ChatVTG: Video Temporal Grounding via Chat with Video Dialogue Large Language Models

Mengxue Qu[1,2*] Xiaodong Chen [3] Wu Liu [3] Alicia Li [4] Yao Zhao[1,2]

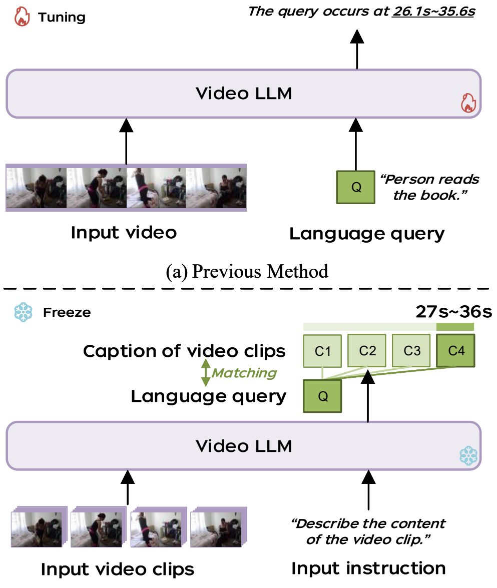
[1]Institute of Information Science, Beijing Jiaotong University
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology
[3]University of Science and Technology of China [4]Horace Mann School

Abstract

Video Temporal Grounding (VTG) aims to ground specific segments within an untrimmed video corresponding to the given natural language query. Existing VTG methods largely depend on supervised learning and extensive annotated data, which is labor-intensive and prone to human biases. To address these challenges, we present ChatVTG, a novel approach that utilizes Video Dialogue Large Language Models (LLMs) for zero-shot video temporal grounding. Our ChatVTG leverages Video Dialogue LLMs to generate multi-granularity segment captions and matches these captions with the given query for coarse temporal grounding, circumventing the need for paired annotation data. Furthermore, to obtain more precise temporal grounding results, we employ moment refinement for fine-grained caption proposals. Extensive experiments on three mainstream VTG datasets, including Charades-STA, ActivityNet-Captions, and TACoS, demonstrate the effectiveness of ChatVTG. Our ChatVTG surpasses the performance of current zero-shot methods.

1. Introduction

With the rapid development of short video platforms, videos have become the primary form of documenting and sharing life experiences for people worldwide. Unlike static image data, videos encapsulate dynamic information, providing a more vivid and immersive way to showcase stories and memories. Understanding the content of videos is crucial as it enables meaningful interactions based on the video context [1–3, 35]. Today's Video Understanding Large Language Models (LLMs) [15, 16, 22], have demonstrated remarkable capabilities in engaging in dialogues based on video content, enhancing the model's ability to explore and comprehend the information conveyed within videos.



(b) Our Method

Figure 1. Comparison between previous methods based on Video LLMs and our approach: Previous methods require fully supervised training of the Video LLM, whereas our method does not require training and can provide temporal grounding zero-shot.

As video content expands, efficiently retrieving specific segments from extensive video libraries remains with the rapid development of short video platforms, videos have become the primary form of documenting and sharing life experiences for people worldwide. Unlike static image data, videos encapsulate dynamic information, providing a more vivid and immersive way to showcase stories and memories. Understanding the content of videos is crucial as it enables meaningful interactions based on the video context. Today's Video Understanding Large Language Models (LLMs) [15, 16, 22], have demonstrated remarkable capabilities in engaging in dialogues based on video content, enhancing the model's ability to explore and comprehend

[*] Work done during an internship at JD Explore Academy.

the information conveyed within videos. crucial but challenging. The task of Video Temporal Grounding (VTG) has been developed to address this by aiming to accurately locate continuous time windows within videos using natural language descriptions, which is essential for enhancing the accessibility and usability of video content amidst improved yet still imperfect comprehension capabilities. Most existing VTG methods [8, 20, 34, 37] heavily rely on supervised training with paired annotation data, which requires extensive effort for precise time labeling. Additionally, the annotation process is further complicated by subjective biases in establishing precise temporal boundaries.

Compared to temporal grounding datasets, text-dialogue video data is more abundant and simpler to gather from the internet, with the potential for easier automated annotation. This data is often employed for tasks like video captioning and video question answering, where Video Language Models (LLMs) trained on it demonstrate notable conversational skills. Therefore, we contemplate *whether it's feasible to leverage Video Dialogue LLMs to zero-shot retrieve the corresponding video segments for a given query.*

In this paper, we introduce an innovative approach that leverages Video Dialogue LLMs for video temporal grounding without the need for additional training data or paired annotations. As shown in Fig. 1, different from existing methods that utilize VTG data to train Video LLMs for temporal grounding capabilities, our approach combines captions and matching to find the moment without training the Video LLM. Specifically, we start by dividing the video into multiple coarse segments and guide the Video LLM to provide captions for each segment through multi-granularity instructions. Then, we match these segments with the query provided by the user to identify coarse moments. To further refine the temporal boundaries, we conduct moment refinement by generating additional moment proposals using a sliding window approach. We then generate captions for proposals that overlap significantly with the coarse moments and further match them with the query to determine the final moment. This approach allows for seamless integration with existing Video LLMs, eliminating the need for additional training or specialized datasets.

To validate the effectiveness of our proposed framework, we conduct experiments on three commonly used VTG datasets, including Charades-STA [6], ActivityNet-Captions [14], TACoS [27]. Our ChatVTG outperforms existing zero-shot methods.

2. Related Work

2.1. Video Temporal Grounding

Video Temporal Grounding (VTG), also known as Video Temporal Localization (VTL) or Video Moment Retrieval (VMR), aims to align or ground a natural language query (often a sentence or phrase) to the corresponding moment or segment within an untrimmed video. Based on the difference in experimental setups and types of supervision during training, we categorize the related studies into (a) fully supervised VTG, (b) weakly supervised VTG, (c) unsupervised VTG, and (d) zero-shot (a.k.a language-free) VTG.

Fully supervised VTG. For this approach, previous methods [17, 20, 34, 37] first extract visual features and textual features from pre-trained models [11, 26, 30]. After that, a multi-model model is trained on a large amount of labeled data, where each video is paired with a corresponding textual description. The model learns to predict the grounding results based on the video input. For instance, VDI [20] generate textual and visual features from the Contrastive Language-Image Pretraining (CLIP) model [26] and propose a generic visual-dynamic injection model for capturing the temporal video changes and aligning them with the corresponding phrases (e.g. verb). While fully-supervised models can achieve high performance on specific datasets, they are the most data-intensive approaches, as they require a large dataset with accurately labeled video-text pairs.

Weakly supervised VTG. To mitigate the fine-grained labeling problem of fully supervised VTG, weakly-supervised methods [10, 33, 39, 40] are proposed to learn the segment-text alignment without marking the precious starting and ending time. During the training stage, only the video-level video-sentence matching is used for supervision. To predict the precision temporal localization under this setting, some methods [9, 23] utilize the multi-instance learning (MIL) technique. In detail, these methods form positive and negative video-level video-language pairs from the original dataset. Then they amplify the matching scores for the correct pairs and diminish those of the incorrect ones. However, an untrimmed video contains multiple events and can not correspond exactly with the sentence describing a specific event at the video level. Therefore, some methods [10, 40] apply the contrastive proposal learning or cycle self-training framework to generate positive and negative proposal at the frame level to optimize the model training under weak supervision.

Unsupervised VTG. Compared with Weakly supervised VTG, the unsupervised methods [5, 18, 25, 32] further removes the paired video-level annotated sentences. Due to the lack of supervision information, these methods optimize the model with pseudo queries or semantic features of the entire quire collection. For example, PSVL [25] first generates the simplified pseudo-quire from the query set of unpaired annotated annotations and then simulates the multimodal model with the pseudo-quire and video segment from the temporal video proposal module. Unlike PSVL, DSCNet [18] propose a linguistic semantic mining module to extract the implicit semantic feature from the unpaired entire query set. These linguistic semantic features guide the composition of the video activities, filtering out the redundant backgrounds and grounding the corresponding events. To be noted, some studies [25, 32] treat these methods as the zero-shot task. However, these methods still have access to unpaired training data, for which we categorize them as unsupervised tasks.

Zero-shot VTG. With a strict zero-shot setting for real-world applications, zero-shot VTG [17], sometimes referred to as language-free VTG, involves completing the video segment grounding corresponding to the given queries without relying on the training dataset. Therefore, these models have no access to training data even unpaired data, and rely entirely on their pre-existing knowledge and generalization capabilities to perform the grounding task. In practical terms, a zero-shot VTG model should typically be pre-trained on a diverse and extensive dataset without the limitation of tasks that enable it to learn a wide array of visual and linguistic concepts. This pre-training allows the model to develop an understanding of how language can describe video content. Then, when presented with the special zero-shot VTG task, the model utilizes this multi-modal comprehension to make inferences about where the described events are likely to occur in a video, even though it has never seen those specific quires during training. For instance, LLM4VG [4] evaluates the performance of existing LLMs under the zero-shot setting. UniVTG [17] propose a multi-modal and multi-task learning pipeline to pre-training the unified large-scale multi-modal model with dozens of unlimited datasets. In the inference stage, UniVTG is applied for the video temporal grounding task without fine-tuning or training on specific datasets.

2.2. Multi-model Large Language Model

Multi-modal Large Language Models (MLLM) [15, 22], like other Large Language Models (LLM), are trained on diverse and extensive multi-model datasets. VideoLLaVa [16] perform visual reasoning capabilities on both images and videos simultaneously with the binding of unified visual representations to the language feature space. Video-ChatGPT [22] is a video conversation model designed for generating meaningful conversations about the given videos. This model combines the capabilities of LLMs with a pre-trained visual encoder adapted for spatio-temporal video representation. VideoChat2 [15] is a robust video MLLM baseline for multi-model reasoning task. This model is progressive multi-modal trained with diverse instruction-tuning data to achieve temporal understanding in dynamic video tasks. The multi-modal comprehension of these models allows them to capture complex patterns and perform a wide variety of tasks. For example, they could potentially describe the content of a given image or video in text and have conversations with the user.

LLaViLo [21] exploits the capabilities of these MLLM in video understanding and designs a specialized adapter for the VTG task. With the joint training of a set prediction objective and a captioning objective, LLaViLo achieves significant performance improvement on the fully supervised VTG. VTimeLLM [8] and PG-Video-LLaVA [24] leverage the powerful conversational capabilities of Vicuna [38] and LLaVA [19] to treat VTG tasks as Q&A tasks, achieving robust fully-supervised performance with fine-tuning. However, the reliance on labeled VTG datasets reduces these models' ability to generalize to real-world scenarios that differ from the training data. To overcome these shortcomings, this paper attempts to accomplish the zero-shot VTG tasks utilizing the video comprehension capabilities of pre-trained MLLMs.

3. Method

In this section, we commence by elucidating the task definition of Video Temporal Grounding (VTG) and provide an overview of existing generic frameworks utilized for Video Large Multimodal Models (LMMs). Following this, we introduce our novel approach, which capitalizes on Video LMMs without the need for training, and proceed to expound upon its details.

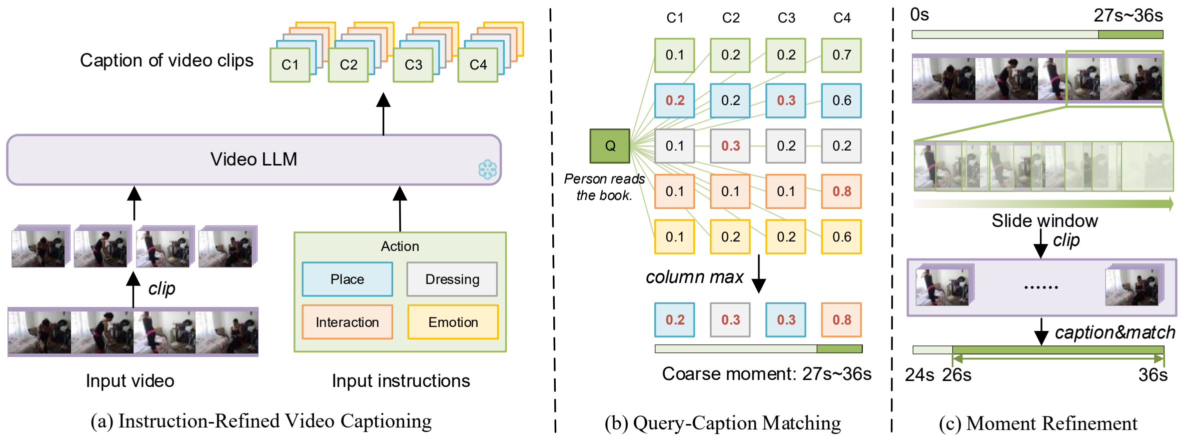
3.1. Task Definition

The goal of the VTG task is to accurately pinpoint video segments within untrimmed videos that correspond to the description provided by a given query description. This can be formally expressed as:

$$[ts, te] = F(V, Q), \tag{1}$$

where ts and te represent the start and end timestamps of the located video segment, respectively. Here, $F(\cdot)$ denotes the VTG model, V denotes the input untrimmed video, and Q represents the query description.

In this study, our objective is to directly utilize Video LMMs to achieve temporal grounding and query matching for the VTG task. These Video LMMs are trained solely on natural dialogues, enabling proficient performance in tasks such as video captioning and video question answering, contributing to a comprehensive understanding of video content. However, they lack training in temporal comprehension, thus unable to directly output moments through dialogue. This setup is defined as zero-shot, wherein the model is not trained on specific VTG datasets yet is expected to perform VTG tasks. In previous zero-shot approaches, some level of training was still required [25, 32]. In contrast, our approach adopts a completely offline, training-free method, obtaining temporal outputs through dialogue with Video LMMs.



(a) Instruction-Refined Video Captioning

(b) Query-Caption Matching

(c) Moment Refinement

Figure 2. Pipeline of ChatVTG. Our method pipeline primarily consists of three components: (a) Instruction-Refined Video Captioning; (b) Query-Caption Matching; and (c) Moment Refinement. Best viewed in color.

3.2. Review of Video LMMs

With the advancement of LMMs, numerous works have focused on training to establish Video Instruction datasets, enabling LMMs to comprehend video content and engage in dialogue based on it. In this section, we briefly review the common network architectures and training pipeline of Video LMMs.

Model Architecture The model architecture of Video LMMs typically involves several components. Initially, visual encoders extract features from raw visual signals. These features are then linearly mapped to the textual feature space, creating a unified visual representation. Subsequently, shared projection layers encode this representation before integration with textual queries, which are then inputted into a Large Language Model (LLM) to generate responses. Alignment between the video and language spaces converges into a unified visual feature space, bridging the gap between different visual signals and preparing them for input into the large language model to generate dialogue.

Training Pipeline The training pipeline of Video LMMs consists of two main stages: understanding training and instruction tuning. During the basic understanding training phase, the model is trained to interpret visual signals using an extensive dataset of image/video-text pairs. Each visual signal corresponds to a single round of conversation data (Xq, Xa), where Xq represents the input query and Xa is the ground truth response. The training objective involves the original auto-regressive loss, enabling the model to learn the basic ability to interpret visual information. Parameters unrelated to vision interpretation are frozen during this process. In the instruction tuning phase, the model is tasked with providing responses based on different instructions, which typically involve more complex visual comprehension tasks beyond simply describing visual signals.

Upon reviewing Video LLMs, it becomes apparent that traditional methods for completing the VTG task require training Video LLMs with paired *video-query-moment* data, a process attempted by some prior studies [8, 24]. However, this approach is characterized by its time-consuming and labor-intensive nature. In this paper, we propose a straightforward approach to address the VTG task, utilizing existing Video Dialogue Large Models. Our strategy involves generating captions and subsequently matching them with queries, aiming to efficiently achieve VTG.

3.3. ChatVTG

As illustrated in Fig. 2, our method pipeline primarily consists of three components: (a) Instruction-Refined Video Captioning; (b) Query-Caption Matching; and (c) Moment Refinement. The specific details of each component are elaborated below.

3.3.1 Instruction-Refined Video Captioning

As shown in Fig. 2 (a), given the untrimmed video for query, to facilitate effective temporal grounding, we initiate the process by segmenting the video V into m equal-length video clips, thus obtaining segments that are more manageable for analysis,

$$V \rightarrow \{S_1, S_2, ..., S_m\} \tag{2}$$

Where S_i denotes the ith video clip, and each clip has a time duration T. Subsequently, we feed each segmented video clip S_i into the Video LLM, leveraging its capabilities to comprehend and interpret visual content. By providing the Video LLM with a specific instruction tailored to the task, we guide the model to generate captions for each video segment,

$$\mathrm{C}_i = \text{Video LLM}(S_i, I) \tag{3}$$

Where C_i represents the generated caption for the i-th video clip, I represents the specific instruction tailored to the task provided to the Video LLM.

Due to the diverse nature of queries in the VTG task, which may describe events from various perspectives, we have established multi-granularity instructions to ensure a comprehensive understanding of video content and provide detailed descriptions. This aims to avoid mismatches between queries and captions resulting from overlooking important details. These instructions cover different aspects of the video, facilitating more nuanced guidance to ensure that the generated captions align with the video content. Specifically, we include the following five multi-granularity instructions, including "action", "place", "dressing". "emotion" and "interaction":

$$\begin{aligned} I_{\text{action}} &= \text{"Describe the action of the person in the video."} \\ I_{\text{place}} &= \text{"Where does this video take place?"} \\ I_{\text{dressing}} &= \text{"What are the people in the video wearing?"} \\ I_{\text{emotion}} &= \text{"illustrate the person's emotion or facial expression."} \\ I_{\text{interact}} &= \text{"Describe the interaction of person and other people or things."} \end{aligned} \tag{4}$$

Through these multi-granularity instructions, we can achieve a more comprehensive understanding of various aspects of the video content, thereby providing more accurate and detailed descriptions for video caption generation, The process can be represented as:

$$I_{mult} = \{I_{\text{action}}, I_{\text{place}}, I_{\text{dressing}}, I_{\text{expression}}, I_{\text{interaction}}\} \tag{5}$$

$$\begin{aligned} C_i &= \text{VideoLLM}(S_i, I_{mult}) \\ &= \{C_{i,\text{action}}, C_{i,\text{place}}, C_{i,\text{dressing}}, C_{i,\text{expression}}, C_{i,\text{interaction}}\} \end{aligned} \tag{6}$$

This formulation encapsulates the essence of our approach, wherein the Video LLM is employed to generate descriptive captions for each segmented video clip based on the provided instruction.

3.3.2 Query-Caption Matching

After obtaining multi-dimensional captions for each video clip, we proceed to match these captions with the query. Specifically, we employ SentenceBERT [28] to encode both the captions and the query,

$$f_Q = SentenceBERT(Q), \tag{7}$$

$$f_{C_i} = SentenceBERT(C_i) \tag{8}$$

We calculate their cosine similarity,

$$\text{Cosine Similarity}(f_{C_i}, f_Q) = \frac{f_{C_i} \cdot f_Q}{\|f_{C_i}\| \|f_Q\|} \tag{9}$$

where f_{C_i} represents the caption feature for the i^{th} video clip, and f_Q represents the query feature. Then we get a cosine similarity score matrix. Let S denote the similarity matrix, where rows represent different granularity captions, and columns represent different video clips. Each entry in the matrix represents the cosine similarity score calculated between the caption and query for a specific video clip. The similarity matrix S can be represented as:

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1m} \\ s_{21} & s_{22} & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nm} \end{bmatrix} \tag{10}$$

where n is the number of different granularity captions and m is the number of video clips. Each s_{ij} represents the cosine similarity score between the caption generated from the i^{th} granularity instruction and the query for the j^{th} video clip.

To ensure fair comparison across different granularity levels and comprehensively consider the maximum potential similarity between each granularity caption and the query, we employ Column-wise Maximum Normalization. This method selects the maximum similarity score $s_{\text{max}}^{(j)}$ for each column j (representing a video clip) across all granularity levels from the similarity matrix S, denoted as $S_{\cdot j}$, and then normalizes these scores. By doing so, it accounts for the highest potential match between each video clip and the query while allowing for fair comparison among different granularity levels. This normalization process scales the similarity scores proportionally, ensuring that the maximum similarity score for each column becomes 1.0, while preserving the relative relationships among the scores within each column. This process is illustrated in Fig. 2 (b).

After obtaining the normalized scores, we need to select the best clip or combinations of clips with closely related scores to determine the final moment window. This is achieved by setting a threshold based on the cosine similarity scores of the video clips. We select the longest consecutive combination of clips with scores exceeding the threshold as the final moment window.

3.3.3 Moment Refinement

Given the segmentation of the video into non-overlapping m clips in the previous step, it's essential to acknowledge that the initial moment boundaries might lack the precision needed for accurate temporal grounding. To address this, we adopt a strategy to refine these boundaries and enhance the fidelity of the moment detection process as in Fig. 2 (c).

Firstly, we segment the video using a sliding window approach into smaller segments. For a video of length L, we use "wide" to represent the window size and "step" to represent the step size, where $wide < L$ and $step < wide$. The number of windows is denoted by $k = \left\lfloor \frac{L-\text{wide}}{\text{step}} \right\rfloor + 1$. The start time of the ith window is $t_i = (i-1) \times$ step, and the end time is $t_i +$ wide. Therefore, the ith window corresponds to the video segment S_i, where $i = 1, 2, ..., k$.

Subsequently, we select the window combinations that have an intersection over union (IoU) greater than 0.7 with the coarse moment proposals obtained initially. This significant overlap between the selected windows and the proposed segments indicates their importance in capturing essential content from the video. Next, we input these selected window combinations into the VideoLLM for caption generation as in Eq. (3). This approach allows us to capture important details.

Query-caption matching proposed in Sec. 3.3.2 is then performed again to determine the final moment. In summary, with moment refinement, we are able to obtain more precise moment boundaries.

4. Experiments

In this section, we first introduce benchmark datasets, evaluation metrics, and implementation. After that, we compare the quantitative results of our ChatVTG with existing methods. At last, we provide ablation studies of each module.

4.1. Datasets

To evaluate the effectiveness of our approach, we perform experiments on three widely used datasets: Charades-STA, ActivityNet Captions, and TACoS. Our method stands out as it operates without any training, refraining from the use of training data during the experiments, with results being directly derived and calculated on the test sets.

Charades-STA [6] The work presented by Gao et al. [6] builds upon the Charades dataset [31] for action recognition and localization. They tailored the Charades dataset for Video Moment Retrieval (VMR) by incorporating query annotations. The resulting Charades-STA dataset comprises 6,670 videos with 16,124 associated queries. The average duration of the videos is 30.59 seconds, while the moments have an average duration of 8.09 seconds. Among the 16,124 queries, there are 37 long moments ($L_{\text{moment}}/L_{\text{vid}} \geq 0.5$) included in this dataset.

ActivityNet-Captions [14] is compiled specifically for the task of video captioning, sourced from ActivityNet [7], where videos are annotated with 200 distinct activity classes. This dataset, named ActivityNet-Captions, comprises 19,811 videos accompanied by 71,957 queries. On average, the videos span approximately 117.75 seconds, with individual moments lasting around 37.14 seconds. Within this dataset, there are 15,736 instances of long moments out of the total 71,957 queries.

TACoS [27] comprises 127 videos sourced from MPI-ICooking [29]. It encompasses a total of 18,818 video-text pairs depicting cooking activities in the kitchen, meticulously annotated by Regneri et al. [27].

4.2. Evaluation Metric

To assess the effectiveness of our model, we employ well-established metrics, including R@tIoU (Recall at temporal Intersection over Union) and mIoU (mean Intersection over Union), as delineated in previous research [20, 34, 37], to ensure a comprehensive and fair evaluation. Specifically, we measure temporal alignment accuracy by calculating the temporal Intersection over Union (tIoU) between predicted and ground-truth boundaries. R@tIoU quantifies the proportion of predictions exceeding predefined thresholds, namely $\{0.3, 0.5, 0.7\}$. Conversely, mIoU offers a consolidated evaluation by averaging the IoU scores across all predictions, providing a holistic perspective of model performance across various thresholds.

4.3. Implementation Details

In the Instruction-Refined Video Captioning stage, we primarily utilize VideoChatGPT [22] as the Video LLM, and conduct ablation experiments on VideoChat2 [15] as well. We adhere to the acceptable input frame numbers for each Video LLM during the video retrieval process. For the Query-caption Matching stage, Sentence-BERT [] is employed as the text feature extraction model, and cosine similarity calculation is used to evaluate the correlation between queries and captions. The initial number of segments is set to 5, and the refinement sliding window has a width of 10 with a step size of 5. These settings are thoroughly compared and explained in the ablation experiments. Our experiments don't involve any training processes, and inference is the only requirement. All experiments are conducted on an NVIDIA A100 GPU.

4.4. Comparison with Existing Methods

We compare our ChatVTG with previous methods across fully supervised, weakly supervised, unsupervised, and zero-shot settings. The performance on Charades-STA and ActivityNet-Captions is reported in Tab. 1. On Charades-STA, our ChatVTG significantly outperforms previous zero-shot methods with an improvement of 8.6% R@0.3 (44.09%→52.69%). For ActivityNet-Captions, the performance approaches the previous unsupervised methods, indicating the robustness of our ChatVTG. Additionally, we report the results on TACoS in Tab. 2. The results indicate that our method outperforms zero-shot UniVTG.

4.5. Ablation Studies

To showcase the individual efficacy of every module within our ChatVTG framework, we conduct comprehensive ablation studies on the Charades-STA dataset.

Impact of Different Instructions. We first investigate the impact of using different instructions. As shown in Tab. 3, the performance comparison from captions generated under different instructions yields relatively similar outcomes. This suggests that queries cover a range of perspectives, with each query potentially focusing on different aspects. However, a notable deviation is observed in the "Dress-

Method	Year	Setup	Charades-STA				ActivityNet-Captions			
			R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R%0.7	mIoU
2D-TAN [37]	ICCV’20	FS	57.31	45.75	27.88	41.05	60.43	43.41	25.04	42.45
MMN [34]	AAAI’22		65.43	53.25	31.52	46.46	64.48	48.24	29.35	46.61
VDI [20]	CVPR’23		-	52.32	31.37	-	-	48.09	28.76	-
UniVTG [17]	ICCV’23		72.63	60.19	38.55	52.17	-	-	-	-
VCA [33]	MM’21	WS	58.58	38.13	19.57	38.49	50.45	31.00	-	33.15
CNM [39]	AAAI’22		60.04	35.15	14.95	38.11	55.68	33.33	13.29	37.55
CPL [40]	CVPR’22		65.99	49.05	22.61	43.23	55.73	31.37	13.68	36.65
Huang et al. [10]	CVPR’23		69.16	52.18	23.94	45.20	58.07	36.91	-	41.02
PSVL [25]	ICCV’21	US	46.47	31.29	14.17	31.24	44.74	30.08	14.74	29.62
Gao et al. [5]	TCSVT’21		46.69	20.14	8.27	-	46.15	26.38	11.64	-
DSCNet [18]	AAAI’22		44.15	28.73	14.67	-	47.29	28.16	-	-
PZVMR [32]	MM’22		46.83	33.21	18.51	32.62	45.73	31.26	17.84	30.35
Kim et al. [13]	WACV’23		52.95	37.24	19.33	36.05	47.61	32.59	15.42	31.85
LLM4VG [4]	ArXiv’23	ZS	25.97	10.91	3.47	-	-	-	-	-
UniVTG* [17]	ICCV’23		44.09	25.22	10.03	27.12	-	-	-	-
Ours	-		**52.69**	**33.01**	**15.89**	**34.87**	**40.67**	**22.49**	**9.42**	**27.21**

Table 1. Performance comparison of methods under different setups on the test set of Charades-STA [6] and ActivityNet-Captions [14]. FS denotes fully supervised, WS denotes weakly supervised, US denotes unsupervised, and ZS denotes zero-shot. UniVTG* represents the results reported in UniVTG [17] under the zero-shot setting.

Method	Setup	R@0.3	R@0.5	R@0.7	mIoU
2D TAN [37]	FS	40.01	27.99	12.92	27.22
VSLNet [36]		35.54	23.54	13.15	24.99
MDETR [12]		37.97	24.67	11.97	25.49
UniVTG [17]		56.11	43.44	24.27	38.63
UniVTG* [17]	ZS	5.17	1.27	0.27	4.40
Ours		**8.12**	**3.74**	**1.25**	**5.54**

Table 2. Performance comparison of methods under different setups on the test set of TACoS [27]. FS denotes fully supervised and ZS denotes zero-shot. UniVTG* represents the results reported in UniVTG [17] under the zero-shot setting.

Instruction	mIoU	R@0.3	R@0.5	R@0.7
Action	32.73	51.05	29.54	13.90
Place	32.46	50.48	27.37	13.60
Dressing	27.38	42.83	21.82	9.10
Emotion	31.71	48.47	28.66	14.49
Interaction	31.15	48.22	27.13	13.61

Table 3. Experimental results using different instructions.

ing” instruction, indicating that certain scenarios may heavily or entirely lack key descriptions related to attire. Depending solely on dressing may not provide sufficient accuracy in event timing matching. Conversely, instructions such as “Action,” “Place,” “Emotion,” and “Interaction” are frequently utilized to differentiate and describe specific segments more effectively.

Different Matching Computation. In our exploration of effectively leveraging multi-granularity captions to match queries and obtain the final moment accurately, we have experimented with various computation methods for merging similarity scores of multi-granularity captions. Reference to Eq. (10), computation methods in Tab. 4 are as follows:

(1) Baseline: The baseline method involves calculating the cosine similarity score between the captions generated solely from the “action” granularity and the query.

(2) Normalization After Summation: This method adds up the cosine similarity score of all granularities before normalizing. With the incorporated information from multiple granularities, it makes a more comprehensive understanding of the video content. However, potential redundancy or conflicting information between different granularities may lead to performance fallback in certain scenarios.

(3) Summation after Normalization: This method normalizes the cosine similarity score of all granularities before adding up. Therefore, each granularity contributes equally to the overall similarity score, regardless of its inherent scale. However, similar to method (2), it may be sensitive to outliers or uneven distributions within the data.

(4) Normalization after Row-wise Maximum: This method identifies the granularity with the highest correlation with the given query and considers its score as the overall caption score. However, this approach overlooks the correlations between the given query and other granularity captions and may lead to biased results.

(5) Normalization after Column-wise Maximum: This method selects the maximum similarity score for each column (video clip) and then normalizes these scores. It comprehensively considers the maximum potential similarity between each granularity caption and the query while allowing for a fair comparison across different granularity levels. It effectively captures the diverse perspectives provided by each granularity level and results in improved performance, as adopted in our final implementation.

In summary, given the rationale and performance of the “Normalization after Column-wise Maximum” approach, we opt to utilize this method for matching computations.

Impact of Clip Number. To understand the impact of the

Fusion method	mIoU	R@0.3	R@0.5	R@0.7
(1)	32.73	51.05	29.54	13.90
(2)	34.17	51.71	31.62	14.85
(3)	34.43	52.45	31.67	15.32
(4)	33.30	51.48	31.75	15.24
(5)	**34.69**	**52.77**	**32.90**	**15.70**

Table 4. Experimental results of selecting the final moment through different matching computation methods.

Clip Num.	mIoU	R@0.3	R@0.5	R@0.7
3	32.28	49.56	29.78	12.55
5	**32.42**	**49.98**	**29.67**	**13.39**
10	30.96	47.79	28.47	12.41
20	28.19	42.50	25.11	11.10

Table 5. Experimental results of different clip numbers.

Slide window	mIoU	R@0.3	R@0.5	R@0.7
Baseline	32.73	51.05	29.54	13.90
(20, 5)	32.54	51.02	31.34	12.90
(10, 5)	34.09	52.69	34.19	14.92
(10, 2)	32.56	50.94	31.67	13.60
Coarse-to-Fine	**34.87**	**52.79**	**33.04**	**15.89**

Table 6. Experimental results of different slide window settings.

number of clips, we conducted ablation experiments on the coarsely segmented clips. We tested different numbers of clips, specifically 3, 5, 10, and 20. As shown in Tab. 5, the results indicate that the performance is relatively better when the number of clips is 5. Therefore, we have selected 5 as the initial number of coarsely segmented clips.

Impact of Slide Window Size. To achieve more precise moments, we partition the coarse-grained moments into smaller and denser time windows using a sliding window approach, followed by re-captioning the segments. The baseline performance represents the no sliding window used method, as shown in Tab. 6. We denote the window width and step size for the sliding window as (wide, step). The results show that employing a sliding window improves performance, especially when the window is set to (10, 5). However, adopting a sliding window for the entire duration increases computational complexity significantly. Additionally, it is not meaningful to refine segments with low matching scores. Therefore, we only select coarse-grained moments with an initial IoU greater than 0.7 for refinement, with a window setting of (10, 5). The results of this approach, as indicated as the "Coarse-to-Fine", show a performance improvement compared to the baseline.

Performance of Different Video LLMs Testing is also conducted on another Video LLM, VideoChat [15], which exhibited superior video understanding performance compared to VideoChatGPT. As shown in Tab. 7, an enhancement in the captioning capability of Video LLMs leads to an overall improvement in VTG performance.

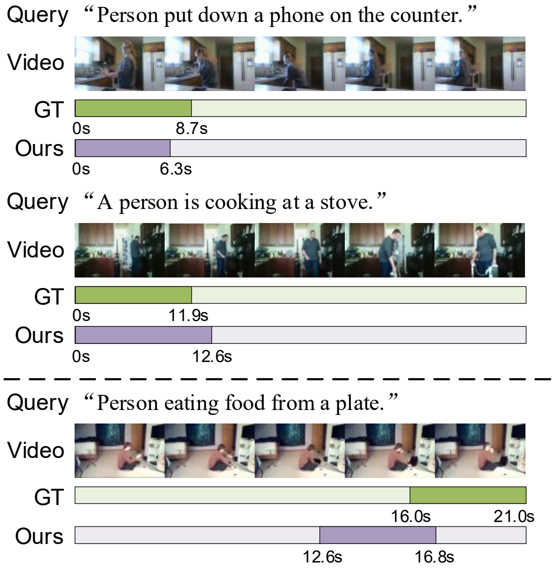

Query "Person put down a phone on the counter."
Video
GT
0s
8.7s
Ours
0s
6.3s
Query "A person is cooking at a stove."
Video
GT
0s
11.9s
Ours
0s
12.6s
Query "Person eating food from a plate."
Video
GT
16.0s
21.0s
Ours
12.6s
16.8s

Figure 3. Above the dashed line are successful examples on the Charades-STA dataset, and below are the failed ones.

Video LLM	R@0.3	R@0.5	R@0.7	mIoU
Video-ChatGPT	34.87	52.79	33.04	15.89
VideoChat2	36.55	56.56	34.52	16.34

Table 7. Experimental results of different Video LLMs

4.6. Qualitative Results

Fig. 3 displays selected prediction outcomes. Successful predictions from the Charades-STA dataset appear above the dashed line, while unsuccessful ones are presented below. Our observations indicate that predictions tend to be more precise for videos featuring simpler, prolonged actions. In contrast, accuracy diminishes in videos characterized by intricate action sequences and rapid alternation of multiple actions.

5. Conclusion and Future Work

We introduce ChatVTG in this paper, a novel and effective approach for zero-shot VTG that leverages the advanced capabilities of Video Dialogue LLMs. The success of ChatVTG signifies a leap forward in VTG tasks, offering a scalable and efficient solution for understanding and interacting with video content through natural language. In the future, our work can be applied to online video search, as its ability to enable the pre-processing of offline conversion from videos to textual descriptions. It can also be used for cross-video matching and retrieving, beyond the confines of individual videos.

Acknowledgements. This work was supported by the Fundamental Research Funds for the Central Universities (2023JBZD003), the National NSF of China (No.U23A20314), the National Key R&D Program of China under Grant No. 2021ZD0112100.

References

[1] Xiaodong Chen, Xinchen Liu, Wu Liu, Kun Liu, Dong Wu, Yongdong Zhang, and Tao Mei. Part-level action parsing via a pose-guided coarse-to-fine framework. In *ISCAS*, pages 419–423. IEEE, 2022. 1

[2] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2694–2703, 2023.

[3] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip H. S. Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, pages 20167–20177. IEEE, 2023. 1

[4] Wei Feng, Xin Wang, Hong Chen, Zeyang Zhang, Zihan Song, Yuwei Zhou, and Wenwu Zhu. LLM4VG: large language models evaluation for video grounding. *CoRR*, abs/2312.14206, 2023. 3, 7

[5] Junyu Gao and Changsheng Xu. Learning video moment retrieval without a single annotated video. *IEEE Trans. Circuits Syst. Video Technol.*, 32(3):1646–1657, 2022. 2, 7

[6] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *ICCV*, pages 5277–5285. IEEE Computer Society, 2017. 2, 6, 7

[7] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970. IEEE Computer Society, 2015. 6

[8] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower LLM to grasp video moments. *CVPR*, 2024. 2, 3, 4

[9] Jiabo Huang, Yang Liu, Shaogang Gong, and Hailin Jin. Cross-sentence temporal and semantic relations in video activity localisation. In *ICCV*, pages 7179–7188. IEEE, 2021. 2

[10] Yifei Huang, Lijin Yang, and Yoichi Sato. Weakly supervised temporal sentence grounding with uncertainty-guided self-training. In *CVPR*, pages 18908–18918. IEEE, 2023. 2, 7

[11] Yuan Jing, Jinshan Hao, and Peng Li. Learning spatiotemporal features of CSI for indoor localization with dual-stream 3d convolutional neural networks. *IEEE Access*, 7:147571–147585, 2019. 2

[12] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1760–1770. IEEE, 2021. 7

[13] Dahye Kim, Jungin Park, Jiyoung Lee, Seongheon Park, and Kwanghoon Sohn. Language-free training for zero-shot video grounding. In *WACV*, pages 2538–2547. IEEE, 2023. 7

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pages 706–715. IEEE Computer Society, 2017. 2, 6, 7

[15] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. *CVPR*, 2024. 1, 3, 6, 8

[16] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *CoRR*, abs/2311.10122, 2023. 1, 3

[17] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *ICCV*, pages 2782–2792. IEEE, 2023. 2, 3, 7

[18] Daizong Liu, Xiaoye Qu, Yinzhen Wang, Xing Di, Kai Zou, Yu Cheng, Zichuan Xu, and Pan Zhou. Unsupervised temporal video grounding with deep semantic clustering. In *AAAI*, pages 1683–1691. AAAI Press, 2022. 2, 7

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 3

[20] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Towards generalisable video moment retrieval: Visual-dynamic injection to image-text pre-training. In *CVPR*, pages 23045–23055. IEEE, 2023. 2, 6, 7

[21] Kaijing Ma, Xianghao Zang, Zerun Feng, Han Fang, Chao Ban, Yuhan Wei, Zhongjiang He, Yongxiang Li, and Hao Sun. Llavilo: Boosting video moment retrieval via adapter-based multimodal modeling. In *ICCV (Workshops)*, pages 2790–2795. IEEE, 2023. 3

[22] Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *CoRR*, abs/2306.05424, 2023. 1, 3, 6

[23] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K. Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, pages 11592–11601. Computer Vision Foundation / IEEE, 2019. 2

[24] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Shahbaz Khan. Pg-video-llava: Pixel grounding large video-language models. *CoRR*, abs/2311.13435, 2023. 3, 4

[25] Jinwoo Nam, Daechul Ahn, Dongyeop Kang, Seong Jong Ha, and Jonghyun Choi. Zero-shot natural language video localization. In *ICCV*, pages 1450–1459. IEEE, 2021. 2, 3, 7

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2

[27] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Trans. Assoc. Comput. Linguistics*, 1:25–36, 2013. 2, 6, 7

[28] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In

EMNLP/IJCNLP (1), pages 3980–3990. Association for Computational Linguistics, 2019. 5

[29] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. Script data for attribute-based recognition of composite activities. In *ECCV (1)*, pages 144–157. Springer, 2012. 6

[30] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. 2

[31] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV (1)*, pages 510–526. Springer, 2016. 6

[32] Guolong Wang, Xun Wu, Zhaoyuan Liu, and Junchi Yan. Prompt-based zero-shot video moment retrieval. In *ACM Multimedia*, pages 413–421. ACM, 2022. 2, 3, 7

[33] Zheng Wang, Jingjing Chen, and Yu-Gang Jiang. Visual co-occurrence alignment learning for weakly-supervised video moment retrieval. In *ACM Multimedia*, pages 1459–1468. ACM, 2021. 2, 7

[34] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *AAAI*, pages 2613–2623. AAAI Press, 2022. 2, 6, 7

[35] Kaining Ying, Qing Zhong, Weian Mao, Zhenhua Wang, Hao Chen, Lin Yuanbo Wu, Yifan Liu, Chengxiang Fan, Yunzhi Zhuge, and Chunhua Shen. Ctvis: Consistent training for online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 899–908, 2023. 1

[36] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, pages 6543–6554. Association for Computational Linguistics, 2020. 7

[37] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, pages 12870–12877. AAAI Press, 2020. 2, 6, 7

[38] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 3

[39] Minghang Zheng, Yanjie Huang, Qingchao Chen, and Yang Liu. Weakly supervised video moment localization with contrastive negative sample mining. In *AAAI*, pages 3517–3525. AAAI Press, 2022. 2, 7

[40] Minghang Zheng, Yanjie Huang, Qingchao Chen, Yuxin Peng, and Yang Liu. Weakly supervised temporal sentence grounding with gaussian-based contrastive proposal learning. In *CVPR*, pages 15534–15543. IEEE, 2022. 2, 7