Moon, W., Hyun, S., Lee, S., & Heo, J.-P. (2023). Correlation-guided query-dependency calibration in video representation learning for temporal grounding. arXiv preprint arXiv:2311.08835, .

Mühling, M., Ewerth, R., Stadelmann, T., Zöfel, C., Shi, B., & Freisleben, B. (2007). University of marburg at TRECVID 2007: Shot boundary detection and high level feature extraction. In *TRECVID*.

Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., Qiao, Y., Yan, J., Gao, C., & Sang, N. (2021). Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 485–494).

Rao, A., Xu, L., Xiong, Y., Xu, G., Huang, Q., Zhou, B., & Lin, D. (2020). A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10146–10155).

Rotman, D., Porat, D., & Ashour, G. (2017). Optimal sequential grouping for robust video scene detection using multiple modalities. *International Journal of Semantic Computing*, *11*(02), 193–208.

Rui, Y., Gupta, A., & Acero, A. (2000). Automatically extracting highlights for TV baseball programs. In *Proceedings of the eighth ACM international conference on multimedia* (pp. 105–115).

Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., & Tao, D. (2023). TriDet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18857–18866).

Shou, Z., Wang, D., & Chang, S.-F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1049–1058).

Shu, Y., Liu, Z., Zhang, P., Qin, M., Zhou, J., Liang, Z., Huang, T., & Zhao, B. (2025). Video-XL: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the computer vision and pattern recognition conference* (pp. 26160–26169).

Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., & Trancoso, I. (2011). Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(8), 1163–1177.

Song, Y., Vallmitjana, J., Stent, A., & Jaimes, A. (2015). TVSum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5179–5187).

Sul, J., Han, J., & Lee, J. (2023). Mr. Hisum: A large-scale dataset for video highlight detection and summarization. In *Thirty-seventh conference on neural information processing systems datasets and benchmarks track*.

Sun, M., Farhadi, A., & Seitz, S. (2014). Ranking domain-specific highlights by analyzing edited videos. In *Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, september 6-12, 2014, proceedings, part i 13* (pp. 787–802). Springer.

Sun, W., Su, R., Yu, Q., & Xu, D. (2022). Slow motion matters: A slow motion enhanced network for weakly supervised temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(1), 354–366.

Sun, X., Gao, J., Zhu, Y., Wang, X., & Zhou, X. (2023). Video moment retrieval via comprehensive relation-aware network. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(9), 5281–5295.

Tan, J., Tang, J., Wang, L., & Wu, G. (2021). Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13526–13535).

Tang, H., Zhu, J., Liu, M., Gao, Z., & Cheng, Z. (2021). Frame-wise cross-modal matching for video moment retrieval. *IEEE Transactions on Multimedia*, *24*, 1338–1349.

Tang, J., Liu, Z., Qian, C., Wu, W., & Wang, L. (2022). Progressive attention on multi-level dense difference maps for generic event boundary detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3355–3364).

Tang, S., Feng, L., Kuang, Z., Chen, Y., & Zhang, W. (2018). Fast video shot transition localization with deep structured models. In *Asian conference on computer vision* (pp. 577–592). Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems: vol. 30* (pp. 5998–6008). Curran Associates, Inc.

Wang, L., Liu, D., Puri, R., & Metaxas, D. N. (2020). Learning trailer moments in full-length movies with co-contrastive attention. In *European conference on computer vision* (pp. 300–316). Springer.

Wang, Z., Jafarpour, A., & Sap, M. (2022). Uncovering surprising event boundaries in narratives. In *Proceedings of the 4th workshop of narrative understanding (WNU2022)* (pp. 1–12).

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., & Vajda, P. (2020). Visual transformers: Token-based image representation and processing for computer vision. arXiv preprint arXiv:2006.03677, .

Xu, H., Das, A., & Saenko, K. (2017). R-C3D: Region convolutional 3D network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 5783–5792).

Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., & Wang, X. (2022). GroupViT: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18134–18144).

Xu, J., Song, L., & Xie, R. (2016). Shot boundary detection using convolutional neural networks. In *2016 visual communications and image processing (VCIP)* (pp. 1–4). IEEE.

Xu, M., Zhao, C., Rojas, D. S., Thabet, A., & Ghanem, B. (2020). G-TAD: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10156–10165).

Xu, Y., Pan, L., Sang, W., Luo, H., Li, L., Wei, P., & Zhu, L. (2024). Joint learning of video scene detection and annotation via multi-modal adaptive context network. *Expert Systems with Applications*, *249*, 123656.

Yang, H., Wang, B., Lin, S., Wipf, D., Guo, M., & Guo, B. (2015). Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE international conference on computer vision* (pp. 4633–4641).

Yang, Y., Huang, Y., Guo, W., Xu, B., & Xia, D. (2023). Towards global video scene segmentation with context-aware transformer. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 3206–3213). (*vol. 37*).

Yang, Z., Chen, G., Li, X., Wang, W., & Yang, Y. (2024). DoraemonGPT: Toward understanding dynamic scenes with large language models (exemplified as a video agent). In *Proceedings of the 41st international conference on machine learning* (pp. 55976–55997).

Yu, Q., Wang, H., Kim, D., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., & Chen, L.-C. (2022a). CMT-DeepLab: Clustering mask transformers for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2560–2570).

Yu, Q., Wang, H., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., & Chen, L.-C. (2022b). k-means mask transformer. In *European conference on computer vision* (pp. 288–307). Springer.

Zabih, R., Miller, J., & Mai, K. (1995). A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of the third ACM international conference on multimedia* (pp. 189–200).

Zhai, Y., Yilmaz, A., & Shah, M. (2005). Story segmentation in news videos using visual and text cues. In *International conference on image and video retrieval* (pp. 92–102). Springer.

Zhang, C.-L., Wu, J., & Li, Y. (2022). ActionFormer: Localizing moments of actions with transformers. In *European conference on computer vision* (pp. 492–510). Springer.

Zhang, D., Dai, X., Wang, X., Wang, Y.-F., & Davis, L. S. (2019). MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1247–1257).

Zhang, S., Peng, H., Fu, J., & Luo, J. (2020). Learning 2D temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 12870–12877). (*vol. 34*).

Zhang, X., Zhao, P., Ji, J., Lu, X., & Yin, Y. (2024). Video corpus moment retrieval via deformable multigranularity feature fusion and adversarial training. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(8), 6686–6698. https://doi.org/10.1109/TCSVT.2023.3294567

Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., & Li, C. (2024). Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713, .

Zhao, B., Gong, M., & Li, X. (2022). Hierarchical multimodal transformer to summarize videos. *Neurocomputing*, *468*, 360–369.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D. (2017). Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2914–2923).

Zhu, W., Huang, Y., Xie, X., Liu, W., Deng, J., Zhang, D., Wang, Z., & Liu, J. (2023). AutoShot: A short video dataset and state-of-the-art shot boundary detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2238–2247).



Xichen Tan is currently pursuing the PhD degree in School of Computer Science and Technology at the National University of Defense Technology. He received his bachelor's degree from Tsinghua University and his master's degree from Duke University. His research interests primarily focus on Multimodal Large Language Models and video understanding.



Yuanjing Luo received the PhD degree in computer science and technology from the National University of Defense Technology, Changsha in 2024. She is currently a Lecturer at the College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha, China. Her research interests are in Information Security, especially image steganography, digital watermarking, and copyright protection in deep learning. Her research has been published in multiple top-tier conferences and journals, such as WWW, AAAI and IEEE TCSVT.



Yunfan Ye is an Assistant Professor in School of Design, Hunan University, China. He earned his PhD degree in December 2023 in National University of Defense Technology, the Master's degree in Computer Science from Stevens Institute of Technology, and Bachelor's degree in Computer Science in from Xiamen University. His research interest mainly include computer vision and graphics, intelligent design and their applications.



Guang Yu received Ph.D. degree in computer science and technology from National University of Defense Technology in 2023. He is currently an assistant research professor with National University of Defense Technology, Changsha, China. His main research interests include spatio-temporal prediction, anomaly detection and pattern recognition.



Chengyu Wang received the master's degree in computer science and technology from National University of Defense Technology, China, in 2021. He is currently pursuing the PhD degree with the College of Computer Science and Technology, National University of Defense Technology, China. His main research interests lie in time series analysis and natural language processing.



Fang Liu is a full professor in the School of Design, Hunan University. She received the BS and PhD degrees in computer science from National University of Defense Technology, Changsha, China in 1999 and 2005, respectively. Her main research interests include edge computing, data storage and management, and intelligent design.



Zhiping Cai received the B.Eng., M.A.Sc., and PhD degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is a full professor in the College of Computer, NUDT. His current research interests include artificial intelligence, network security and big data. He is a senior member of the CCF and a member of the IEEE.