RESEARCH ARTICLE

LLaVA-MR: Large Language-and-Vision Assistant for Video Moment Retrieval

Jian Li[1]

1 Independent researcher

Abstract

Multimodal Large Language Models (MLLMs) are widely used for visual perception, understanding, and reasoning. However, long video processing and precise moment retrieval remain challenging due to LLMs' limited context size and coarse frame extraction. We propose the Large Language-and-Vision Assistant for Moment Retrieval (LLaVA-MR), which enables accurate moment retrieval and contextual grounding in videos using MLLMs. LLaVA-MR combines Dense Frame and Time Encoding (DFTE) for spatial-temporal feature extraction, Informative Frame Selection (IFS) for capturing brief visual and motion patterns, and Dynamic Token Compression (DTC) to manage LLM context limitations. Evaluations on benchmarks like Charades-STA and QVHighlights demonstrate that LLaVA-MR outperforms 11 state-of-the-art methods, achieving an improvement of 1.82% in R1@0.5 and 1.29% in mAP@0.5 on the QVHighlights dataset. Our implementation will be open-sourced upon acceptance.

1. Introduction
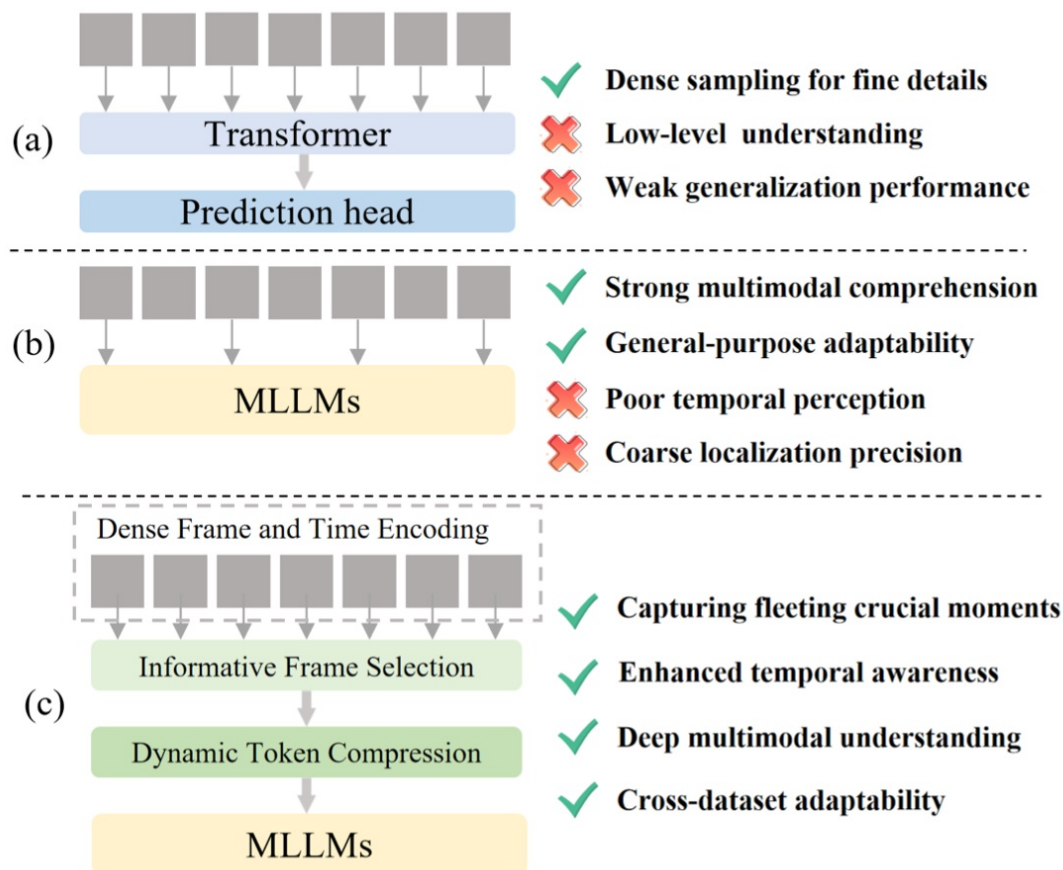
Figure 1. Advantages and Comparison of LLaVA-MR over prior work. (a) Traditional transformer-based methods. (b) Previous MLLMs-based methods. (c) Our LLaVA-MR.

Large language models (LLMs) are widely recognized for their strong semantic understanding capabilities. Multimodal large language models (MLLMs) [1][2], which extend LLMs to process multimodal data like video, have shown superior performance in tasks such as video description and visual question answering. They are also effective for specific visual tasks like object detection and visual grounding. However, using MLLMs for moment retrieval in long-duration videos is still a major challenge, as it requires a deep understanding of visual changes and precise integration of temporal information.

Video Moment Retrieval (MR) seeks to locate specific time segments within an untrimmed video based on a natural language query describing an event, requiring the model to effectively understand the video content and precisely capture the transient details relevant to the query. As illustrated in Fig. 1(a), previous approaches, such as Moment-DTER[3], CGDETR[4], UniLoc[5], and UniMD[6], have primarily tackled this task by utilizing transformer structures to fuse image-text features, with task-specific prediction heads producing a fixed set of moment candidates. However, these methods are limited by their reliance on frame-level feature extraction, which limits understanding, and the design of prediction heads reduces robustness when applied to new data. Mr. BLIP[7] represents a breakthrough by framing moment retrieval as an open-ended sequence-to-sequence problem with a generative MLLM, offering strong multimodal understanding and versatile adaptability, as shown in Fig. 1(b). However, this method is constrained by the limited context length of LLMs to

process long sequences, which forces it to use sparse frame sampling, leading to poor temporal perception and coarse localization precision.

We identify two key challenges limiting the effectiveness of MLLMs in moment retrieval: (1) improving their temporal awareness and perception to enable more precise alignment of video frames with time encoding, and (2) ensuring more accurate localization of brief moments within long videos. Due to limited context size, MLLMs often rely on sparse sampling, which reduces their ability to capture transient visual details in longer videos.

To address these challenges, we propose the **Large Language-and-Vision Assistant for Moment Retrieval (LLaVA-MR)**, which optimizes MLLMs for moment retrieval tasks by enhancing their ability to capture critical, brief moments within long video frame sequences, as illustrated in Fig. 1(c). To enhance temporal awareness and achieve higher-precision temporal perception, we apply Dense Frame and Time Encoding (DFTE) to extract fine-grained spatial and temporal features. To precisely locate brief moments in long videos, we developed an Informative Frame Selection (IFS) strategy to capture short-lived visual and motion patterns, and applied Dynamic Token Compression (DTC) to reduce sequence length while preserving essential information. This enables MLLMs to retrieve moments over extended video frame sequences effectively. Our approach achieves state-of-the-art results on widely used benchmarks, including Charades-STA[8] and QVHighlights[9]. Notably, it achieves an improvement of 1.82% in R1@0.5 and 1.29% in mAP@0.5 on the QVHighlights dataset.

In summary, our main contributions are the following:

- We propose LLaVA-MR, which applies MLLMs to the moment retrieval task, enhancing temporal awareness and effectively capturing critical, transient information in long videos.
- We introduce Dense Frame and Time Encoding (DFTE) to extract fine-grained spatial and temporal features, Informative Frame Selection (IFS) to capture critical dynamic moments, and Dynamic Token Compression (DTC) to compress redundant information.
- Our method achieves state-of-the-art performance on the moment retrieval task across both the Charades-STA and QVHighlights datasets, demonstrating a greater advantage on more complex, longer videos.

2. Related Work

2.1. Moment Retrival

Moment Retrieval (MR) aims to identify video segments that match natural language queries, with Temporal Action Detection (TAD) [10] serving as a sub-task to localize and classify actions within untrimmed videos. Early MR methods, such as TALL[8], relied on sliding window techniques for temporal localization, though these were computationally expensive. Hendricks[11] improved efficiency by introducing attention-based boundary prediction. Recent architectures, such as Semantic Conditioned Dynamic Modulation by Yuan[12] and the Moment Alignment Network (MAN) by Zhang[13], dynamically adjust video features based on query semantics and iteratively align video and text using graph adjustments.

To capture temporal dependencies, Zhang[14] introduced a 2D Temporal Adjacent Network employing 2D convolutions. In TAD, multi-stage CNNs[15], TCNs[16], and BSN[17] improved boundary proposals, evolving into structured models like SSN[18] and GCNs[19]. Transformer-based models, such as Girdhar's attention-based model[20] and ActionFormer by Zhang[21], along with multi-scale techniques by Dai[22][23] and enriched context modeling by Zhu[24] and Sardari[25], have further advanced temporal modeling within MR. Additionally, transformer-based models have gained prominence, with Liu[26] introducing Unified Multi-Modal Transformers (UMT) for joint moment retrieval and highlight detection. Mun[27] improved temporal grounding by integrating local-global video-text interactions through multi-level architecture. These methods, while promising, often suffer from annotation bias due to the structure of the prediction head, which impacts their generalization. LLaVA-MR, built on MLLMs, not only offers greater general-purpose versatility but also demonstrates improved performance due to its deeper multimodal understanding.
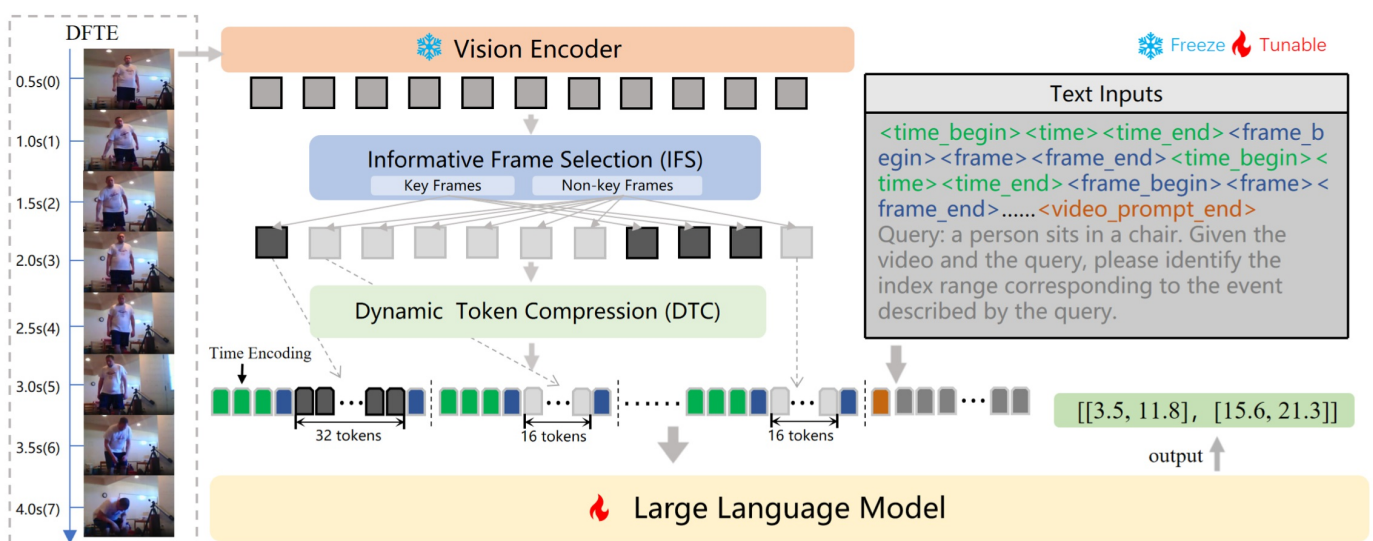
Figure 2. LLaVA-MR model overview. We leverage a pretrained MLLM such as BLIP-2[28]. Our model primarily consists of Dense Frame and Time Encoding (DFTE), Informative Frame Selection (IFS), and Dynamic Token Compression (DTC).

2.2. Multimodal Large Language Models

Building on the success of Large Language Models (LLMs), researchers are advancing Multimodal Large Language Models (MLLMs) to enhance cross-modal understanding by integrating visual, auditory, and textual data. Some methods[29][30][31][32] enhance LLMs by incorporating additional parameters, such as gated cross-attention layers or adapter layers, to enable them to process multimodal inputs. Other approaches[33][34][35] utilize modules like projection layers or Q-Formers to map the outputs from visual encoders into the input space of LLMs. Video LLMs[36][37][38][39] extend MLLMs for video tasks, primarily employing projection layers or Q-Formers for mapping visual tokens. Mr. BLIP[7], leveraging MLLMs, has achieved state-of-the-art results in moment retrieval with better semantic understanding and generalization. However, Mr. BLIP's sparse sampling, constrained by the LLM's window size, tends to miss fleeting visual details. Building upon Mr. BLIP, we capture more granular visual and temporal information with Dense Frame and Time Encoding (DFTE), while using Informative Frame Selection (IFS) and Dynamic Token Compression

(DTC) to preserve critical dynamic moments and compress redundant data.

3. Method

Our work enhances MLLMs for moment retrieval by improving temporal perception and capturing critical moments in long video sequences. § 3.1 provides an overview of the LLaVA-MR architecture.§ 3.2 details the **Dense Frame and Time Encoding (DFTE)**, which extracts fine-grained visual features with time encoding for improved temporal awareness§ 3.3 covers **Informative Frame Selection (IFS)** for capturing critical, brief moments within long video frame sequences. Finally, § 3.4 describes how the **Dynamic Token Compression (DTC)** reduces sequence length while preserving essential information. § 3.5 outlines the model training and inference process.

3.1. LLaVA-MR Architecture

Our model architecture is illustrated in Fig. 2. LLaVA-MR starts with a frozen image encoder that extracts visual features from all sampled frames, generating a sequence of image embeddings defined as $\mathbf{F}^V = [\mathbf{f}_1^V, ..., \mathbf{f}_N^V] \in \mathbb{R}^{N \times P \times D_0}$, where N represents the number of video frames, P is the number of patches, and D_0 is the visual feature dimension. We then apply Informative Frame Selection (IFS) to evaluate visual changes between frames, categorizing them as *key* or *non-key*. *Key* frames capture dynamic variations, often at event boundaries, while *non-key* frames show minimal change and carry more redundant information. Next, Q-Former, a query-based transformer, refines each frame embedding by extracting relevant information. *Key* frame embeddings are denoted as $\mathbf{K}^q = [\mathbf{f}_{i_1}^q, \mathbf{f}_{i_2}^q, ..., \mathbf{f}_{i_k}^q] \in \mathbb{R}^{k \times Q \times D_1}$, and *non-key* frame embeddings are denoted as $\mathbf{N}^q = [\mathbf{f}_{j_1}^q, \mathbf{f}_{j_2}^q, ..., \mathbf{f}_{j_{N-k}}^q] \in \mathbb{R}^{(N-k) \times Q \times D_1}$, where k is the number of *key* frames, Q represents the number of queries, and D_1 is the feature dimension after Q-Former. For *non-key* frames, we apply Dynamic Token Compression (DTC) within $\mathbf{N}^q$ to reduce sequence length while preserving essential information. Both $\mathbf{K}^q$ and the compressed $\mathbf{N}^q$ are projected into the language space, forming a unified embedding space, which is defined as $\mathbf{F}^L = [\mathbf{f}_1^L, \mathbf{f}_2^L, ..., \mathbf{f}_N^L]$.

We then construct an interleaved multimodal input sequence consisting of time tokens t_i, frame embeddings $\mathbf{f}_i^L \in \mathbf{F}^L$, a query q, and a task prompt p. We position each time token directly before its corresponding frame embeddings, using special tokens T_B, T_E, F_B, and F_E to mark the start and end of the time and frame segments. Our interleaved multimodal sequence is structured as:

$$x = [T_B, t_1, T_E, F_B, \mathbf{f}_1^L, F_E, ..., q, p]$$

where t_1 represents the first time token, and $\mathbf{f}_1^L$ represents the first frame embedding.

Finally, this sequence is fed into the LLM to predict the time segments relevant to query q. Following the approach of Mr. BLIP[7], the LLM is tasked with predicting the sequence of potentially multiple relevant moments, each defined by a start and end time in seconds. The prediction follows the format of a nested list of moments: $y = [[t_1^{start}, t_1^{end}], [t_2^{start}, t_2^{end}], ...]$.

3.2. Dense Frame and Time Encoding

We use Dense Frame and Time Encoding (DFTE) to capture detailed visual information and associate each frame with its corresponding time, enhancing the temporal awareness of MLLMs. Previous MLLM-based methods primarily rely on sparse sampling, which risks losing brief yet essential details. To address this, we increase the number of sampled frames, effectively capturing dense visual information and ensuring that critical transitions are not missed.

To improve MLLMs' temporal awareness, we propose choosing time encoding based on the frame sampling rate, using either **relative frame indices** or **timestamps** in seconds to avoid confusion from split tokens in large or decimal values.

We define the **sampling rate** R_{frames} as the ratio of the total sampled frames to the video duration $R_{frames} = \frac{N}{T}$), where N is the number of sampled frames, and T is the video length. Two design scenarios are considered: (1) When $R_{frames} \geq 1$, using relative frame indices as time tokens works best. Rounding timestamps in seconds to the nearest integer can create overlaps, which may confuse the LLM. For example, timestamps like 2.8 and 3.3 seconds would both round to 3, causing confusion for the LLM. In contrast, frame indices provide a clearer, non-overlapping representation. (2) When $R_{frames} < 1$, where frames are sampled more sparsely, and the time interval between adjacent indices is relatively large. Here, using timestamps as time tokens provides a clearer temporal reference. § 4.3 will discuss additional temporal representations and explore the effect of different time representations as time tokens regarding model performance.

3.3. Informative Frame Selection

Dense sampling captures detailed visual information but adds redundancy, making it crucial to isolate key visual and motion patterns. Brief transitions, like setting down a cup before picking up a phone, often hold more useful information, which is unevenly distributed across frames. To address this, we introduce the **Informative Frame Selection (IFS)** module, which captures brief yet crucial visual and motion variations.

In the IFS module, We categorize frames into two types: *key* frames and *non-key* frames. *Key* frames capture notable changes, often marking important event transitions and carrying valuable information. In contrast, *non-key* frames show minimal change, typically representing ongoing activities already covered by previous frames. So how do we distinguish between key frames and non-key frames? For $\mathbf{F}^v$ obtained from the image encoder, we first compute the change between adjacent frames by subtracting each frame from the preceding one, denoted as $\Delta \mathbf{F}^v = [\Delta \mathbf{f}_1^v, \ldots, \Delta \mathbf{f}_N^v] \in \mathbb{R}^{N \times P \times D_0}$, where each component represents the frame-to-frame variation across patches. For each $\Delta \mathbf{f}_i^v$ in $\Delta \mathbf{F}^v$, we then compute its L2 norm to obtain $\mathbf{d} = [d_1, d_2, \ldots, d_N]$. To further reduce the impact of video jitter, we apply a Gaussian filter to $\mathbf{d}$, resulting in the adjacent feature distance $\hat{\mathbf{d}}$, where each element $\hat{d}_i$ represents the change of the i-th frame relative to the previous one:

$$\hat{d}_i = \begin{cases} \max(\mathbf{d}), & \text{if}~i = 0 \\ G(\|\Delta \mathbf{f}_i^v\|_2), & \text{if}~i > 0 \end{cases}$$

where, $\|\Delta \mathbf{f}_i^v\|_2$ denotes the L2 norm of the frame difference matrix $\Delta \mathbf{f}_i^v$, and G represents the Gaussian filter applied to

smooth $\mathbf{d}$. Then, we select the top k frames with the highest $\hat{d}$ values as key frames, while the remaining frames are classified as non-key frames. We can define this formally as follows:

$$\mathbf{K}^V = [\mathbf{f}^V_{i_1}, \ldots, \mathbf{f}^V_{i_k}], i_j \in \mathrm{top}_k(\hat{\mathbf{d}})$$

$$\mathbf{N}^V = [\mathbf{f}^V_{j_1}, \ldots, \mathbf{f}^V_{j_{N-k}}], j_l \notin \mathrm{top}_k(\hat{\mathbf{d}})$$

where $\mathbf{K}^V$ represents the image embeddings of *key* frames, and $\mathbf{N}^V$ represents image embeddings of *non-key* frames. We will discuss the selection of the total frame number N and the setting of k in § 4.3.
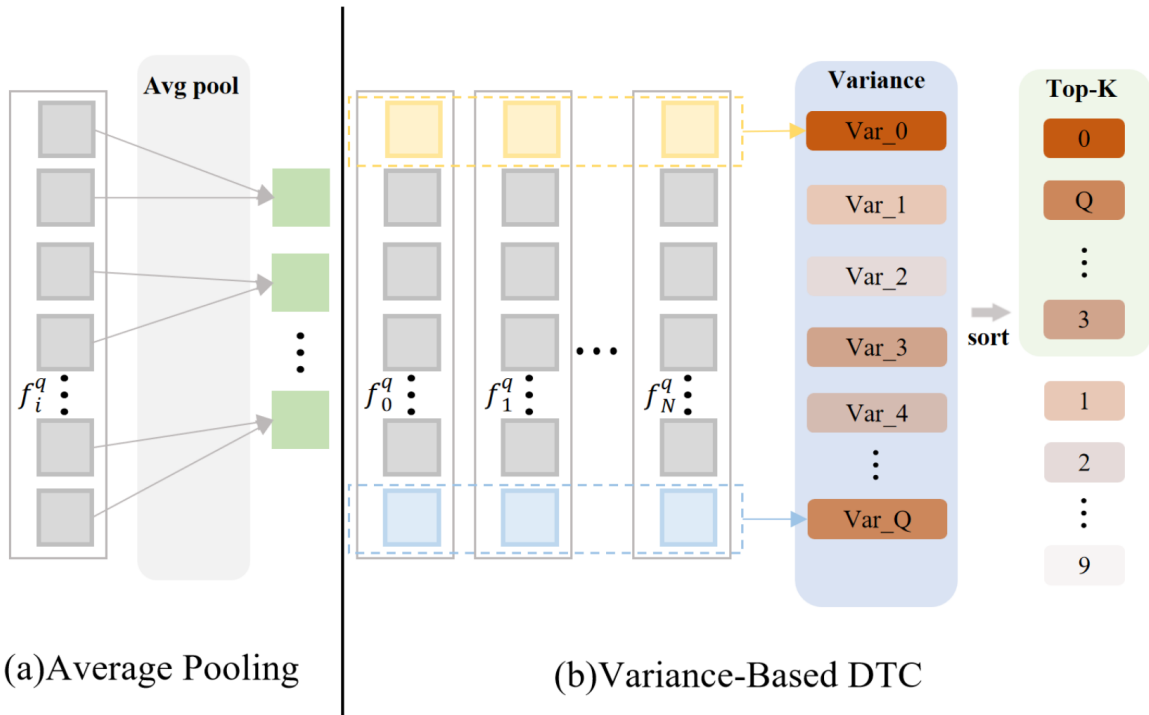
Figure 3. Two compression methods in **DTC**. (a) Average Pooling reduces tokens by averaging feature values within a window of 2, where $f^q_i \in N^q$. (b) Variance-Based DTC calculates the variance of each query in the Q-Former across all frames (where Q is the number of queries), sorts queries by variance, and selects the top half of queries to retain tokens focusing on dynamic content.

3.4. Dynamic Token Compression

As highlighted in § 3.3, *non-key* frames typically exhibit fewer dynamic changes, often representing the repetitive or static continuation of an event. On the other hand, when frame sequences are too long, this not only reduces the performance of the LLM but also leads to significant resource consumption. Therefore, we employ the Dynamic Token Compression (DTC) module to compress *non-key* frame embeddings, thereby reducing the sequence length while retaining valuable information.

After dividing the frames into *key* and *non-key* frames, we input $\mathbf{K}^V$ and $\mathbf{N}^V$ into the Q-Former to refine the embeddings. The refined embeddings for *key* and *non-key* frames are $\mathbf{K}^q \in \mathrm{R}^{k \times Q \times D_1}$ and $\mathbf{N}^q \in \mathrm{R}^{(N-k) \times Q \times D_1}$ (§ 3.1). We propose two

compression methods: average pooling, and Variance-Based Dynamic Token Compression (Variance-Based DTC). As shown in Fig. 3, average pooling reduces tokens by averaging feature values within a given window. In contrast, the variance-based DTC method aims to retain Q-Former queries that capture dynamic content in the video frames. Considering the idea that each query in the Q-Former focuses on different regions or aspects of the visual content, some queries, such as those tracking static elements like backgrounds, may offer little to no guidance in predicting relevant results. Based on this perspective, we calculate the variance of each query across frames, retaining only those queries with the highest variance—queries more likely to capture moving or dynamic aspects of the scene. § 4.3 will discuss the impact of different token compression methods on our model performance.

3.5. Model Training and Inference

During training, we fine-tune our generative MLLM to tackle moment retrieval by reframing it as an open-ended, sequence-to-sequence task in natural language. The model is trained using a maximum likelihood objective. For each input x and target y, the loss is minimized as:

$$L(x, y) = - \sum_{i=1}^{L-1} \log p_\theta(y_{i+1} \mid x, y_{1:i})$$

where L is the length of the target sequence. We employ the LoRA[40] parameter-efficient fine-tuning approach to effectively adapt with minimal additional parameters by injecting low-rank matrices into the model layers.

For inference, we sample frames and use the text decoder to generate the moment sequence, guided by the model's likelihood with beam search. The final sequence is output as a Python-style list of predicted moments, with minor formatting adjustments made as needed. Notably, when using relative frame indices as time encoding, we first record the mapping between these indices and actual timestamps, and then convert the output of the MLLM back to specific timestamps. This step is unnecessary when timestamps are used directly as the temporal representation.

4. Experiment

This section outlines the experimental details and evaluates the effectiveness of our design choices. In§ 4.1, we first provide an in-depth description of our experimental setup and implementation details. In § 4.2, We then present a comparison with the state-of-the-art methods and qualitative results from our approach. Finally, We explore the results of our ablation studies in § 4.3.

4.1. Experiment Setup

Benchmarks.

We validate our LLaVA-MR on the two most widely used video moment retrieval (MR) datasets:

Charades-STA[8] comprises 9,848 videos with an average duration of 30.6 seconds. The dataset includes 16,128 annotations, with an average moment duration of 8.1 seconds and an average query length of 7.22 words. It is split into two subsets: training (12,408 annotations) and testing (3,720 annotations).

QVHighlights[9] is a recent benchmark with 10,148 videos, each lasting 150 seconds. These videos are cropped from YouTube content and annotated with at least one query, averaging 11.3 words, to describe the relevant moments. The target moments have an average duration of 24.6 seconds. The dataset is divided into training, validation, and test sets, comprising 7,218, 1,150, and 1,542 queries, respectively. To ensure fair evaluation, the test set targets are kept hidden, and performance on the test split is only measurable through prediction submissions to the evaluation server.[1]

Metrics.

Common metrics for Moment Retrieval (MR) include Recall@K, mean Average Precision (mAP), and mean Intersection over Union (mIoU), each evaluated at different IoU thresholds. Recall@K represents the percentage of the top-K predicted segments with an IoU above a given threshold (e.g., R1@0.5 is Recall@1 at 0.5 IoU). mIoU measures the average IoU across predictions, while mAP indicates mean precision at specific IoU thresholds, such as mAP@0.5 and mAP@0.75. In brief, R1@0.5 and R1@0.7 focus on the top-1 prediction's recall at different IoU thresholds, mIoU measures average overlap, and mAP captures precision at varying IoUs.

Implementation details.

Our experimental setup mainly follows the configurations of Mr. BLIP[7]. Considering the relatively small scale of the datasets, we employ LoRA-based parameter-efficient finetuning to train only 0.63% of the total parameters in the model. Due to the output instability of LLMs, we mitigate minor formatting inconsistencies by applying post-processing heuristics to improve prediction accuracy. Our optimizer is AdamW[41], with an initial learning rate (LR) of 1e-8, linearly warmed up to 3e-4 over 10% of iterations, followed by cosine LR decay. Frames are sampled randomly during training.

For Charades-STA, we sample 60 frames per video, training for up to 50 epochs with a batch size of 16 across 4 A100-80GB GPUs, totaling around 70 GPU hours. For QVHighlights Captions, we sample 80 frames per video, train up to 50 epochs with a batch size of 32 on 8 A100-80GB GPUs, and set gradient accumulation to 4, totaling about 33 GPU hours.

4.2. Results

Comparison to the State-of-the-Art.

To evaluate our approach for moment retrieval, we compare LLaVA-MR against SoTA methods on two widely used datasets: Charades-STA and QVHighlights. As summarized in Table 1, LLaVA-MR achieves the highest performance across all key metrics on both datasets. This not only demonstrates the superiority of our approach but also confirms its robustness across datasets with varying complexities and multiple evaluation metrics. Notably, on the QVHighlights dataset, which contains longer videos and requires a deeper understanding compared to the Charades-STA dataset,

LLaVA-MR demonstrates a more substantial improvement. On the validation set, it achieves a **2%** increase in R1@0.5, while on the test set, it outperforms other methods with gains of **1.82%** in R1@0.5, **0.97%** in R1@0.7, **1.29%** in mAP@0.5, and **1.02%** in mAP@0.75. These results underscore our model's enhanced capability to handle moment retrieval tasks in longer, more complex videos.

Inference Time.

We compared the inference times of LLaVA-MR and MR. BLIP[7] with the same number of sampled frames. After averaging 10 tests on an A100-80G GPU, LLaVA-MR took 1.31s, while MR. BLIP took 1.57s. The addition of modules such as IFS was offset by the compression mechanism, resulting in faster inference.
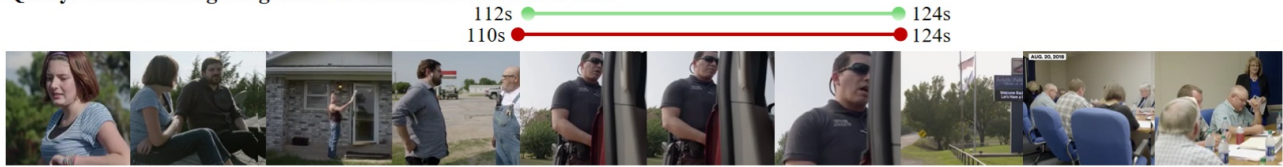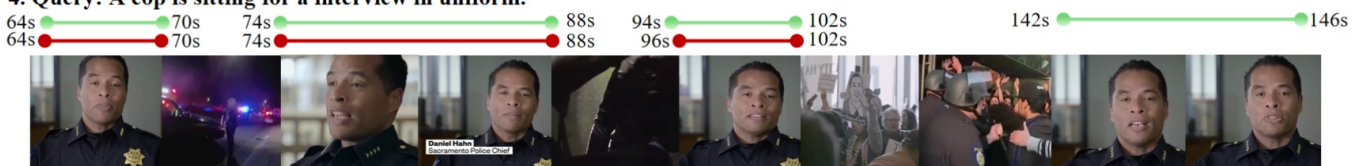
Qualitative Results.

Figure 4 visualizes qualitative results, with ground truth segments for query events alongside highlighted predicted intervals. In examples 1, 2, and 3, LLaVA-MR accurately identifies the segments matching the natural language queries, with predicted start and end times closely aligning with ground truth. This demonstrates the model's strong ability to perceive temporal boundaries. In example 4, the model not only outputs the correct segment but also detects an additional relevant segment at the video's end that was not annotated. Example 5 illustrates a failure case where the model predicts an obvious segment but misses another, less prominent one.

Table 1. Comparison with State-of-the-Art on Charades-STA and QVHighlights Datasets. CLIP*: UnLoc[44] pretrains the backbone with action classification datasets Kinetics 400/700. SF: Slow-Fast backbone.

Method	Backbone	mIoU	R1@0.5	R1@0.7	mAP@0.5	mAP@0.75
QVHighlights Validation set						
Moment-DETR[3]	SF+CLIP	–	59.68	40.84	–	–
EaTR[42]	I3D	–	61.36	45.79	61.86	41.91
QD-DETR[43]	SF+CLIP	–	62.68	46.66	62.23	41.82
UnLoc-L[44]	CLIP*	–	66.10	46.70	–	–
CG-DETR[4]	SF+CLIP	–	67.40	52.10	65.60	45.70
Mr. BLIP[7]	BLIP-2	–	76.13	63.35	69.39	55.78
LLaVA-MR	BLIP-2	**71.85**	**78.13**	**64.13**	**69.64**	**56.32**
QVHighlights Test set						
SeViLa[45]	BLIP-2	–	54.50	36.50	–	–
UniVTG[46]	SF+CLIP	–	58.86	40.86	57.60	35.59
QD-DETR[43]	SF+CLIP	–	62.40	44.98	62.52	39.88
CG-DETR[4]	SF+CLIP	–	65.43	48.38	64.51	42.77
Mr. BLIP[7]	BLIP-2	–	74.77	60.51	68.12	53.38
LLaVA-MR	BLIP-2	–	**76.59**	**61.48**	**69.41**	**54.40**
Charades-STA Test set						
Moment-DETR[3]	–	–	53.63	31.37	–	–
QD-DETR[43]	SF+CLIP	–	57.31	32.55	–	–
UniVTG[46]	SF+CLIP	50.10	58.01	35.65	–	–
CG-DETR[4]	SF+CLIP	50.13	58.44	36.34	–	–
UnLoc-L[5]	CLIP*	–	60.80	38.40	–	–
UniMD+Sync[6]	–	–	63.98	44.46	–	–
InternVideo2-1B[47]	–	–	68.36	45.03	–	–
EaTR[42]	I3D	–	68.47	44.92	–	–
Mr. BLIP[7]	BLIP-2	58.63	69.31	49.29	–	–
InternVideo2-6B[47]	–	–	70.03	48.95	–	–
LLaVA-MR	BLIP-2	**59.78**	**70.65**	**49.58**	**69.96**	**39.66**

Figure 4. Qualitative results on the Charades-STA and QVHighlights datasets, with ground truth segments for query events alongside highlighted predicted intervals.

4.3. Ablation Studies

To evaluate the contribution of each component in our approach, we conducted a comprehensive ablation study on the Charades-STA dataset. Table 2 shows these results. The baseline (a) involves sparse sampling from the video and inputting it into the MLLM, achieving an R1@0.5 score of **65.48**, confirming the feasibility of using MLLM for the moment retrieval task. In experiment (b), we simply increased the number of sampled frames based on the baseline to capture more visual information, which led to an improvement of **+0.89** in R1@0.5. Experiment (c) builds upon (b) by optimizing temporal representation, enhancing the model's perception of temporal sequences, resulting in a further improvement of **+2.15** in R1@0.5. Finally, in experiment (d), we incorporated the IFS and DTC modules to capture crucial moment and compress redundant information, leading to a significant improvement of **+2.13** in R1@0.5.

Table 2. Comprehensive ablation study on Charades-STA

	Dense Frame	Time Encoding	IFS&DTC	R1@0.5	R1@0.7	mIoU
(a)				65.48	42.71	55.17
(b)	✓			66.37	44.65	57.50
(c)	✓	✓		68.52	47.29	58.16
(d)	✓	✓	✓	**70.65**	**49.58**	**59.78**

Table 3. Ablation on Time Encoding. Evaluation on Charades-STA and QVHighlights.

Time Representation	R1@0.5	R1@0.7	mIoU
Charades-STA Test Set			
Relative frame indices	**67.83**	**46.68**	**57.78**
Timestamps	66.37	44.65	57.50
QVHighlights Validation Set			
Relative frame indices	74.77	61.74	68.45
Timestamps	**76.06**	**62.65**	**70.01**

Design of Time Encoding.

Designing an effective time encoding method is essential for enabling MLLMs to associate each frame with its corresponding time. Common approaches include using frame indices or timestamps to indicate each frame's position. Frame indices can be either relative, representing the order within the sampled sequence (e.g., $1, 2, 3, …, 60$), or absolute, marking the frame's position in the full video (e.g., $1, 15, 30, …, 900$). Since larger numbers are often split into multiple tokens during tokenization, leading to potential numerical confusion and lower performance for LLMs[7], we focus on relative indices over absolute ones. Similarly, directly inputting decimal timestamps can cause tokenization issues; thus, rounding to the nearest integer is a common approach, which we apply as well when using timestamps.

We selected the Charades-STA and QVHighlights datasets to validate the hypothesis that different time representations should be chosen based on the sampling rate, R_{frames} (see § 3.2). In Table 3, we show the impact of using different time representations on the Charades-STA and QVHighlights datasets. For the Charades-STA dataset (with an average video length of 30 seconds, each video sampled at 60 frames, $R_{frames} \approx 60/30 > 1$), we observed that using relative frame indices improved the R1@0.5 score by +1.46 compared to using timestamps. We infer this occurs because when $R_{frames} \geq 1$, neighboring timestamps rounded to integers can become identical (e.g., 0.7 and 1.1 both round to 1), causing MLLMs to confuse the corresponding frames. Hence, using incremental relative frame indices as the time encoding provides a clearer representation in this case. Conversely, for the QVHighlights dataset (with an average video length of 150 seconds, each video sampled at 80 frames, $R_{frames} \approx 80/150 < 1$), using frame indices as time tokens led to better performance, improving the R1@0.5 score by +1.29. We hypothesize that this happens because the time gap between sampled frames is relatively large. For example, a frame with a relative index of 1 could represent 0.2 or 1.4 seconds, leading to coarser time granularity and information loss. Timestamps, however, provide finer precision.

Additionally, to enhance the model's ability to distinguish between tokens representing temporal and frame-based information, we introduced special tokens: <time_begin>, <time_end>, <frame_begin>, and <frame_end>. We evaluated their impact on Charades-STA, with results presented in Table 4, demonstrating their effectiveness in improving model performance.

Table 4. Ablation evaluation of prompt design on Charades-STA. B/E are special tokens indicating the begin/end of time or frames.

Input Format	R1@0.5	R1@0.7	mIoU
No special tokens	67.83	46.68	57.78
B/E for time	67.98	46.80	57.93
B/E for frames	68.25	47.07	58.02
B/E for both	**68.52**	**47.29**	**58.16**

Table 5. Ablation Study on Different Token Compression Methods. Evaluation on Charades-STA.

Compression Method	R1@0.5	R1@0.7	mIoU
No Compression	68.52	47.29	58.16
Average Pooling	69.43	48.17	58.77
Variance-Based DTC	**70.65**	**49.58**	**59.78**

Design of Informative Frame Selection.

The Informative Frame Selection (IFS) module introduced in § 3.3 captures brief but critical visual and motion variations. This module classifies frames into key and non-key categories by calculating the adjacent feature distance $\hat{d}$. We selected a sample video and plotted $\hat{d}$, as shown in Fig. 5. It can be seen that $\hat{d}$ peaks when the person picks up or puts down the guitar, while remaining low during the guitar playing. This supports the hypothesis that frames with more significant dynamic changes not only contain more valuable visual information but are also more likely to lie at event boundaries, while frames within event interiors remain relatively static.

We select the top-k frames with the highest change levels as key frames, with the remaining frames classified as non-key. The selection of the total number of frames N and top-k requires a balance between capturing sufficient visual information and managing the sequence length constraints of the LLM input. As illustrated in Figure 4, we observe that when k is too small, excessive compression results in a significant loss of visual information, reducing model performance. Conversely, when k is too large, or even equals N, redundant information increases the sequence length excessively, which also

degrades model performance. Similarly, increasing N does not always improve performance; while N should ensure comprehensive sampling of visual variations, overly large values introduce longer sequence lengths and more redundant information, which can interfere with model performance. We found that $N = 60$ and $k = 32$ yield the best performance on Charades-STA. Following a similar approach, we set $N = 80$ and $k = 32$ for the QVHighlight dataset.
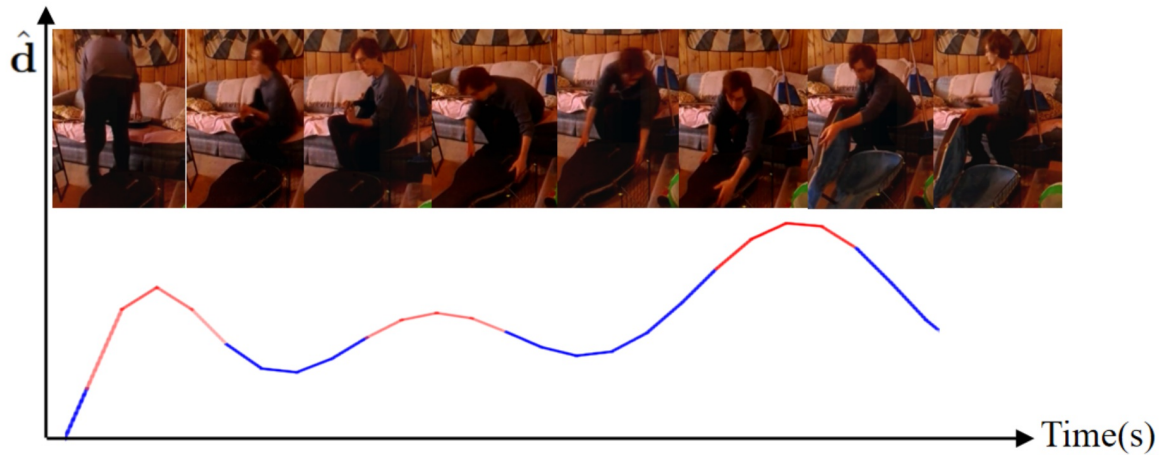

Figure 5. Temporal correspondence between video frames and adjacent frame feature distance $\hat{d}$.
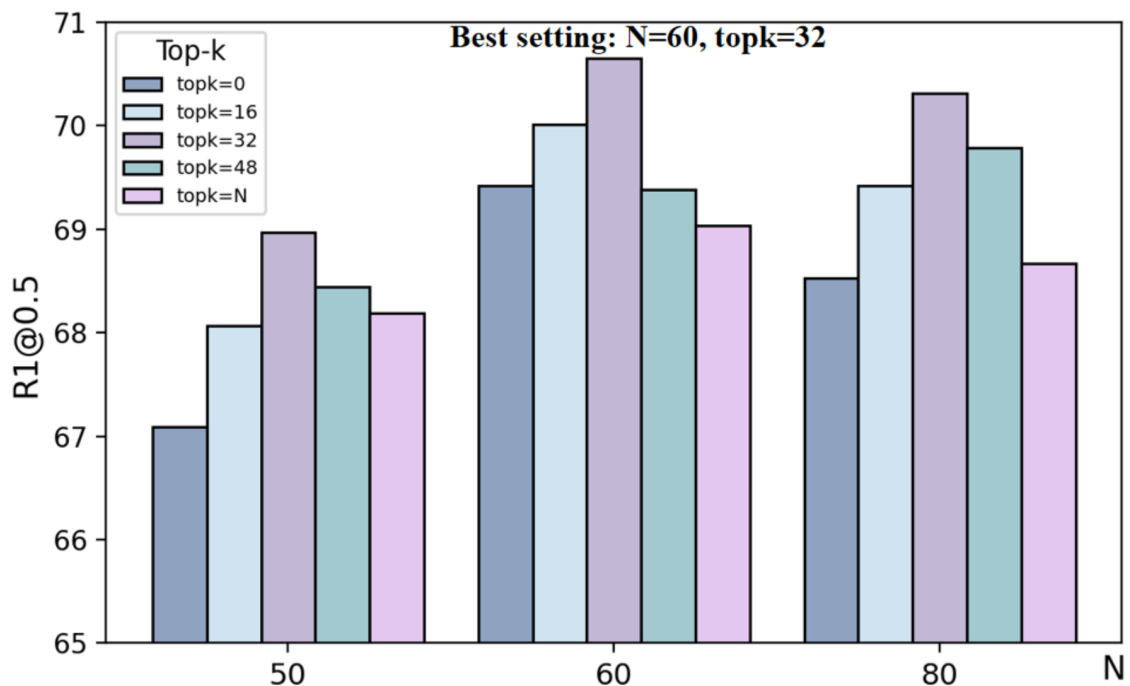
Figure 6. Hyperparameter optimization results on the Charades-STA dataset to determine the optimal values for N and k.

Design of Dynamic Token Compression.

After dividing the frames into key and non-key frames, we employ the Dynamic Token Compression (DTC) module from § 3.4 to compress the features of non-key frames, thereby reducing the sequence length and eliminating redundancy. This

compression strategy is designed to retain essential information while filtering out redundant data.

Table 5 shows the impact of different compression methods on the model's performance. Without using any compression strategy, the model achieves a score of **68.52** on R1@0.5. We first applied average pooling to N^q, reducing the token corresponding to each frame by half. This helped reduce redundancy and shorten the sequence length, achieving a score of **69.43** on R1@0.5. However, we infer that simply applying average pooling does not guarantee that more valuable information in N^q is preserved. This led us to the design of Variance-Based DTC, with an aim to retain Q-Former queries that capture dynamic content in the video frames. For example, in a video where a person is drinking water, a query focused on a static background element is unlikely to contribute meaningfully to understanding the action. Variance-Based DTC outperformed average pooling, improving R1@0.5 to **70.65**. We attribute this improvement to the fact that queries with higher variance are more likely to capture dynamic aspects of the scene, while lower-variance queries correspond to relative stable features across frames and can be more effectively complemented by features from other frames after compression.

5. Conclusion

We introduce LLaVA-MR, a novel approach that optimizes MLLMs for moment retrieval in long videos by enhancing their capability to capture brief but critical moments within extensive sequences. Our method combines Dense Frame and Time Encoding (DFTE), Informative Frame Selection (IFS), and Dynamic Token Compression (DTC), creating a robust framework for precise moment retrieval. Through comprehensive evaluations, LLaVA-MR demonstrates significant performance improvements, outperforming ten leading methods, especially on complex, long-video benchmarks like QVHighlights.

6. Additional Experiments

We present additional experimental results, including a comparison with state-of-the-art methods on the ActivityNet Captions dataset and more findings from hyperparameter optimization.

6.1. Comparison on ActivityNet Captions Dataset

ActivityNet Captions[48] is a large-scale dataset consisting of 20,000 videos, each averaging 2 minutes in duration. It includes 72,000 segments, each paired with a human-written caption averaging 13.5 words. The dataset is organized into three subsets: training (37,421 segments), validation 1 (val_1, 17,505 segments), and validation 2 (val_2, 17,031 segments). In our experiments, the training set is used to train the model, while the val_2 set is employed as the test set.

The results in Table 6 show that LLaVA-MR achieves state-of-the-art performance on the ActivityNet Captions dataset, outperforming all baseline methods in both R1@0.5 and R1@0.7 metrics. Specifically, it attains an R1@0.5 score of **55.16**, surpassing Mr. BLIP[7] by **1.24** points, highlighting the effectiveness of the LLaVA-MR architecture.

6.2. Hyperparameter Optimization

Compression strategies and ratios:

As discussed in the main paper, we investigated how different compression strategies impact model performance. Beyond the choice of strategy, the compression ratio is crucial. Each frame initially corresponds to 32 tokens, determined by the *query_number* of the Q-former, and the compressed token count is denoted as T. Low compression levels fail to reduce redundancy and alleviate sequence length constraints, while high compression levels lead to excessive information loss, degrading performance.

Table 7 shows that the model performs best with $T = 16$, achieving R1@0.5 of **70.65** and R1@0.7 of **49.58**. Performance drops significantly with both lower and higher compression levels, emphasizing the need for a balanced compression ratio.

Frame and top-k selection:

We also studied the optimal number of frames (N) and top-k values on the QVHighlights[3] dataset, aiming to balance sufficient visual information with the sequence length constraints of large language models (LLMs). Based on experiments on the Charades-STA[8] dataset, where $N = 60$ and $k = 32$ yielded the best result, we adjusted these parameters for QVHighlights, which features significantly longer videos. As shown in Table 8, the best performance was achieved with $N = 80$ and $k = 32$. Note, however, that these results were obtained without post-processing, so the identified configuration may not represent the best possible outcome overall.

Table 6. Comparison with State-of-the-Art on ActivityNet Captions. **CLIP***: UnLoc[44] pretrains the backbone with action classification datasets Kinetics 400/700.

Method	Backbone	R1@0.5	R1@0.7
DRN[49]	–	45.45	24.36
VLG-Net[50]	–	46.32	29.82
UniLoc-B[5]	CLIP*	48.00	29.70
UnLoc-L[5]	CLIP*	48.30	30.20
GVL[51]	–	49.18	29.69
Mr. BLIP[7]	BLIP-2	53.92	35.55
LLaVA-MR	BLIP-2	**55.16**	**35.68**

Table 7. Evaluation of Different Compression Ratios on Charades-STA. T is the number of tokens after compression, and the ratio is the number of compressed tokens divided by the original (32).

Ratio	T	R1@0.5	R1@0.7	mIoU
1:1	32	68.52	47.29	58.16
3:4	24	69.49	47.70	58.35
1:2	16	**70.65**	**49.58**	**59.78**
1:4	8	69.21	46.85	58.82

Table 8. Evaluation of Different Top-k and N Settings on QVHighlights Validation set.

N	Top-k	R1@0.5	R1@0.7	mAP@0.5	mAP@0.75
60	32	76.58	63.68	68.29	55.21
60	50	76.19	63.10	69.20	55.27
60	60	76.06	62.65	68.64	53.83
80	32	**76.97**	**63.74**	68.40	**55.30**
80	50	76.65	62.84	**68.95**	54.42
80	60	76.52	62.45	68.53	54.02
100	32	76.77	62.52	68.96	54.75
100	40	76.58	63.03	68.63	55.67

7. Post-Processing of LLM Outputs

Algorithm 1 Post-Processing Predicted Output

1: **function** `post_process(pred)`
2: pred ← clean input (e.g., remove extraneous markers)
3: **if** pred is invalid or empty **then**
4:     **return** "[[-1, -1]]"
5: **end if**
6: windows ← split pred into cleaned sub-windows
7: output ← empty list
8: **for** each window in windows **do**
9:     t_start, t_end ← extract and validate numbers
10:    **if** t_start > t_end **then**
11:       swap t_start and t_end
12:    **end if**
13:    append `[t_start, t_end]` to output
14: **end for**
15: **return** formatted output

Despite training LLMs specifically for the moment retrieval task and instructing them to output results in the format of a nested list of moments:

$$y = [[t_1^{start}, t_1^{end}], [t_2^{start}, t_2^{end}], \ldots]$$

LLMs can still produce outputs that deviate from the required structure due to inherent instability. For instance, instead of the expected output [[1.5, 4.3], [6.7, 9.2]], the LLM might generate malformed results such as [[1.5, 4.3],[6.7, 9.2] or [[1.5, 4.3][6.7, 9.2]], causing errors during parsing. To address this issue, we implemented a robust post-processing mechanism to normalize the outputs and ensure strict adherence to the required format. This approach effectively mitigates formatting inconsistencies and supports reliable downstream processing.
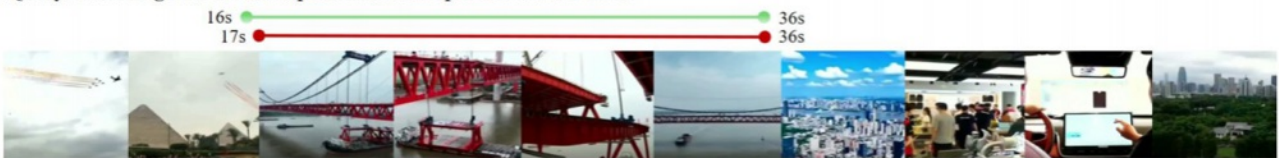
The `post_process` function refines predictions by removing extraneous markers (e.g., "</s>"), verifying compliance with the required nested list format, and correcting formatting issues. If the output does not match the expected structure—where each element contains a start and end time—the function attempts to extract valid number pairs. It splits the output into individual moment windows using special symbols, cleans each window by removing extra commas, adjusts the format, and extracts start and end times. These values are validated, adjusted to valid indices, and swapped if necessary. The processed windows are then appended to the output list.

This post-processing pipeline ensures the final output adheres to the specified nested list format. Testing the model on over 50,000 samples confirmed the robustness of this approach, successfully normalizing all outputs for further analysis and moment retrieval tasks.

Figure 7. Qualitative results across various types of videos, with ground truth segments for query events alongside highlighted predicted intervals.

8. More Qualitative Results

One of the key advantages of using MLLMs for moment retrieval tasks is their strong generalization and adaptability. Unlike transformer models with task-specific prediction heads, which are prone to annotation bias, MLLMs utilize cross-modal representations learned from large-scale multimodal data, enabling them to handle a wide range of video types effectively.

We demonstrated LLaVA-MR's effectiveness on datasets like Charades-STA and QVHighlights, which feature relatively simple videos with minimal scene transitions. To further evaluate its versatility, we tested LLaVA-MR on more complex videos from diverse domains, including news clips and movies. These results, shown in Figure 7, highlight the model's adaptability to intricate video styles and scenarios.

News Clips:

We evaluated LLaVA-MR on challenging examples from news videos:

- **Example 1:** A fast-paced news flash with frequent scene transitions and multiple reports, where LLaVA-MR accurately retrieved a moment showing "automated devices operating in a modern factory", demonstrating its ability to accurately locate specific moments amidst diverse and rapidly changing visuals.
- **Example 2:** A moderately paced news report on a bridge construction project. This coverage included various angles of the bridge and the construction process. The model accurately identified the relevant time intervals without missing any critical scenes.
- **Example 3:** A political report on the Paralympics, where LLaVA-MR correctly retrieved a segment featuring an interview with the Olympic Committee President. This result highlights LLaVA-MR's capability to handle complex queries, including those involving domain-specific terminology.

Movie Clips:

We evaluated LLaVA-MR on movie clips, which present greater challenges for moment retrieval due to their intricate scene transitions and complex narrative structures compared to short YouTube videos.

- **Example 4:** The model successfully captured the full context of a snowy scene, including the pivotal moment when the main characters meet and converse.
- **Example 5:** Despite the scene being filmed from multiple perspectives, depicting a boy talking with his mother in a car, LLaVA-MR accurately identified the complete interval.
- **Example 6:** In a scene where the male protagonist regains his freedom, the model divided the moment into two intervals, but their combined duration closely aligned with the ground truth.
- **Example 7:** The model mistakenly identified Zhu Bajie (a character from a Chinese classic) eating watermelon as a farmer eating watermelon, likely due to challenges in distinguishing the character's unique features.
- **Example 8:** While the model correctly identified key frames, it missed distant views, potentially due to resolution

limitations affecting the detection of smaller visual elements.

These results highlight LLaVA-MR's effectiveness but also suggest areas for improvement, such as increasing resolution and enhancing its ability to recognize fine-grained details.

9. Future research potential

The moment retrieval task has a wide range of applications, including video summarization, content-based retrieval, multimedia search, video recommendation systems, and educational video indexing. This section explores potential future directions for enhancing the performance of MLLMs in moment retrieval tasks.

A significant area for improvement is enabling MLLM-based models to output relevance scores for video clips with respect to a given query. Currently, LLaVA-MR is designed solely to predict the temporal intervals relevant to the query. However, unlike transformer-based methods equipped with prediction heads, LLaVA-MR lacks the capability to assign confidence scores to individual video clips. This limitation reduces its utility in applications that require score-based ranking or thresholding. Future research could focus on integrating MLLMs with prediction head structures, allowing the model to retain its strong contextual understanding for interval prediction while also generating relevance scores for each clip. Such an enhancement would bridge the gap between interpretability and robustness, significantly increasing the model's versatility in real-world applications.

Expanding the inclusion of additional modalities in training and inference processes offers a promising direction for advancing moment retrieval tasks. Currently, our approach primarily relies on video frame sequences and textual queries, leaving other modalities underutilized. On one hand, the information extracted from videos remains incomplete, as it focuses solely on visual content while neglecting the audio component. Developing effective methods to process and integrate audio alongside visual data is essential for enabling the model to reason holistically across both modalities. This integration could significantly enhance the model's ability to capture richer and more nuanced contextual information. On the other hand, the current prompt format is limited to text. Broadening this scope to include prompts in other formats, such as images or audio, could greatly expand the applicability of moment retrieval tasks. For example, an image prompt featuring a close-up of a bouquet of flowers could help identify the corresponding time segment where the bouquet appears in a video. Similarly, an audio prompt featuring cheering sounds could assist in locating scenes associated with such sounds. By accommodating diverse query formats, models could unlock new opportunities for moment retrieval in real-world applications, paving the way for more adaptable and robust solutions.

Furthermore, leveraging advanced strategies from LLMs, such as Chain-of-Thought (CoT) reasoning, Retrieval-Augmented Generation (RAG), and Reinforcement Learning (RL), presents promising pathways for enhancing MLLMs in moment retrieval tasks.Chain-of-Thought (CoT) reasoning involves breaking down complex reasoning processes into intermediate steps, enabling the model to address intricate tasks more effectively. In the context of moment retrieval, CoT could help the model analyze the relationships between time intervals in a sequential manner, thereby improving its ability to identify temporally complex events. Additionally, Retrieval-Augmented Generation (RAG) can enhance model

performance by incorporating external knowledge or context. For example, the model could use RAG to retrieve and integrate similar scenes or textual descriptions from external resources, improving localization accuracy. This approach could also help resolve ambiguities, such as distinguishing between visually similar objects in different contexts. Moreover, Reinforcement Learning (RL) offers a framework for iterative, reward-based training, optimizing model performance over time. In moment retrieval tasks, rewards based on the alignment between predicted and actual intervals could refine the model's temporal precision, ensuring better performance in tasks that require exact boundary identification. Together, these strategies provide compelling opportunities to deepen, adapt, and strengthen the effectiveness of MLLMs, enabling them to tackle the unique challenges of moment retrieval with greater precision and versatility.

Footnotes

1 evaluation server: https://codalab.lisn.upsaclay.fr/competitions/6937

References

1. ^Jin Y, Li J, Liu Y, Gu T, Wu K, Jiang Z, He M, Zhao B, Tan X, Gan Z, et al. Efficient multimodal large language models: A survey. arXiv preprint arXiv:2405.10739. 2024.

2. ^Li J, Lu W, Fei H, Luo M, Dai M, Xia M, Jin Y, Gan Z, Qi D, Fu C, Tai Y, Yang W, Wang Y, Wang C (2024). "A Survey on Benchmarks of Multimodal Large Language Models". arXiv. arXiv:2408.08632 [cs.CL].

3. a, b, c, dLei J, Berg TL, Bansal M. "QVHighlights: Detecting Moments and Highlights in Videos via Natural Language Queries". arXiv [cs.CV]. 2021. Available from:https://arxiv.org/abs/2107.09609.

4. a, b, c, dMoon W, Hyun S, Lee S, Heo J (2024). "Correlation-Guided Query-Dependency Calibration for Video Temporal Grounding". arXiv. arXiv:2311.08835 [cs.CV].

5. a, b, c, dShen Y, Xiong X, Nagrani A, Arnab A, Wang Z, Ge W, Ross D, Schmid C (2023). "UnLoc: A Unified Framework for Video Localization Tasks". arXiv. arXiv:2308.11062 [cs.CV].

6. a, bZeng Y, Zhong Y, Feng C, Ma L.UniMD: Towards Unifying Moment Retrieval and Temporal Action Detection. 2024. Available from: https://arxiv.org/abs/2404.04933.

7. a, b, c, d, e, f, g, h, i, j, kMeinardus B, Batra A, Rohrbach A, Rohrbach M. "The Surprising Effectiveness of Multimodal Large Language Models for Video Moment Retrieval". arXiv [cs.CV]. 2024. Available from: https://arxiv.org/abs/2406.18113.

8. a, b, c, dGao J, Sun C, Yang Z, Nevatia R. "Tall: Temporal activity localization via language query". InICCV. 2017. p. 5267–5275.

9. a, bLei J, Berg TL, Bansal M (2021). "Detecting moments and highlights in videos via natural language queries". NeurIPS. 34: 11846–11858.

10. ^Lin C, Li J, Wang Y, Tai Y, Luo D, Cui Z, Wang C, Li J, Huang F, Ji R. Fast learning of temporal action proposal via dense boundary generator. Proceedings of the AAAI conference on artificial intelligence 2020; 34(07):11499-11506.

11. ^Anne Hendricks LA, Wang O, Shechtman E, Sivic J, Darrell T, Russell B. Localizing moments in video with natural language. In: ICCV. 2017. p. 5803–5812.

12. ^Yuan Y, Ma L, Wang J, Liu W, Zhu W (2019). "Semantic conditioned dynamic modulation for temporal sentence grounding in videos". NeurIPS. **32**.

13. ^Zhang D, Dai X, Wang X, Wang YF, Davis LS. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In: CVPR. 2019. p. 1247–1257.

14. ^Zhang S, Peng H, Fu J, Luo J (2020). "Learning 2d temporal adjacent networks for moment localization with natural language". Proceedings of the AAAI Conference on Artificial Intelligence **34** (07): 12870–12877.

15. ^Shou Z, Wang D, Chang SF. "Temporal action localization in untrimmed videos via multi-stage cnns." InCVPR; 2016. p. 1049-1058.

16. ^Lea C, Flynn MD, Vidal R, Reiter A, Hager GD (2017). "Temporal convolutional networks for action segmentation and detection". In: CVPR. pp. 156–165.

17. ^Lin T, Zhao X, Su H, Wang C, Yang M. "Bsn: Boundary sensitive network for temporal action proposal generation." In: ECCV; 2018. p. 3-19.

18. ^Zhao Y, Xiong Y, Wang L, Wu Z, Tang X, Lin D. "Temporal action detection with structured segment networks." In: ICCV; 2017. p. 2914-2923.

19. ^Zeng R, Huang W, Tan M, Rong Y, Zhao P, Huang J, Gan C. Graph convolutional networks for temporal action localization. In: ICCV. 2019. p. 7094-7103.

20. ^Girdhar R, Carreira J, Doersch C, Zisserman A. Video action transformer network. In:CVPR; 2019. p. 244-253.

21. ^Zhang CL, Wu J, Li Y. Actionformer: Localizing moments of actions with transformers. InECCV. Springer; 2022. p. 492-510.

22. ^Dai R, Das S, Bremond F (2021). "Ctrn: Class-temporal relational network for action detection".arXiv preprint arXiv:2110.13473. Available from:https://arxiv.org/abs/2110.13473.

23. ^Dai R, Das S, Kahatapitiya K, Ryoo MS, Brémond F. "Ms-tct: Multi-scale temporal convtransformer for action detection". In: CVPR. 2022. p. 20041–20051.

24. ^Zhu Z, Tang W, Wang L, Zheng N, Hua G (2021). "Enriching local and global contexts for temporal action localization". In: ICCV. pp. 13516–13525.

25. ^Sardari F, Mustafa A, Jackson PJ, Hilton A. Pat: Position-aware transformer for dense multi-label action detection. In: CVPR. 2023. p. 2988-2997.

26. ^Liu Y, Li S, Wu Y, Chen CW, Shan Y, Qie X (2022). "Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection". In: CVPR. pp. 3042–3051.

27. ^Mun J, Cho M, Han B. Local-global video-text interactions for temporal grounding. In:CVPR; 2020. p. 10810-10819.

28. ^Li J, Li D, Savarese S, Hoi S. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". arXiv [cs.CV]. 2023. Available from:https://arxiv.org/abs/2301.12597.

29. ^Alayrac JB, Donahue J, Luc P, Miech A, Barr I, Hasson Y, et al. Flamingo: a visual language model for few-shot

learning. In: NeurIPS. 2022. p. 23716–23736.

30. ^*Awadalla A, Gao I, Gardner J, Hessel J, Hanafy Y, Zhu W, Marathe K, Bitton Y, Gadre S, Sagawa S, Jitsev J, Kornblith S, Koh PW, Ilharco G, Wortsman M, Schmidt L. "OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models". arXiv [cs.CV]. 2023. Available from: https://arxiv.org/abs/2308.01390.*

31. ^*Li B, Zhang Y, Chen L, Wang J, Pu F, Yang J, Li C, Liu Z (2023). "MIMIC-IT: Multi-Modal In-Context Instruction Tuning". arXiv. Available from: https://arxiv.org/abs/2306.05425.*

32. ^*Zhang R, Han J, Liu C, Gao P, Zhou A, Hu X, Yan S, Lu P, Li H, Qiao Y (2024). "LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention". arXiv. arXiv:2303.16199 [cs.CV].*

33. ^*Dai W, Li J, Li D, Tiong AMH, Zhao J, Wang W, Li B, Fung P, Hoi S (2023). "InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning". arXiv. arXiv:2305.06500 [cs.CV].*

34. ^*Liu H, Li C, Wu Q, Lee YJ (2023). "Visual Instruction Tuning".arXiv. arXiv:2304.08485 [cs.CV].*

35. ^*Zhu D, Chen J, Shen X, Li X, Elhoseiny M (2023). "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models". arXiv. Available from: https://arxiv.org/abs/2304.10592.*

36. ^*Li K, He Y, Wang Y, Li Y, Wang W, Luo P, Wang Y, Wang L, Qiao Y (2024). "VideoChat: Chat-Centric Video Understanding". arXiv. arXiv:2305.06355.*

37. ^*Zhang H, Li X, Bing L (2023). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". arXiv. Available from: https://arxiv.org/abs/2306.02858.*

38. ^*Maaz M, Rasheed H, Khan S, Khan FS. "Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models". arXiv [cs.CV]. 2024. Available from: https://arxiv.org/abs/2306.05424.*

39. ^*Luo R, Zhao Z, Yang M, Dong J, Li D, Lu P, Wang T, Hu L, Qiu M, Wei Z (2023). "Valley: Video Assistant with Large Language Model Enhanced Ability". arXiv. arXiv:2306.07207 [cs.CV].*

40. ^*Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W (2021). "Lora: Low-rank adaptation of large language models". arXiv preprint arXiv:2106.09685. 2021.*

41. ^*Loshchilov I, Hutter F (2019). "Decoupled Weight Decay Regularization".arXiv. arXiv:1711.05101 [cs.LG].*

42. a, b*Jang J, Park J, Kim J, Kwon H, Sohn K (2023). "Knowing Where to Focus: Event-aware Transformer for Video Grounding". arXiv. arXiv:2308.06947 [cs.CV].*

43. a, b, c*Moon W, Hyun S, Park S, Park D, Heo JP (2023). "Query-Dependent Video Representation for Moment Retrieval and Highlight Detection". arXiv. arXiv:2303.13874 [cs.CV].*

44. a, b, c*Shen Y, Xiong X, Nagrani A, Arnab A, Wang Z, Ge W, Ross D, Schmid C (2023). "UnLoc: A Unified Framework for Video Localization Tasks". arXiv. arXiv:2308.11062 [cs.CV].*

45. ^*Yu S, Cho J, Yadav P, Bansal M (2023). "Self-Chained Image-Language Model for Video Localization and Question Answering". arXiv. arXiv:2305.06988 [cs.CV].*

46. a, b*Lin KQ, Zhang P, Chen J, Pramanick S, Gao D, Wang AJ, Yan R, Shou MZ (2023). "UniVTG: Towards Unified Video-Language Temporal Grounding". arXiv. arXiv:2307.16715 [cs.CV].*

47. a, b*Wang Y, Li K, Li X, Yu J, He Y, Wang C, Chen G, Pei B, Yan Z, Zheng R, Xu J, Wang Z, Shi Y, Jiang T, Li S,*

Zhang H, Huang Y, Qiao Y, Wang Y, Wang L (2024). "InternVideo2: Scaling Foundation Models for Multimodal Video Understanding". arXiv. arXiv:2403.15377 [cs.CV].

48. ^Krishna R, Hata K, Ren F, Fei-Fei L, Niebles JC (2017). "Dense-Captioning Events in Videos"arXiv. arXiv:1705.00754 [cs.CV].

49. ^Zeng R, Xu H, Huang W, Chen P, Tan M, Gan C (2020). "Dense Regression Network for Video Grounding".arXiv. arXiv:2004.03545 [cs.CV].

50. ^Soldan M, Xu M, Qu S, Tegner J, Ghanem B (2021). "VLG-Net: Video-Language Graph Matching Network for Video Grounding". arXiv. arXiv:2011.10132 [cs.CV].

51. ^Wang T, Zhang J, Zheng F, Jiang W, Cheng R, Luo P (2023). "Learning Grounded Vision-Language Representation for Versatile Understanding in Untrimmed Videos". arXiv. arXiv:2303.06378 [cs.CV].