Momentor: Advancing Video Large Language Model with Fine-Grained Temporal Reasoning

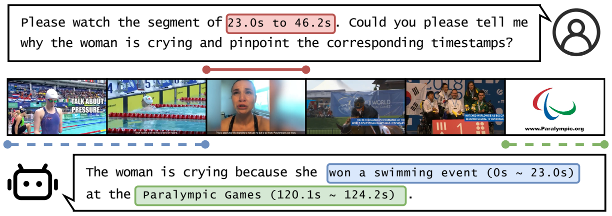
Long Qian [1] Juncheng Li [1] Yu Wu [2] Yaobo Ye [1] Hao Fei [3] Tat-Seng Chua [3] Yueting Zhuang [1] Siliang Tang [1]

Abstract

Large Language Models (LLMs) demonstrate remarkable proficiency in comprehending and handling text-based tasks. Many efforts are being made to transfer these attributes to video modality, which are termed Video-LLMs. However, existing Video-LLMs can only capture the coarse-grained semantics and are unable to effectively handle tasks related to comprehension or localization of specific video segments. In light of these challenges, we propose Momentor, a Video-LLM capable of accomplishing fine-grained temporal understanding tasks. To support the training of Momentor, we design an automatic data generation engine to construct Moment-10M, a large-scale video instruction dataset with segment-level instruction data. We train Momentor on Moment-10M, enabling it to perform segment-level reasoning and localization. Zero-shot evaluations on several tasks demonstrate that Momentor excels in fine-grained temporally grounded comprehension and localization. Our project is available at https://github.com/DCDmllm/Momentor.

1. Introduction

Inspired by the success of ChatGPT (OpenAI, 2022), numerous studies across various fields are attempting to integrate Large Language Models (LLMs) with their domain-specific tasks, seeking to bring innovation to these fields. For example, Video Large Language Models (Video-LLMs) such as VideoChat (Li et al., 2023d) and Video-ChatGPT (Maaz et al., 2023) adapt LLM to video modality, striving to merge the understanding, reasoning and interactive skills of LLM with video perception. They typically sample multiple frames from the video, use an image encoder to encode these frames separately, and employ a projection layer (e.g. a linear layer or Q-Former (Li et al., 2023a)) to adapt the visual features to the feature space of an open-source LLM ((Touvron et al., 2023), (Chiang et al., 2023)). By training on video-level captioning and QA tasks, they establish coarse-grained multimodal feature alignment and acquire the capability of instruction following.



Please watch the segment of 23.0s to 46.2s. Could you please tell me why the woman is crying and pinpoint the corresponding timestamps?

The woman is crying because she won a swimming event (0s ~ 23.0s) at the Paralympic Games (120.1s ~ 124.2s).

Figure 1. Momentor can perform comprehensive reasoning across multiple segments in a video.

Despite being effective, existing Video-LLMs exhibit two limitations: **(1) Lack of effective temporal representation.** Existing models encode each sampled frame independently and perform feature projection without retaining precise temporal information in visual features. They lack an effective temporal representation for encoding time positions at inputs and expressing temporal positions accurately at outputs. While directly expressing timestamps in text format seems to be a feasible approach, such a method suffers inherently from precision variability and tokenization complexity of decimals in LLM. **(2) Lack of segment-level modeling.** Existing models mainly focus on capturing of global visual semantics, while neglecting the modeling of segment-level semantics and relationships. They are typically trained on trimmed videos (usually around a few seconds) for video-level semantic alignment (video captioning) and instruction-following (video QA). However, common untrimmed videos generally last for several minutes and consist of multiple segments with various contents. Consequently, existing Video-LLMs are unable to provide appropriate responses based on certain segments specified by the user, or locate the segment containing specific content precisely.

To address these challenges, we propose Momentor, a Video-LLM with fine-grained temporal awareness and segment-level reasoning capability. To enhance temporal modeling, we introduce innovations in both model architec-

[1] Zhejiang University [2] Wuhan University [3] National University of Singapore. Correspondence to: Juncheng Li <junchengli@zju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Dataset	Total Dur.	Avg Dur.	#Videos	#Instructions	#Segments	#Instances Tracks	#Actions	No Human Annotation	Segment-Level Comprehension	Temporal Localization	Instance Reference	Task Taxonomy
VideoChat (Li et al., 2023d)	41h	18s	8.2k	11.2k	✗	✗	✗	✓	✗	✗	✗	✗
Valley (Luo et al., 2023)	608h	40s	54.7k	73.1k	✗	✗	✗	✓	✗	✗	✗	✓
Video-ChatGPT (Maaz et al., 2023)	432h	117s	13.3k	100k	✗	✗	✗	✗	✗	✗	✗	✗
Moment-10M	**7260h**	**403s**	**64.9k**	**10.4M**	**1.46M**	**451.5k**	**1.51M**	✓	✓	✓	✓	✓

Table 1. Comparison between Moment-10M and existing video instruction datasets

ture and training methodology. For model architecture, we present **Temporal Perception Module**, which is designed to flexibly represent accurate temporal positions within videos and inject temporal information into frame features. Temporal Perception Module extends the LLM's vocabulary with a series of temporal tokens designed for temporal positioning and encoding, allowing LLM to precisely perceive fine-grained temporal information and flexibly output accurate timestamps. To avoid the quantization error in representing time with discrete tokens, we incorporate a continuous interpolation mechanism and construct a continuous temporal feature space on top of these temporal tokens. Further, we design a neighboring token propagation mechanism, which propagates the parameter updates of each temporal token to its neighboring tokens to enhance the quality and continuity of the temporal representations. For training, we propose a **Grounded Event-Sequence Modeling** stage, which trains Momentor to consecutively ground each event in the untrimmed video and caption the corresponding segment with aligned timestamps. Such a temporally grounded event-sequence decoding training bridges the gap between coarse-grained video-level understanding and fine-grained segment-level grounding. It enables Momentor to learn the temporal token space and understand untrimmed videos with complex event sequences.

With fine-grained temporal modeling, we expect that Momentor can learn to perform various segment-level reasoning tasks via instruction tuning. However, existing video instruction datasets do not include segment-level instruction data. Therefore, we propose Moment-10M, a large-scale video instruction fine-tuning dataset with extensive segment-level annotations (*e.g.*, actions, tracks). To construct Moment-10M, We design an innovative and automatic data generation engine. Specifically, given a video, we first track all the instances in the video. Then, we design an event boundary detection algorithm to temporally segment the video into coherent events based on video content and instance behaviours. After that, we develop a structured information extraction framework to derive instance, attribute, and event information from the video. We apply a LLM (Chiang et al., 2023) to synthesize these information and generate instruction data. To facilitate comprehensive segment-level reasoning, we design not only **single-segment tasks** that involve only a single segment, but also **cross-segment tasks**, which require reasoning over multiple segments to provide correct responses. Employing the data generation engine, we generated 10 million instructions to form Moment-10M. As shown in Table 1, Moment-10M comprises 1.5 million segments and 451.5 thousand instance tracks while featuring a larger number of videos as well as significantly longer video durations.

We conduct extensive experiments with our proposed Momentor. The results indicate that our Momentor outperforms previous Video-LLMs in multiple tasks involving precise temporal position, such as temporal grounding, dense captioning, action segmentation, and highlight moment retrieval. Momentor demonstrates advanced proficiency in temporal perception. It can provide appropriate responses based on user-indicated segments as well as quickly locate target segments that meet user requirements.

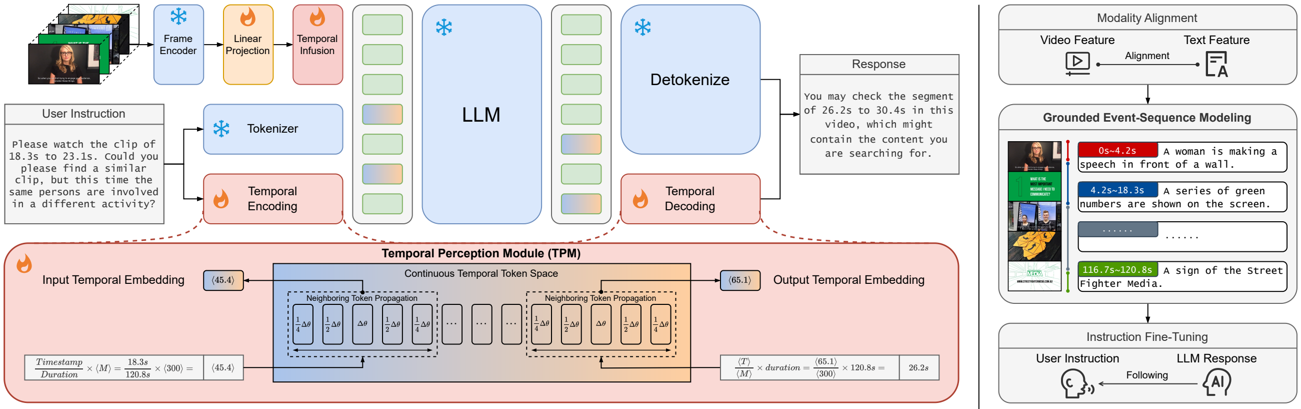
2. Related Work

2.1. Vision and Language Understanding

With the rise of deep learning methods in the fields of computer vision and natural language processing, many efforts have been made to explore more complex multimodal understanding of vision and language. For example, tasks such as image and video-based QA, captioning and retrieval have been extensively discussed and explored by many existing studies (Antol et al., 2015; Vinyals et al., 2015; Faghri et al., 2017; Pan et al., 2023; Tapaswi et al., 2016; Venugopalan et al., 2015; Dong et al., 2021). Inspired by the success of the pre-training paradigm in natural language processing and computer vision, many works (Radford et al., 2021; Li et al., 2023a; 2022a; Sun et al., 2019) propose multimodal pre-trained models with excellent generalization by pre-training on a large amount of image-text or video-text pairs.

2.2. Temporally Grounded Video Understanding

Fine-grained video understanding tasks usually demand the model to view a video as a series of interconnected events and comprehend or locate them in a temporally grounded manner. For instance, *action segmentation* (Singh et al., 2016; Du et al., 2022; Behrmann et al., 2022) requires the model to temporally split the video and output the action label for each segment; *temporal grounding* (Gao et al., 2017; Zhang et al., 2020b;a; Li et al., 2022b; 2023c) demands the model to identify the start and end timestamps of the video segment corresponding to a given natural language query; *highlight moment retrieval* (Lei et al., 2021; Lin et al., 2023) requires the model to find out the central event in a video from a natural language description and pinpoint all related segments; *dense video captioning* (Alwassel et al., 2021; Yang et al., 2023a) requires the model to list out all


Frame Encoder
Linear Projection
Temporal Infusion
LLM
Detokenize
Tokenizer
Temporal Encoding
Temporal Decoding
User Instruction
Response
Temporal Perception Module (TPM)
Input Temporal Embedding
⟨45.4⟩
Continuous Temporal Token Space
Neighboring Token Propagation
$\frac{1}{4}\Delta\theta$ $\frac{1}{2}\Delta\theta$ $\Delta\theta$ $\frac{1}{2}\Delta\theta$ $\frac{1}{4}\Delta\theta$
⟨65.1⟩
Output Temporal Embedding
$\frac{Timestamp}{Duration} \times \langle M \rangle = \frac{18.3s}{120.8s} \times \langle 300 \rangle =$ ⟨45.4⟩
$\frac{\langle T \rangle}{\langle M \rangle} \times duration = \frac{\langle 65.1 \rangle}{\langle 300 \rangle} \times 120.8s =$ 26.2s
Modality Alignment
Video Feature
Text Feature
Alignment
Grounded Event-Sequence Modeling
Instruction Fine-Tuning
User Instruction
LLM Response
Following

Please watch the clip of 18.3s to 23.1s. Could you please find a similar clip, but this time the same persons are involved in a different activity?

You may check the segment of 26.2s to 30.4s in this video, which might contain the content you are searching for.

0s~4.2s A woman is making a speech in front of a wall.

4.2s~18.3s A series of green numbers are shown on the screen.

....

116.7s~120.8s A sign of the Street Fighter Media.

(a) Architecture of Momentor

(b) Training of Momentor

Figure 2. The **(a)** overall architecture and **(b)** training of Momentor.

events contained in a video along with their start and end timestamps. Previous methods typically train a task-specific model for each task, whereas we aim to design a unified Video-LLM that can solve these tasks in a zero-shot manner.

2.3. Multimodal Large Language Models

Many efforts have been made to transfer the task-handling capability of Large Language Models (LLMs) to the vision modality, enabling them to complete various tasks based on image content in accordance with user instructions (Liu et al., 2024; Li et al., 2023b; Pan et al., 2024; Zhu et al., 2023; Ge et al., 2024; Gao et al., 2024; Zhang et al., 2024). Several models (Li et al., 2023d; Maaz et al., 2023; Zhang et al., 2023; Luo et al., 2023; Huang et al., 2023; Ren et al., 2023) also incorporate temporal information aggregation module with LLM, in order that they can understand video content. Despite being effective in captioning or QA on short videos, the lack of fine-grained temporal modeling in these models prevents them from understanding or locating specific segments in long videos. In contrast, Momentor employs a Temporal Perception Module that integrates a continuous temporal token space for precise temporal positioning and modeling.

3. Momentor

In this section, we present Momentor, a Video-LLM designed for fine-grained comprehension and localization in videos, as shown in Figure 2. To empower Momentor with fine-grained temporal awareness, we propose the Temporal Perception Module (TPM) (Section 3.2), which facilitates precise temporal positioning and fine-grained temporal information injection. To better train TPM, we introduce Grounded Event-Sequence Modeling (Section 3.3) as an additional pre-training stage, which enables Momentor to comprehend videos in a temporally grounded manner and prepares it for segment-level instruction following tasks.

3.1. Overall Pipeline

Momentor is composed of a frame encoder (Dosovitskiy et al., 2020), a linear projection layer, a Temporal Perception Module (TPM), and a Large Language Model (LLM) (Touvron et al., 2023). After receiving one input video, Momentor will first uniformly sample multiple frames from the video and encode each frame independently to get frame features. These frame features will be projected into the LLM's feature space by the linear projection layer. The projected features are then processed in the TPM for temporal information injection, which are then concatenated with tokenized user instructions to be the input of LLM. During training, the frame encoder and LLM are kept frozen, while only the linear projection layer and TPM are updated.

3.2. Temporal Perception Module (TPM)

We propose the Temporal Perception Module to equip Momentor with fine-grained temporal awareness and provide an interface to express precise temporal positions. Specifically, Temporal Perception Module incorporates a continuous temporal token space and employs neighboring token propagation to facilitate the continuity in token space.

Continuous Temporal Token Space. We employ a continuous feature space for precise temporal positioning. Specifically, we uniformly divide the video into $N-1$ segments, and then define N learnable anchor point features to represent the $N-2$ split points and 2 endpoints, encompassing the relative temporal positions within the video. Then we apply interpolation to define the feature of each temporal point in the timeline, thereby constructing a continuous temporal feature space. With the temporal feature space, we can precisely represent arbitrary temporal positions, enabling Momentor to input or output exact time positions. To unify the training process, we incorporate these anchor point features as specialized temporal tokens into the LLM's vocabulary, denoted as $\langle 1 \rangle$, $\langle 2 \rangle$, ..., $\langle \text{N} \rangle$, and the outlined feature space is referred as the continuous temporal token space. Therefore, we can train Momentor in an auto-aggressive manner using a unified cross-entropy loss. Studies like

Vid2Seq (Yang et al., 2023a) also add specialized tokens to the text decoder's vocabulary to express temporal positions. However, they directly use the discrete tokens for temporal positioning in continuous timelines, which introduces quantization error and prevents them from precise temporal localization. In contrast, our approach solves this problem by constructing a continuous temporal token space on top of these temporal tokens, thereby avoiding quantization error and enabling precise temporal position representation.

Neighboring Token Propagation. Unlike language tokens, temporal tokens have a clear sequential relationship. We expect continuity among these temporal tokens, meaning that the embeddings of adjacent tokens should be more similar to each other than those of tokens that are farther apart. However, existing models that use discretized tokens to represent temporal positions have not incorporated any techniques to highlight such continuity. To tackle this issue, we employ a neighboring token propagation mechanism, which enhances continuity by propagating the parameter updates of one temporal token to its adjacent tokens. For any temporal token ⟨k⟩ involved in the training process, we have:

$$\tilde{t_k} = t_k + t_{adj} - StopGrad(t_{adj}), \tag{1}$$

$$t_{adj} = \sum_{i=1}^{N} \frac{1}{2^{|i-k|}} \cdot t_i, \tag{2}$$

where $\tilde{t_k}$ is the embedding of temporal token ⟨k⟩ after neighboring token propagation, t_i is the original embedding for temporal token ⟨i⟩, $StopGrad$ is the operation to detach a variable's gradient, and t_{adj} is a variable that gathers gradients from all adjacent temporal tokens through a weighted sum. By adding t_{adj} to t_k and subsequently subtracting the gradient-detached t_{adj}, we incorporate adjacent temporal tokens into the computation graph, allowing them to receive parameter updates along with t_k, while keeping the value of t_k unchanged for precise temporal representation. The weight of each adjacent temporal token in t_{adj} decreases exponentially as their distance to t_k increases. Consequently, temporal tokens closer to t_k receive more similar parameter updates compared to those farther away, and adjacent temporal tokens tend to have more similar embeddings, thereby strengthening the continuity among temporal tokens. We use $\tilde{t_k}$ instead of t_k in training.

Temporal Information Injection. Since each sampled frame is encoded and projected separately, their features do not contain the corresponding temporal position information. After constructing a continuous temporal token space and applying the neighboring token propagation, now we can actually obtain temporal embeddings corresponding to any timestamp, which contain precise temporal position information and possess the valuable property of temporal continuity. Therefore, we obtain the temporal embeddings at the positions of the sampled frames and directly add them to the projected frame features, as they share the same dimensionality, serving as a form of temporal position encoding to inject fine-grained temporal information.

3.3. Grounded Event-Sequence Modeling

Common untrimmed videos often span several minutes and contain numerous events with diversified content. To facilitate multi-event comprehension, we introduce Grounded Event-Sequence Modeling, an additional pre-training stage focusing on event-sequence decoding, which enables the Temporal Perception Module to align its temporal token space with video timelines and comprehend events in a temporally-grounded manner. We conduct Grounded Event-Sequence Modeling after modality alignment, building temporal awareness upon the aligned multimodal semantics.

Modality Alignment. To align the visual and textual modalities, we train the linear projection layer with a broadly collected dataset of image-text and video-text pairs:

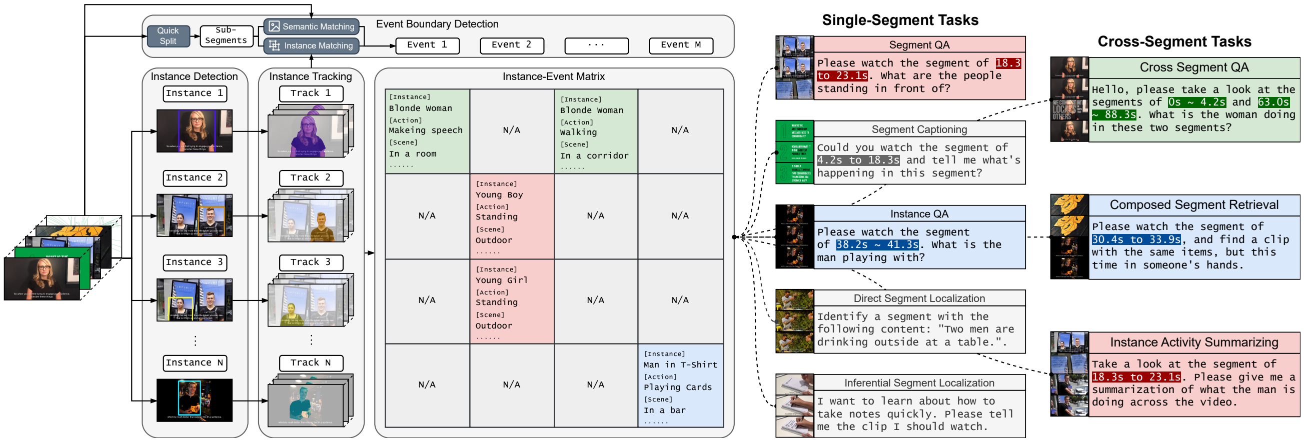
$$\mathcal{L}_{align} = -\frac{1}{l} \sum_{i=0}^{l} \log p(T_C^{i+1} | T_v, T_C^{1:i}), \tag{3}$$

where T_C^i is the i th token of the image or video caption T_C, and T_v is the frame features.

Event-Sequence Decoding. After the stage of modality alignment, the model only learns the coarse-grained correspondence between visual and textual data. It still lacks fine-grained temporal awareness, so fine-tuning it directly on instruction data with precise timestamps can lead to slow convergence and ineffective event-sequence modeling. Therefore, we apply event-sequence decoding as an intermediary task that bridges the gap between low-level semantic alignment and high-level conceptual interaction. To be precise, given an untrimmed video as input, we require the model to output the event-sequence within it. We represent the k th event as $E_k = [t_{start}^k, t_{end}^k, w_1^k, ..., w_{l_k}^k]$, where t_{start}^k, t_{end}^k are the continuous temporal embeddings at the start and end of the k th event, and $[w_1^k, ..., w_{l_k}^k]$ is a general caption composed of l_k tokens for this event. The timestamps and general captions of each event in the event-sequence can be conveniently obtained during our instruction generation process without additional calculation (Section 4.2). We concatenate all the events to formulate the event-sequence $T_E = \{E_i\}_{i=1}^{N_E}$, where N_E is the number of events in the untrimmed video. We apply a language modeling loss for event-sequence decoding:

$$\mathcal{L}_{decode} = -\frac{1}{l} \sum_{i=0}^{l} \log p(T_E^{i+1} | T_v, T_E^{1:i}), \tag{4}$$

where T_E^i is the i th token of the event-sequence T_E, and T_v is the frame features. With Grounded Event-Sequence Modeling, we establish a preliminary association between the temporal token space and the relative temporal positions



[Instance] Blonde Woman [Action] Makeing speech [Scene] In a room	N/A	[Instance] Blonde Woman [Action] Walking [Scene] In a corridor	N/A
N/A	[Instance] Young Boy [Action] Standing [Scene] Outdoor	N/A	N/A
N/A	[Instance] Young Girl [Action] Standing [Scene] Outdoor	N/A	N/A
N/A	N/A	N/A	[Instance] Man in T-Shirt [Action] Playing Cards [Scene] In a bar ......

Segment QA: Please watch the segment of 18.3 to 23.1s. What are the people standing in front of?

Segment Captioning: Could you watch the segment of 4.2s to 18.3s and tell me what's happening in this segment?

Instance QA: Please watch the segment of 38.2s ~ 41.3s. What is the man playing with?

Direct Segment Localization: Identify a segment with the following content: "Two men are drinking outside at a table.".

Inferential Segment Localization: I want to learn about how to take notes quickly. Please tell me the clip I should watch.

Cross Segment QA: Hello, please take a look at the segments of 0s ~ 4.2s and 63.0s ~ 88.3s. What is the woman doing in these two segments?

Composed Segment Retrieval: Please watch the segment of 30.4s to 33.9s, and find a clip with the same items, but this time in someone's hands.

Instance Activity Summarizing: Take a look at the segment of 18.3s to 23.1s. Please give me a summarization of what the man is doing across the video.

Figure 3. The pipeline of our automatic instruction data generation engine, which can automatically extract structured information from videos and generate diversified instruction data.

within videos, laying the groundwork for segment-level instruction following.

4. Moment-10M

Teaching a Video-LLM to locate specific segments in untrimmed videos and perform complex reasoning on these segments requires substantial training data with fine-grained annotation. However, existing video instruction datasets don't contain instructions with precise timestamps, and their task formats are often limited to captioning, summarizing and basic QA, which overlook the logical associations between events and instances. In light of this, we propose `Moment-10M`, a large-scale video instruction fine-tuning dataset with segment-level reasoning tasks. To construct `Moment-10M`, we design a data generation engine that can automatically extract instance and event information along with their relationships from the videos, and then generate corresponding instruction data based on these information, as shown in Figure 3. We meticulously design various types of instruction-following tasks, aiming to enhance `Momentor` in comprehensive segment-level reasoning.

4.1. Structured Information Extraction

The relationships between instances and events in an untrimmed video can be extremely complex. A particular instance might appear in different events that are far apart, and an event might contain several instances that seem unrelated. To fully explore the associations between instances and events within a video, we propose an Event Boundary Detection algorithm that can accurately detect the event boundaries in the video based on the instance information and video content. We then construct an Instance-Event Matrix, to extract and organize the visual information in a structured way, where the spatio-temporal correspondences from a video can be effectively captured.

Event Boundary Detection. For an arbitrary video to be processed, we first uniformly sample multiple frames from the video. We employ Grounding DINO (Liu et al., 2023) to extract instance information from these sampled frames, and then compare and merge the instances across the sampled frames to obtain the spatio-temporal trajectories of instances in the video, termed as instance tracks. The instance tracks show the dynamics of each instance over time, which also reflect the event transitions in the video. Based on video content and instance dynamics, we design a comprehensive event boundary detection method. We first use PySceneDetect (Castellano, 2018) to calculate frame-by-frame differences in the video, resulting in an array of frame difference scores. Then, we apply a Gaussian filter to reduce noise and smooth these scores. We select local maxima that are higher than a certain threshold as split points, to divide the video into several sub-segments. Since such segmentation only considers changes in RGB values and doesn't account for semantic transitions, we adopt a semantic-based merging algorithm to merge adjacent sub-segments that experience abrupt visual changes but still belong to the same event. To be precise, for the two adjacent sub-segments, we extract the last frame from the previous sub-segment and the first frame from the next sub-segment and calculate their consistency value as:

$$\begin{aligned} Consistency &= \cos(F^{'}, F^{''}) \\ &+ \frac{1}{|U_I|}\sum_{i=1}^{|U_I|}\cos(F^{'}_{I_i}, F^{''}_{I_i}) \cdot (1 - Dist(I^{'}_i, I^{''}_i)), \end{aligned} \tag{5}$$

where $F^{'}$ and $F^{''}$ are visual features of the last frame in the previous sub-segment and the first frame in the next sub-segment, and U_I is the union of instances shown in these two frames. $F^{'}_{I_i}$ and $F^{''}_{I_i}$ are ROI aligned (He et al., 2017) features of the i th instance, and $Dist(I^{'}_i, I^{''}_i)$ is the normalized distance between the positions of the i th instance in these two frames with a value in $[0, 1]$. We set this distance to be 1 if the i th instance appears in only one of these two frames. All visual features involved have been obtained during object detection, thus not incurring additional computational costs. We merge two adjacent sub-segments if

Model	Action Segmentation								Dense Video Captioning		
	Breakfast				50Salads				ActivityNet-Captions		
	MoF	F1@{10, 25, 50}			MoF	F1@{10, 25, 50}			SODA_c	CIDEr	METEOR
Video-ChatGPT (7B) (Maaz et al., 2023)	5.1	7.8	2.4	0.5	9.6	7.1	3.1	1.1	0.4	2.1	0.7
VideoChat (7B) (Li et al., 2023d)	7.9	8.8	5.3	2.8	13.3	10.6	3.5	1.1	0.7	3.3	1.2
Video-LLaMA (7B) (Zhang et al., 2023)	11.6	15.2	8.8	4.2	14.3	12.9	4.0	1.2	0.9	4.6	2.4
Valley (7B) (Luo et al., 2023)	4.1	7.4	4.5	2.4	13.2	11.3	3.5	1.8	0.3	1.8	0.8
Momentor (7B)	**24.4**	**41.2**	**33.6**	**21.8**	**17.8**	**22.8**	**15.9**	**13.0**	**2.3**	**14.9**	**4.7**

Table 2. Comparison with existing Video-LLMs on dense video captioning and action segmentation

Model	Temporal Grounding								Highlight Moment Retrieval	
	ActivityNet-Captions				Charades-STA				QVHighlights	
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	mAP	R1@0.5
Video-ChatGPT (7B) (Maaz et al., 2023)	19.5	10.6	4.8	14.2	27.2	6.2	1.9	19.7	3.8	8.7
VideoChat (7B) (Li et al., 2023d)	23.5	12.6	6.0	17.4	32.8	8.6	0.0	25.9	4.1	7.0
Video-LLaMA (7B) (Zhang et al., 2023)	21.9	10.8	4.9	16.5	25.2	10.6	3.4	16.8	2.1	6.6
Valley (7B) (Luo et al., 2023)	30.6	13.7	8.1	21.9	28.4	1.8	0.3	21.4	5.3	8.7
Momentor (7B)	**42.9**	**23.0**	**12.4**	**29.3**	**42.6**	**26.6**	**11.6**	**28.5**	**7.6**	**17.0**

Table 3. Comparison with existing Video-LLMs on temporal grounding and highlight moment retrieval

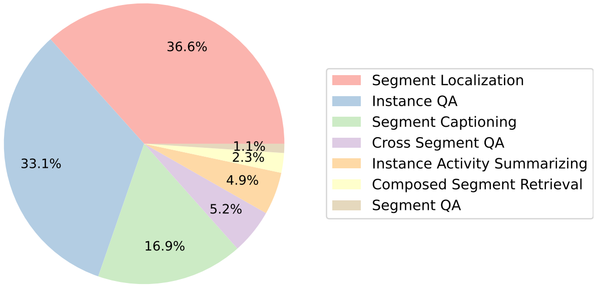


Figure 4. Distribution of different tasks in `Moment-10M`.

their consistency value is higher than a set threshold. Consequently, we obtain a series of segments with semantic consistency, each encompassing a coherent event.

Instance-Event Matrix. Based on the result of instance tracking and event segmentation, we construct an Instance-Event Matrix, where each row represents an instance track (the video itself also counts as a track), and each column represents an event. The instance-event matrix shares certain similarities with video scene graphs (Shang et al., 2017; Yang et al., 2023b) as both involve instance behaviour tracking and structured semantic representation, but the instance-event matrix places greater emphasis on modeling the complex associations between events. We traverse the matrix and utilize several multimodal pre-trained models to extract visual clues such as scenes, instances, actions and attributes from each track. With the structured information organized in instance-event matrix, we can quickly generate instruction data that includes various spatio-temporal associations.

4.2. Instruction Generation

We feed the information in the instance-event matrix into Vicuna (Chiang et al., 2023), an open-source text-based LLM, to generate instruction data. We design various types of instruction-following tasks to comprehensively train and evaluate Video-LLMs. We incorporate 5 tasks focusing on single segment understanding as well as 3 tasks that involve reasoning across multiple segments, as shown in Figure 3. We utilize various prompts to guide Vicuna in generating instruction data for different tasks. Data from all 8 task types are used for instruction fine-tuning, while segment captioning data organized chronologically will be utilized for Grounded Event Sequence Modeling (Section 3.3). Detailed task descriptions and prompts can be found in Appendix C and D. We select a substantial number of videos from YTTemporal-1B (Zellers et al., 2022) to build `Moment-10M`. Figure 4 shows the distribution of each type of instruction data in `Moment-10M`. As shown in Table 1, `Moment-10M` comprises 10 million instruction data over 1.5 million segments and 451.5 thousand instance tracks. On average, each video contains 22.7 segments, which reflects the complexity of the event-sequences in the videos. We fine-tune `Momentor` on `Moment-10M`, enabling it to perform segment-level reasoning and localization.

5. Experiments

5.1. Experiment Setup

To comprehensively evaluate `Momentor` in fine-grained understanding and precise localization, we assess it in a zero-shot setting across four tasks, i.e., *action segmentation*, *dense video captioning*, *temporal grounding*, and *highlight moment retrieval*, using datasets such as Breakfast (Kuehne et al., 2014), 50 Salads (Stein & McKenna, 2013), ActivityNet Captions (Krishna et al., 2017), Charades-STA (Gao et al., 2017), and QVHighlights (Lei et al., 2021). We also perform evaluation on Video QA datasets such as ActivityNet-QA (Yu et al., 2019), MSRVTT-QA, and MSVD-QA (Xu et al., 2017) to evaluate `Momentor` in general question answering. Implementation details of `Momentor` can be found in Appendix B.

Model	Video QA					
	MSVD-QA		MSRVTT-QA		ActivityNet-QA	
	Acc.	Score	Acc.	Score	Acc.	Score
Video-ChatGPT (7B)	64.9	3.3	49.3	2.8	35.2	2.7
VideoChat (7B)	56.3	2.8	45.0	2.5	26.5	2.2
Video-LLaMA (7B)	51.6	2.5	29.6	1.8	12.4	1.1
Valley (7B)	65.4	3.4	51.1	**3.0**	**45.1**	**3.2**
Momentor (7B)	**68.9**	**3.6**	**55.6**	**3.0**	40.8	**3.2**

Table 4. Existing Video-LLMs' performance on Video QA

5.2. Action Segmentation

Given a video, action segmentation requires the model to divide the video into multiple non-overlapping segments and assign an action category label to each segment. Since Momentor's output is free-form text rather than action category labels, we use a sentence transformer (Reimers & Gurevych, 2019) to convert the output from Momentor into features, which are then compared with the features of action category labels to determine their corresponding action categories. We evaluate Momentor on Breakfast and 50 Salads, of which the results can be referenced in Table 2. From the results we can infer: **(1)** Overall, Momentor can effectively segment and recognize actions in input videos. In the setting of zero-shot action segmentation, Momentor achieves the highest accuracy among existing Video-LLMs. **(2)** Despite only being trained on generating free-form texts rather than action labels, Momentor's proficiency in visual information capturing still allows it to effectively generate texts that closely align with action label words, enabling it to perform accurate action classification.

5.3. Dense Video Captioning

Given a video, dense video captioning requires the model to output all events contained in the video along with their start and end timestamps. We test Momentor on ActivityNet Captions, and the results can be found in Table 2, from which we can conclude: **(1)** Compared to existing Video-LLMs, Momentor provides more detailed event descriptions and more accurate event boundaries. **(2)** Thanks to Grounded Event-Sequence Modeling, Momentor can capture the events in a video as completely as possible, while also providing precise start and end timestamps and accurate descriptions of each event. The model's leading performance just validates our viewpoint.

5.4. Temporal Grounding

Given a video and a natural language query, temporal grounding requires the model to identify the start and end timestamps of the segment corresponding to the query in the video. We evaluate Momentor on ActivityNet Captions and Charades-STA, with the results available in Table 3. Based on the experiment results, we can draw the following conclusions: **(1)** Momentor achieves the highest mean IoU (Intersection over Union) among existing Video-LLMs. **(2)** With the neighboring token propagation mechanism in the Temporal Perception Module, Momentor constructs a continuous and precise temporal token space, laying the foundation for accurate event localization. Ablation studies and visualization in **Section** 5.7 also validate this point.

Setting	ActivityNet		Breakfast	QVHighlights
	mIoU	CIDEr	MoF	mAP
Momentor (7B)	29.3	14.6	24.4	7.6
w/o CI	27.6	13.1	22.5	7.1
w/o NTP	25.4	10.3	19.3	6.1
w/o GESM	27.8	9.8	19.5	6.8
w/o Cross-Segment Tasks	29.0	12.1	21.6	6.4

Table 5. Performance of ablation models. CI: Continuous Interpolation, NTP: Neighboring Token Propagation, GESM: Grounded Event-Sequence Modeling

5.5. Highlight Moment Retrieval

Given a video and a description of the highlight activities within the video, highlight moment retrieval requires the model to locate all the highlighted segments corresponding to the description. We evaluate Momentor on QVHighlights, and the results can be referenced in Table 3. From these results we can know: **(1)** Among all existing Video-LLMs, Momentor achieves state-of-the-art performance on highlight moment retrieval. **(2)** Thanks to the multi-event reasoning ability developed on the cross-segment tasks, Momentor can perceive the overall video semantics from a global perspective and effectively comprehend the relationships between different events, which is a key factor in highlight moment retrieval.

5.6. Video QA

We test Momentor on ActivityNet-QA, MSRVTT-QA, and MSVD-QA. As shown in Table 4, Momentor achieves state-of-the-art or comparative performance among Video-LLMs across all tested datasets, demonstrating its capability in coarse-grained video understanding.

5.7. In-Depth Analysis

Ablation Studies. We conduct ablation experiments to assess the effectiveness of each component. The experiments are conducted under the following settings: (1) w/o continuous interpolation: We still use temporal tokens to express temporal positions, but without integrating the continuous interpolation mechanism. (2) w/o neighboring token propagation: We use the continuous temporal token space for temporal positioning, but without applying the neighboring token propagation mechanism when training. (3) w/o grounded event-sequence modeling: After modality alignment, we proceed directly to instruction fine-tuning without grounded event-sequence modeling. (4) w/o cross-segment tasks: We remove all instructions from cross-segment tasks and use only single-segment tasks for fine-tuning. We train Momentor with these settings and evaluate perfor-

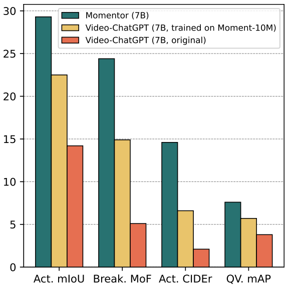


Figure 5. Dataset validation. Act.: ActivityNet. Break.: Breakfast. QV.: QVHighlights.

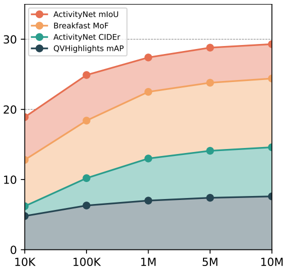
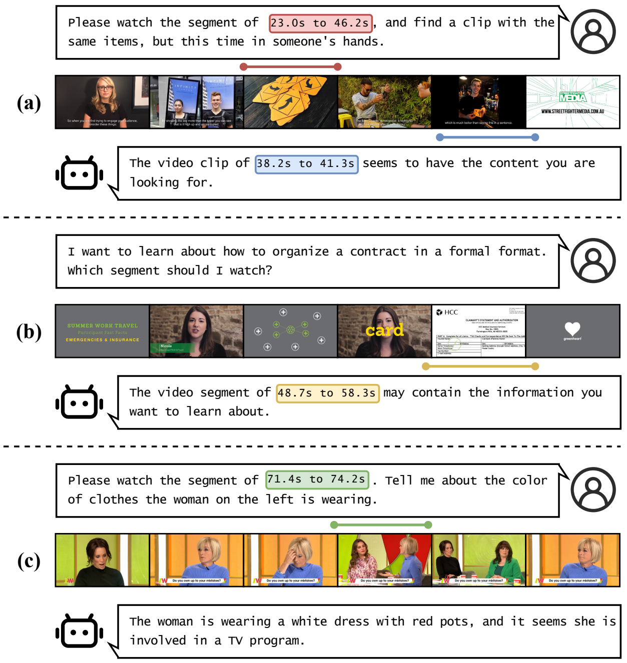


Figure 6. Impact of data scale. Generally, the performance improves as data scale increases.

mances on ActivityNet Captions (temporal grounding and dense video captioning), Breakfast (action segmentation) and QVHighlights (highlight moment retrieval). The results of the ablation experiments can be referenced in Table 5.

Overall, removing any one of these components results in a decrease in the model's overall performance. From Table 5, we can analyze the impact of removing different components on model performance separately. After removing the continuous interpolation mechanism, due to the quantization error, Momentor experiences a minor decline in localization-related metrics across all tasks, while the caption quality-related metrics are not significantly affected. Removing the neighboring token propagation mechanism leads to a performance drop in all metrics. Without neighboring token propagation, the temporal tokens are updated as multiple unrelated tokens rather than as an ordered sequence, which undermines the temporal representation and modeling. Visualizations of the temporal tokens (Section 8) also confirm this observation. Removing grounded event-sequence modeling leads to a significant performance decline in dense prediction tasks like dense video captioning and action segmentation, which indicates that grounded event-sequence modeling plays an important role in sequential semantics comprehension. The removal of cross-segment tasks has minimal impact on the performance of temporal grounding, as it does not involve cross-segment understanding. Performance on other tasks generally decreases, as both dense video captioning and action segmentation involve comprehension of multiple segments, and highlight moment retrieval also requires the model to distinguish between highlight segments and background segments.

Validation of `Moment-10M`. We train Video-ChatGPT (Maaz et al., 2023) on our `Moment-10M` to validate its efficacy in improving fine-grained temporal reasoning. Despite being inefficient in temporal representation, we still use textual timestamps to represent temporal positions since Video-ChatGPT doesn't provide alternative temporal representation methods. As shown in Figure 5, Video-ChatGPT trained on `Moment-10M` shows a great improvement on fine-grained temporal reasoning tasks.



Please watch the segment of 23.0s to 46.2s, and find a clip with the same items, but this time in someone's hands.

The video clip of 38.2s to 41.3s seems to have the content you are looking for.

I want to learn about how to organize a contract in a formal format. Which segment should I watch?

The video segment of 48.7s to 58.3s may contain the information you want to learn about.

Please watch the segment of 71.4s to 74.2s. Tell me about the color of clothes the woman on the left is wearing.

The woman is wearing a white dress with red pots, and it seems she is involved in a TV program.

Figure 7. Analysis on special cases.

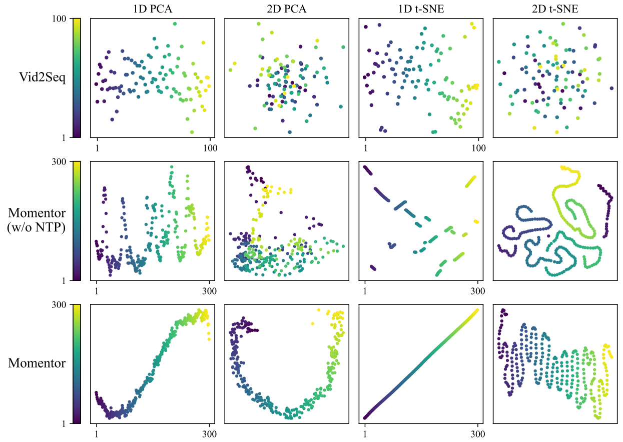


Figure 8. Visualization of temporal tokens in `Momentor` and time tokens in *Vid2Seq*. NTP: neighboring token propagation.

Impact of Data Scale. We train `Momentor` with different amounts of instruction data, while the proportions of different tasks are kept the same. The results can be referenced in Figure 6. Generally, the model's performance improves as the amount of training data increases, but slows down once the training data reaches a million-level scale.

Case Studies. We provide qualitative examples to demonstrate the fine-grained reasoning capability of `Momentor`. As shown in Figure 7(a), `Momentor` can integrate visual and textual input for comprehensive localization of target segment. Moreover, even when only a vague scene or requirement description is provided, `Momentor` can still understand the user's intent and pinpoint the segment containing relevant information, as exemplified in Figure 7(b). Additionally, although we don't incorporate spatial modeling, Momentor can still understand which instance the user is referring to and provide appropriate responses, as illustrated in Figure 7(c).

Visualization of Temporal Tokens. Since temporal tokens are used to represent uniformly distributed temporal positions, we expect them to exhibit continuity in their embeddings. We use PCA (Abdi & Williams, 2010) and t-SNE (Van der Maaten & Hinton, 2008) to reduce the dimensionality of temporal tokens of Momentor and time tokens of *Vid2Seq* (Yang et al., 2023a) into 1D and 2D for visualization. To validate the effectiveness of neighboring token propagation, we also visualize the temporal tokens trained without neighboring token propagation. For a fair comparison, we set the random state of t-SNE fixed to be 0. For the 1D reductions, we use the token indices as the x-axis and the reduced values as the y-axis; for the 2D reductions, we directly use the reduced values as coordinates. We employ a gradient color scheme, where the color of the data points will change progressively with the token index, as shown in Figure 8. It is evident that with neighboring token propagation, the embeddings of temporal tokens in Momentor are significantly more continuous. In contrast, embeddings of temporal tokens without neighboring token propagation and time tokens of *Vid2Seq* exhibit much less continuity, as their correlation can only be learned indirectly and inefficiently.

6. Conclusion

We propose Momentor, a Video-LLM with segment-level comprehension and localization capabilities, and Moment-10M, a video instruction dataset comprising 10 million diversified instructions with segment-level annotation. We design a Temporal Perception Module to provide fine-grained temporal representation, and apply Grounded Event-Sequence Modeling to promote multi-event modeling in untrimmed videos. We train Momentor on Moment-10M, enabling it to perform comprehensive segment-level reasoning. Extensive experiments on various tasks demonstrate Momentor's proficiency in fine-grained video understanding.

Impact Statement

Our dataset, sourced from internet videos, is meticulously curated with stringent privacy safeguards. We acknowledge the potential presence of personal information and have instituted comprehensive measures to ensure its protection. Our model is conscientiously developed to be free from social harm and ethical breaches, embodying our commitment to responsible and beneficial technological advancement.

Acknowledgements

This work was supported by the Key Research and Development Projects in Zhejiang Province (No. 2024C01106), the NSFC (No. 62372341), the National Key Research and Development Project of China (2018AAA0101900), Ant Group, and Research funding from FinVolution Group.

References

Abdi, H. and Williams, L. J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.*, 2(4):433–459, 2010.

Alwassel, H., Giancola, S., and Ghanem, B. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3173–3183, 2021.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Behrmann, N., Golestaneh, S. A., Kolter, Z., Gall, J., and Noroozi, M. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *European Conference on Computer Vision*, pp. 52–68. Springer, 2022.

Castellano, B. Pyscenedetect: Intelligent scene cut detection and video splitting tool. https://pyscenedetect.readthedocs.io/en/latest/, 2018.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Dong, J., Li, X., Xu, C., Yang, X., Yang, G., Wang, X., and Wang, M. Dual encoding for video retrieval by text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4065–4080, 2021.

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

Du, Z., Wang, X., Zhou, G., and Wang, Q. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3323–3332, 2022.

Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

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

Gao, M., Chen, S., Pang, L., Yao, Y., Dang, J., Zhang, W., Li, J., Tang, S., Zhuang, Y., and Chua, T.-S. Fact: Teaching mllms with faithful, concise and transferable rationales. *arXiv preprint arXiv:2404.11129*, 2024.

Ge, Z., Huang, H., Zhou, M., Li, J., Wang, G., Tang, S., and Zhuang, Y. Worldgpt: Empowering llm as multimodal world model. *arXiv preprint arXiv:2404.18202*, 2024.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Huang, B., Wang, X., Chen, H., Song, Z., and Zhu, W. Vtimellm: Empower llm to grasp video moments. *arXiv preprint arXiv:2311.18445*, 2(3):9, 2023.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Carlos Niebles, J. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Kuehne, H., Arslan, A., and Serre, T. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 780–787, 2014.

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

Li, J., He, X., Wei, L., Qian, L., Zhu, L., Xie, L., Zhuang, Y., Tian, Q., and Tang, S. Fine-grained semantically aligned vision-language pre-training. *Advances in neural information processing systems*, 35:7290–7303, 2022a.

Li, J., Xie, J., Qian, L., Zhu, L., Tang, S., Wu, F., Yang, Y., Zhuang, Y., and Wang, X. E. Compositional temporal grounding with structured variational cross-graph correspondence learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3032–3041, 2022b.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023a.

Li, J., Pan, K., Ge, Z., Gao, M., Ji, W., Zhang, W., Chua, T.-S., Tang, S., Zhang, H., and Zhuang, Y. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*, 2023b.

Li, J., Tang, S., Zhu, L., Zhang, W., Yang, Y., Chua, T.-S., Wu, F., and Zhuang, Y. Variational cross-graph reasoning and adaptive structured semantics learning for compositional temporal grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12601–12617, 2023c.

Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., and Qiao, Y. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023d.

Lin, K. Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A. J., Yan, R., and Shou, M. Z. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2794–2804, 2023.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Luo, R., Zhao, Z., Yang, M., Dong, J., Qiu, M., Lu, P., Wang, T., and Wei, Z. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023.

Maaz, M., Rasheed, H., Khan, S., and Khan, F. S. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

OpenAI. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`, 2022. Accessed on: November 30, 2022.

Pan, K., Li, J., Song, H., Fei, H., Ji, W., Zhang, S., Lin, J., Liu, X., and Tang, S. Controlretriever: Harnessing the power of instructions for controllable retrieval. *arXiv preprint arXiv:2308.10025*, 2023.

Pan, K., Tang, S., Li, J., Fan, Z., Chow, W., Yan, S., Chua, T.-S., Zhuang, Y., and Zhang, H. Auto-encoding morph-tokens for multimodal llm, 2024.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Ren, S., Yao, L., Li, S., Sun, X., and Hou, L. Timechat: A time-sensitive multimodal large language model for long video understanding. *arXiv preprint arXiv:2312.02051*, 2023.

Shang, X., Ren, T., Guo, J., Zhang, H., and Chua, T.-S. Video visual relation detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1300–1308, 2017.

Singh, B., Marks, T. K., Jones, M., Tuzel, O., and Shao, M. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1961–1970, 2016.

Stein, S. and McKenna, S. J. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 729–738, 2013.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7464–7473, 2019.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pp. 4534–4542, 2015.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., and Zhuang, Y. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

Yang, A., Nagrani, A., Seo, P. H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., and Schmid, C. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10714–10726, 2023a.

Yang, J., Peng, W., Li, X., Guo, Z., Chen, L., Li, B., Ma, Z., Zhou, K., Zhang, W., Loy, C. C., et al. Panoptic video scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18675–18685, 2023b.

Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., and Tao, D. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9127–9134, 2019.

Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., and Choi, Y. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16375–16387, 2022.

Zhang, H., Sun, A., Jing, W., and Zhou, J. T. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020a.

Zhang, H., Li, X., and Bing, L. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Zhang, S., Peng, H., Fu, J., and Luo, J. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12870–12877, 2020b.

Zhang, W., Lin, T., Liu, J., Shu, F., Li, H., Zhang, L., Wanggui, H., Zhou, H., Lv, Z., Jiang, H., et al. Hyperllava: Dynamic visual and language expert tuning for multimodal large language models. *arXiv preprint arXiv:2403.13447*, 2024.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A. Overview

In this appendix we present:

- Implementation details of `Momentor` (Section B).
- Descriptions of the tasks in `Moment-10M` (Section C).
- Prompts used for instruction generation (Section D).

B. Implementation

We utilize the CLIP (Radford et al., 2021) ViT-L/14 as the frame encoder and LLaMA (Touvron et al., 2023) (7B) as the LLM. We initialize the linear projection layer with parameters from Video-ChatGPT's (Maaz et al., 2023) equivalent component. We incorporate $N = 300$ temporal tokens for temporal positioning. For each video, we uniformly sample $M = 300$ frames for fine-grained reasoning. We freeze the frame encoder and LLM during training, while only the linear projection layer and TPM are updated. We train `Momentor` on 8 A100 GPUs for around 60 hours. Our project is available at https://github.com/DCDmllm/Momentor.

C. Task Formats

Single-Segment Tasks:

- *Segment Captioning*: Given a segment, the Video-LLM is required to output a caption to conclude its content.
- *Segment QA*: Given a segment, the Video-LLM is required to answer questions about that segment.
- *Instance QA*: Given an instance at a certain moment, the Video-LLM is required to answer questions about that instance's behavior at that moment.
- *Direct Segment Localization*: Given a query text, the Video-LLM is required to locate the described segment in the video and output its timestamp.
- *Inferential Segment Localization*: Given a hypothetical scenario, the Video-LLM is required to find the scene in the video that likely correspond to that scenario and output its timestamp.

Cross-Segment Tasks:

- *Composed Segment Retrieval*: Given a source segment and the differences between the target and source segments, the Video-LLM is required to identify the target segment based on the source segment and these differences, and output its timestamp.
- *Instance Activity Summarizing*: Given an instance, the Video-LLM is required to summarize the activities of this instance throughout the entire video.
- *Cross-Segment QA*: Given multiple segments, the Video-LLM is required to combine information from all these segments to answer questions.

D. Prompts

Below are the prompts used for generation of different kinds of instruction data. Due to page length constraints, we have omitted some in-context examples in certain tasks.

Segment Captioning

Hello, I want you to act as a comprehensive video captioner. You will receive a list of frame-by-frame descriptions extracted from one video. Since some of these descriptions might be noisy, you should comprehend the major content of the video, assess the correctness of different pieces of information, and filter out erroneous, repetitive, noisy, or irrelevant details. After receiving and analyzing all the descriptions, please generate a comprehensive caption that effectively summarizes the events taking place in the video. Below are the information extracted from the video:

{descriptions}

There are some requirements that you should follow:
1. Your comprehensive video caption should be comprehensive, concise, informative, and LESS than 20 words.
2. You should output ONLY THE COMPREHENSIVE VIDEO CAPTION, and NO OTHER CONTENTS should be printed.
3. Your comprehensive video caption MUST NOT mention the concept of ”frame” or ”video”.
Now please print out your comprehensive video caption.

The comprehensive video caption:

Segment QA

Generate a concise dialogue with factual questions and their answers based on the following video segment caption:

{segment_caption}

The answers should be directly inferred from the provided segment caption. Keep the questions and answers brief, with no more than 20 words each. Using ”User” and ”Assistant” as roles for questions and answers, respectively. Answer as if the ”Assistant” can directly watch the video. Speak as a friendly and helpful assistant.

Instance QA

Generate a concise dialogue about the {instance_class} with factual questions and their answers based on the following video segment caption:

{segment_caption}

The answers should be directly inferred from the provided segment caption. Keep the questions and answers brief, with no more than 20 words each. Using ”User” and ”Assistant” as roles for questions and answers, respectively. Answer as if the ”Assistant” can directly watch the video. Speak as a friendly and helpful assistant.

Inferential Segment Localization

Hello, and I’d like you to act as a question generator. You will receive a sentence describing a clip in one video, and your task is to generate two questions with hypothetical scenario contexts to test if a deep learning model can retrieve the given clip based on the provided scenario information by asking about ”what scene would you see” and ”which clip should I watch” under that circumstance. Below are a few examples:

<Example 1>

[Clip Content]

A elderly man is giving a speech in front of a blackboard.

[Question]

1. Suppose you are a college student and you are in class one morning. Which clip might demonstrate the scene in front of you at this time?
2. If I want to know how older generations teach classes, which clip should I watch?

<Example 2>

[Clip Content]

A young woman is seen standing in a room and dancing around.

[Question]

1. You are a dance instructor. You are coaching your students in preparation for the next dance competition. What might you see at this moment? Please find the clip which might show this scene.
2. I want to learn to dance. Could you please tell me which clip of this video I should watch?

<Example 3>

[Clip Content]

A dog in socks walks slowly out onto the floor as a lady films him.

[Question]

1. A female animal behavior researcher is studying the walking patterns of dogs when their feet tend to slip in your laboratory. Which clip in the video might you see at this point?
2. I’m feeling anxious and need to watch some funny animal videos to relax. Could you please help me find such a clip in the video?

Now given the following clip content sentence, please generate two questions with hypothetical scenario contexts to test if a deep learning model can retrieve the given clip based on the provided scenario information by asking about ”what scene would you see” and ”which clip should I watch” under that circumstance.

[Clip Content]

{content}

Cross Segment QA

You are a visual assistant. Given several video clip descriptions, you are tasked to generate a concise factual question and its answer by combining information from all the video clip descriptions. Note that the {instance_class} of all video clip descriptions in an input are the same, namely the video clip descriptions could describe the {instance_class} at different time points. The answer should be directly inferred from the provided sentence. Keep the question and answer brief. Using ”User” and ”Assistant” as roles for questions and answers, respectively. Speak as a friendly and helpful assistant. Below are some examples:

<Example 1>

[Input]

15.50s-30.75s : A group of children play in a park, running around and laughing.
45.20s-58.90s : A dog chases a frisbee, jumps to catch it, and returns it to its owner.

[Output]

User: What activities do the children and the dog engage in during the given video clips?

Assistant: The children play in a park, running and laughing, while the dog chases a frisbee, jumps to catch it, and returns it to its owner.

<Example 2>

[Input]

10.50s-30.88s : A chef in a white apron chops vegetables on a wooden cutting board.
50.20s-63.40s : A close-up of a sizzling steak on a hot grill.
80.16s-92.74s : A chef takes freshly baked bread out of the oven and places it on a cooling rack.

[Output]

User: What cooking activities can be observed in the video?

Assistant: The video shows a chef chopping vegetables, frying steak on a hot grill, and taking freshly baked bread out of the oven.

Now given the following video clip descriptions, please generate a question-answer pair as it is in the examples. Note that the {instance_class} of all video clip descriptions in an input are the same, namely the video clip descriptions may show the events or actions surrounding the {instance_class} at different time points. The answer should be directly inferred from the provided sentence. Keep the question and answer brief. Using ”User” and ”Assistant” as roles for questions and answers, respectively. Speak as a friendly and helpful assistant.

[Input]

{segment_caption}

Instance Activity Summarizing

Hello, I want you to act as a comprehensive video captioner. You will receive a list of clip-by-clip descriptions extracted from one video. Since some of these descriptions might be noisy, you should comprehend the major content of the video, assess the correctness of different pieces of information, and filter out erroneous, repetitive, noisy, or irrelevant details. After receiving and analyzing all these descriptions, please generate a comprehensive caption that effectively summarizes the events taking place in the video about the {instance_class}. Below are the clip descriptions extracted from the video:

{descriptions}

There are some requirements that you should follow:
1. Your comprehensive video caption should be comprehensive, concise, informative, and LESS than 20 words.
2. You should output ONLY THE COMPREHENSIVE VIDEO CAPTION, and NO OTHER CONTENTS should be printed.
3. Your comprehensive video caption MUST NOT mention the concept of ”frame” or ”video”.
Now please print out your comprehensive video caption.

The comprehensive video caption:

Composed Retrieval

Hello, and I’d like you to act as a question generator. You will receive two descriptions about one source clip and one target clip. Your task is to generate one question with differences of the two clips to test if a deep learning model can retrieve the target clip based on the content of the source clip by asking about ”could you please find a clip with following differences”. Below are some examples:
<Example 1>

[Source Clip Content]

An old professor is giving a lecture in front of a blackboard.

[Target Clip Content]

An elderly man is giving a speech in front of a blackboard, holding a ruler.

[Major Differences]

The man in the target clip content is holding a ruler.

[Instruction]

Please watch the {{SOURCE_CLIP}}. Could you please find a similar clip, but this time the speaker is holding something at hand?

<Example 2>

[Source Clip Content]

A beautiful scene of primeval forest.

[Target Clip Content]

A beautiful view of coral reef taken in shallow sea.

[Major Differences]

The scene in the target clip is a seascape rather than a forest landscape.

[Instruction]

Please watch the {{SOURCE_CLIP}}. Is there any similar clip with a different kind of scenery?

Now given the following descriptions about one source clip and one target clip, please generate one question with differences of the two clips to test if a deep learning model can retrieve the target clip based on the content of the source clip by asking about ”could you please find a clip with following differences”.

[Source Clip Content]

{source_clip_content}

[Target Clip Content]

{target_clip_content}