

VTG-LLM: Integrating Timestamp Knowledge into Video LLMs for Enhanced Video Temporal Grounding

Yongxin Guo^{1*}, Jingyu Liu², Mingda Li², Dingxin Cheng³, Xiaoying Tang^{1,4,5†}, Dianbo Sui⁶,
Qingbin Liu², Xi Chen^{2†}, Kevin Zhao²

¹ School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China

² Tencent PCG

³ Shandong University

⁴ The Shenzhen Future Network of Intelligence Institute, CUHK-Shenzhen, 518172, P.R. China

⁵ The Guangdong Provincial Key Laboratory of Future Networks of Intelligence, CUHK-Shenzhen, 518172, P.R. China

⁶ Harbin Institute of Technology

yongxinguo@link.cuhk.edu.cn, tangxiaoying@cuhk.edu.cn, jasonxchen@tencent.com

Abstract

Video Temporal Grounding (VTG) strives to accurately pinpoint event timestamps in a specific video using linguistic queries, significantly impacting downstream tasks like video browsing and editing. Unlike traditional task-specific models, Video Large Language Models (video LLMs) can handle multiple tasks concurrently in a zero-shot manner. Consequently, exploring the application of video LLMs for VTG tasks has become a burgeoning research area. However, despite considerable advancements in video content understanding, video LLMs often struggle to accurately pinpoint timestamps within videos, limiting their effectiveness in VTG tasks. To address this, we introduce VTG-LLM, a model designed to enhance video LLMs' timestamp localization abilities. Our approach includes: (1) effectively integrating timestamp knowledge into visual tokens; (2) incorporating absolute-time tokens to manage timestamp knowledge without concept shifts; and (3) introducing a lightweight, high-performance, slot-based token compression technique designed to accommodate the demands of a large number of frames to be sampled for VTG tasks. Additionally, we present VTG-IT-120K, a collection of publicly available VTG datasets that we have re-annotated to improve upon low-quality annotations. Our comprehensive experiments demonstrate the superior performance of VTG-LLM in comparison to other video LLM methods across a variety of VTG tasks.

Code — <https://github.com/gyxyg/VTG-LLM>

Introduction

Video Temporal Grounding (VTG) is a crucial component of video understanding. It requires models to accurately pinpoint event timestamps within a video based on the given query. This task is essential for subsequent operations such as video browsing and editing. In this paper, we adopt the categorization described by Lin et al. (2023b), and divide

*This work was done when Yongxin Guo was an intern at Tencent PCG.

†Corresponding authors.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the VTG task into four primary sub-tasks. For instance, moment retrieval and dense video captioning tasks (Fabian Caba Heilbron and Niebles 2015; Zhou, Xu, and Corso 2018) require models to generate one or multiple timestamp intervals, each accompanied by captions. The video highlight detection task (Lei, Berg, and Bansal 2021) necessitates models to produce a prominent score curve, while the video summarization task (Song et al. 2015) demands that models output a series of timestamps, each associated with their corresponding video frames.

Despite significant efforts by traditional task-specific models designed for various VTG tasks (Yang et al. 2023; Luo et al. 2023a; Lei, Berg, and Bansal 2021), these methods are limited in their ability to (1) handle multiple VTG tasks simultaneously and (2) provide zero-shot capabilities on VTG tasks, which are crucial for real-world applications. As a remedy, recent research has begun exploring the use of video LLMs (Lin et al. 2023a; Zhang, Li, and Bing 2023; Li et al. 2023a) as generalists (Ren et al. 2023; Huang et al. 2023) for addressing VTG tasks due to their capacity to handle various tasks concurrently in a zero-shot manner. However, several persistent issues hinder the effectiveness of current video LLMs in understanding timestamp knowledge, which in turn affects their performance on VTG tasks:

- Visual inputs should contain sufficient and accurate timestamp information to help models understand when the visual content occurs in the videos.
- Concept shifts occur when varying input data produce identical output targets, potentially obscuring decision boundaries (Moreno-Torres et al. 2012). This issue arises when using shared token embedding and classification heads for all digit-related knowledge. For example, the number '20' can appear in both a counting context, such as 'There are 20 people,' and a temporal context like 'From 20-30 seconds.' Although these digits have different meanings in these situations, they are forced to share the same decision boundary, making classification more challenging.
- VTG tasks necessitate sampling more frames compared

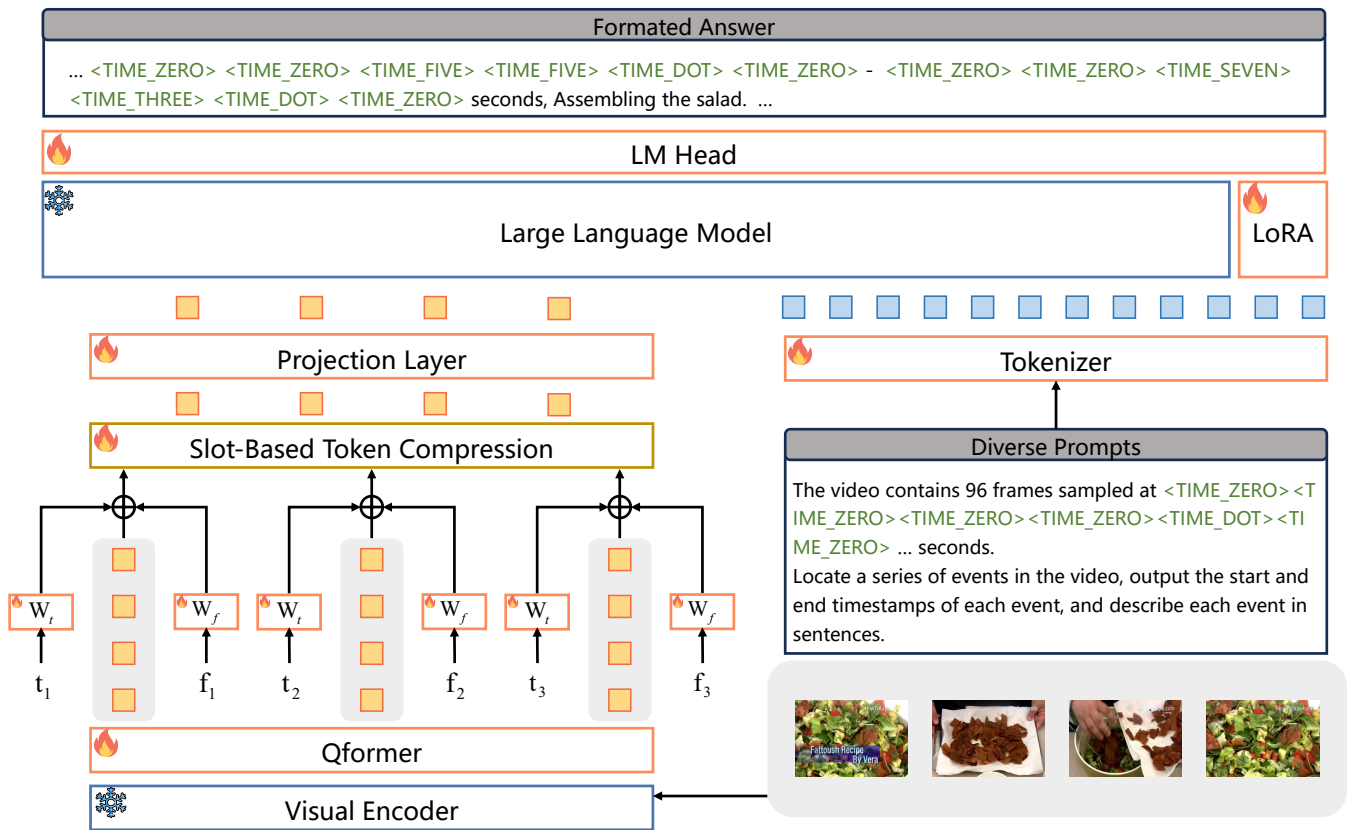


Figure 1: Overview of the VTG-LLM model.

to other video tasks, i.e., VQA tasks typically require only 8 or 16 frames (Lin et al. 2023a; Cheng et al. 2024). However, models cannot make reliable predictions on visual content that is not included in the sampled frames fed into language models. Therefore, due to the limited context length of LLMs, it is essential to develop effective token compression methods that enable the sampling of more frames.

To address these issues, in this paper, we present a novel video LLM model, VTG-LLM, comprising three components that efficiently integrate timestamp knowledge into video LLMs, thereby enhancing their performance on VTG tasks: (1) we introduce a sequence-time embedding method that integrates accurate timestamp data into visual tokens; (2) we incorporate absolute-time tokens that specifically handle timestamp knowledge without introducing quantization errors; (3) we employ a lightweight and high-performance slot-based token compression method to decrease the number of visual tokens to a fixed length, allowing video LLMs to sample more video frames.

In addition to VTG-LLM, we also discovered that existing datasets either suffer from low quality (Zellers et al. 2021) or exhibit significant task imbalance (Ren et al. 2023; Wang et al. 2024a). To address this issue, we introduce VTG-IT-120K, a dataset comprising 47.2K videos and 120K annotations. We collected this dataset from publicly avail-

able sources and re-annotated the low-quality captions using Gemini-1.5 Pro. VTG-IT-120K covers most mainstream VTG tasks, including moment retrieval (63.2K), dense video captioning (37.2K), video summarization (15.2K), and video highlight detection (3.9K). This dataset exhibits a more balanced task distribution compared to existing datasets (Ren et al. 2023; Wang et al. 2024a).

Numerical results illustrate the superior performance of the VTG-LLM compared to state-of-the-art video LLM methods. Additionally, ablation studies reveal that absolute-time tokens and sequence-time embedding significantly enhance the model’s performance in accurately locating timestamps. Furthermore, the slot-based token compression method outperforms the naive token compression baselines such as sampling and cross-attention in our studied cases. Our key contributions are summarized as follows:

- We present VTG-LLM, a versatile model aims to capture all four VTG tasks. By effectively incorporating timestamp information into video LLMs, VTG-LLM equips these models with the capability to comprehend and process timestamps. Moreover, we have proposed carefully crafted initialization strategies for VTG-LLM to take advantage of pretrained video LLM weights.
- We present VTG-IT-120K, a high-quality instruction tuning dataset consisting of 51.9K re-annotated data annotations with superior quality.

- Quantitative results demonstrate the superior performance of VTG-LLM across various datasets, such as Charades-STA, QVHighlights, and YouCook2. Our code and datasets are provided in the Supplementary Material.

Related Works

Video Temporal Grounding (VTG) aims to accurately locate the timestamps of events within a given video (Lin et al. 2023b). This encompasses tasks such as moment retrieval (Gao et al. 2017; Zala et al. 2023b; Oncescu et al. 2021; Wang et al. 2024b), dense video captioning (Zellers et al. 2021; Zala et al. 2023b; Tang et al. 2019; Fabian Caba Heilbron and Niebles 2015), video summarization (Song et al. 2015; Gygli et al. 2014), and video highlight detection (Lei, Berg, and Bansal 2021; Xiao et al. 2023). Traditional methods mainly address VTG tasks through large-scale video-text pre-training, using training objectives such as video-text contrastive learning (Xu et al. 2021; Wang et al. 2022), video-text matching (Li et al. 2023b; Chen et al. 2024), and masked auto-encoding (Tong et al. 2022; Zhao et al. 2024). Although these methods have shown satisfactory results, they require resource-consuming pre-training, lack zero-shot capabilities, and often need further fine-tuning on many downstream tasks.

Large language models (LLMs) (Achiam et al. 2023; Touvron et al. 2023) have exhibited considerable potential in capturing knowledge and tackling real-world challenges using a zero-shot approach. Recently, research has explored integrating knowledge from other modalities, such as vision (Liu et al. 2024) and audio (Ghosal et al. 2023), to enhance the capabilities of LLMs. Within the visual modality, video large language models (video LLMs) have emerged as a significant research area (Lin et al. 2023a; Zhang, Li, and Bing 2023; Li et al. 2023a; Song et al. 2024a,b). Traditional video LLMs primarily generate captions that summarize videos (Zhang, Li, and Bing 2023; Lin et al. 2023a; Li et al. 2023a; Maaz et al. 2023; Zhu et al. 2023), but they struggle to accurately pinpoint event timestamps within videos. Some studies have attempted to address this limitation, such as TimeChat (Ren et al. 2023), which constructs time-sensitive instruction tuning datasets and encodes timestamp knowledge into visual tokens. VTimeLLM (Huang et al. 2023) proposes a LLaVA-like three-stage training method, while LITA (Huang et al. 2024) introduces fast-slow visual tokens and adds time tokens to LLM tokenizers. Momentor (Qian et al. 2024) proposes a time encoder to solve time token quantization errors, and HawkEye (Wang et al. 2024a) constructs a high-quality instruction tuning dataset based on InternVid (Wang et al. 2022) for the moment retrieval task. NumPro (Wu et al. 2024) add frame numbers to frames for ease of temporal understanding. However, existing instruction tuning datasets for VTG tasks are often low-quality or exhibit extreme task imbalance, which hinders model performance across tasks. To address this issue, we propose a high-quality dataset, VTG-IT, and introduce VTG-LLM, a model comprising three well-designed components to enhance video LLM performance on VTG tasks.

Method

In this section, we introduce the VTG-LLM model and the VTG-IT-120K dataset. The overview of the VTG-LLM model can be found in Figure 1.

Overview of VTG-LLM

In this subsection, we present the detailed structure of VTG-LLM, which comprises three key components designed to enhance video LLMs’ understanding of timestamps. Specifically, we propose: (1) a sequence-time embedding mechanism that directly incorporates timestamp information into visual tokens; (2) the introduction of absolute-time tokens without quantization errors to differentiate time-related knowledge from other digit-related knowledge; and (3) the implementation of slot-based token compression to enable video LLMs to effectively process more frames. We will discuss these components in detail throughout this section.

Sequence-Time Embedding

Sequence embedding. Existing studies (Zhang, Li, and Bing 2023; Ren et al. 2023) employ sequence embedding to incorporate relative time information into visual tokens. Specially, given $N \times M$ tokens $\{\mathbf{z}_{i,j} | 1 \leq i \leq N, 1 \leq j \leq M\}$ sampled from N frames, the process of adding sequence embedding onto visual tokens can then be represented by:

$$\hat{\mathbf{z}}_{i,j} = \mathbf{z}_{i,j} + [\mathbf{W}_s]_i, \quad (1)$$

where $\mathbf{W}_s \in \mathbb{R}^{N \times d}$ is the weight of sequence-embedding matrix.

Sequence-Time Embedding. Although promising, the sequence embedding only contains information about the temporal order and may not accurately represent knowledge about the absolute timestamps. For instance, sampled frames may not be uniformly distributed throughout the entire video (Ren et al. 2023; Huang et al. 2023). Moreover, the sampling intervals can vary significantly for videos of different lengths. These issues make it difficult to infer the timestamps of the frames simply using sequence embedding. To address this issue, in addition to the sequence embedding, we also add the absolute time embedding as shown in the following equation

$$\hat{\mathbf{z}}_{i,j} = \mathbf{z}_{i,j} + [\mathbf{W}_s]_i + [\mathbf{W}_t]_t. \quad (2)$$

Here, t is the absolute timestamp (in seconds) of the corresponding frames. $\mathbf{W}_t \in \mathbb{R}^{T \times d}$ is the absolute time embedding. T is the maximum timestamp (in seconds). *It’s important to note that to avoid disrupting the visual tokens generated by pre-trained visual encoders and to speed up the convergence, \mathbf{W}_t is initialized by setting all entries to zero.* Since we employ pretrained vision encoder and Qformer modules, the performance of VTG-LLM is significantly degraded without the zero-initialization method. This is evident in the “TE Random Initialize” section of Table 2.

Due to the imbalanced nature of video lengths and the gap between training and test data, there may be timestamps for which the absolute time embedding has not been trained. To address this issue, we further introduce a test-time interpolation mechanism. In detail, for the timestamp t which is not

trained during training, we first find the timestamps t_l and t_r that satisfy:

$$t_l = \arg \max_{t_l} t_l < t, t_l \in \mathcal{T}_{tr}, t_r = \arg \min_{t_r} t_r > t, t_r \in \mathcal{T}_{tr}, \quad (3)$$

where \mathcal{T}_{tr} is the set of timestamps with trained absolute-time embedding. Then the absolute-time embedding of timestamp t is given by

$$[\mathbf{W}_t]_t = \frac{t - t_l}{(t_r - t_l)} [\mathbf{W}_t]_{t_l} + \frac{t_r - t}{(t_r - t_l)} [\mathbf{W}_t]_{t_r}. \quad (4)$$

Please refer to Appendix (Guo et al. 2024) for ablation studies on using test-time interpolation.

Discussion on existing techniques that integrating information of timestamps into visual tokens. In addition to incorporating time embeddings into visual tokens, there are other techniques for integrating timestamp information into visual tokens. For instance, adding text inputs into Qformer (Ren et al. 2023) and inserting time tokens before visual tokens (Hua et al. 2024). However, we believe that these techniques are orthogonal to the time embedding approach, and it is also possible to combine these approaches in future work.

Absolute-Time Tokens

The incorporation of unique time tokens in the tokenizer has shown benefits in video temporal grounding tasks, as evidenced by various studies (Yang et al. 2023; Qian et al. 2024; Huang et al. 2024). However, using relative time tokens (frame ID) exposes some limitations. First, quantization errors grow linearly with video length, complicating fine-grained predictions for longer videos. Second, training token embedding from scratch hinders leveraging pretrained video LLMs’ benefits. Our goal is to create a novel time token mechanism that accurately represents fine-grained timestamps and easily adapts to pretrained video LLMs.

Absolute-time tokens for resolving the quantization errors. To resolve the issue of quantization errors, we introduce the concept of *absolute-time tokens*. As depicted in Figure 1, we have incorporated eleven time tokens into the tokenizer. These consist of ten digit time tokens representing the digits from 0 to 9, and an additional token for a decimal point. All timestamps are represented using six time tokens. For instance, the time 120.5 seconds would be formatted as $\langle t_0 \rangle \langle t_1 \rangle \langle t_2 \rangle \langle t_0 \rangle \langle t_{dot} \rangle \langle t_5 \rangle$. This formulation allows the time tokens to handle videos more than 1 hours in length, with the precision remaining constant regardless of video length increases. Importantly, it is essential to format all timestamps using the same number of time tokens. As shown in the “Time Token not Formatted” section of Table 2, we found that maintaining a consistent format for all timestamps significantly improves model performance.

Initialization of token embedding for absolute-time tokens. While using absolute time tokens eliminates quantization errors, we found that randomly initialized time tokens adversely affect the original token embedding space, hindering LLMs from learning precise time token knowledge.

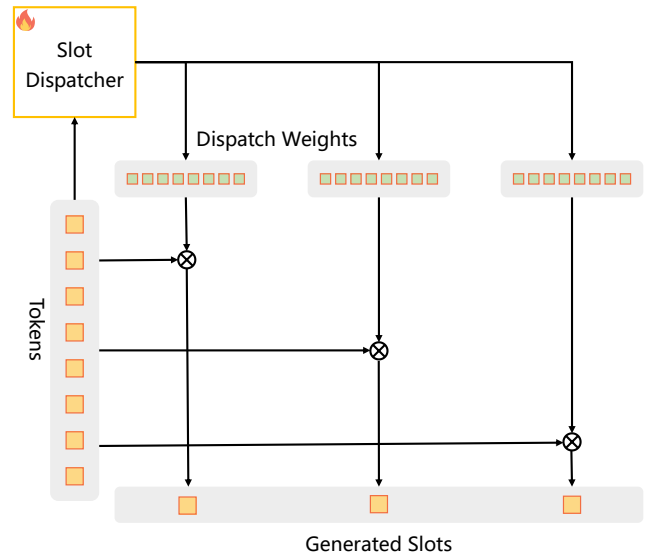


Figure 2: Overview of slot based token compression.

Consequently, this leads to subpar model performance, as demonstrated in the “Token Embedding not Initialized” section of Table 2. To tackle this issue, our goal is to initialize the time-related knowledge using the digit-related knowledge, as the digit-related knowledge has already been well-trained during the LLM pretraining stage.

This is achieved by adjusting the initialization of the weights of token embedding and LLM prediction head. For example, consider the token embedding matrix \mathbf{W}_{token} , when initializing the embedding for the time token “ $\langle t_1 \rangle$ ”, we align the embedding of this time token with the embedding of the token “1”:

$$[\mathbf{W}_{token}]_{ID(\langle t_1 \rangle), j} = [\mathbf{W}_{token}]_{ID('1'), j}, \forall 0 \leq j < d, \quad (5)$$

where $ID()$ denotes the token ID of the given token string. Similarly, for the LM prediction head, we employ the same method to transfer the knowledge from the digit tokens to the time tokens.

Slot-Based Token Compression

Compressing visual tokens is crucial in VTG tasks. On one hand, models cannot produce reliable predictions without adequate visual input. On the other hand, the context length of LLMs inherently imposes a limitation. To address this, we propose a straightforward yet efficient approach called slot-based token compression, which compresses visual tokens to a fixed number, thereby enabling models to sample more frames.

As illustrated in Figure 2, given N visual tokens $\mathbf{z}_1, \dots, \mathbf{z}_N$, and the trainable slot dispatcher $\Phi \in \mathbb{R}^{K \times d}$, where K represents the number of slots, the visual tokens first pass through the slot dispatcher to obtain the dispatch weights, and the tokens are mixed based on the dispatch weights to generate the slots. For instance, the slot k is gen-

erated by

$$\mathbf{s}_k = \sum_{i=1}^N \frac{\exp(\Phi_k^T \mathbf{z}_i)}{\sum_{j=1}^N \exp(\Phi_k^T \mathbf{z}_j)} \mathbf{z}_i. \quad (6)$$

The slots are subsequently fed into the visual projection layers and serve as visual tokens.

Discussion on slot mechanism The slot-based token compression is inspired by the slot mechanism in SoftMoE (Puigcerver et al. 2023). However, in SoftMoE, the slots are not used to reduce sequence length, which is distinct from our primary objective. Moreover, some existing multi-modal LLMs compress the token number through cross-attention (Zhang, Li, and Bing 2023; Bai et al. 2023; Ren et al. 2023). Our approach involves training only one matrix Φ , making it more computationally efficient and less data-consuming, and thereby achieve better performance (Table 2) on relatively small-scale instruction tuning datasets.

VTG-IT-120K: Formatted Time-Sensitive Instruction Tuning Dataset

In this section, we introduce VTG-IT-120K, a dataset comprising 120K publicly available video-text pairs, building on the TimeIT dataset (Ren et al. 2023). Additionally, we have re-annotated 51.9K low-quality video annotations using Gemini 1.5-Pro¹. A comparison between the original and new annotations can be seen in Figure 3. It is clear that the revised captions are more succinct and contain less information unrelated to the visual content. The VTG-IT-120K dataset encompasses four distinct video temporal grounding tasks.

- *Moment Retrieval (63.2K)*: For the moment retrieval task, we use HiREST_{grounding} (Zala et al. 2023a), QuerYD (Oncescu et al. 2021), DiDeMo (Hendricks et al. 2018), and VTG-IT-MR.
- *Dense Video Captioning (37.2K)*: For the dense video captioning task, we use HiREST_{step} (Zala et al. 2023a), COIN (Tang et al. 2019), ActivityNet Captions (Fabian Caba Heilbron and Niebles 2015), and VTG-IT-DVC.
- *Video Summarization (15.2K)*: For the video summarization task, we use TVSum (Song et al. 2015), SumMe (Gygli et al. 2014), and VTG-IT-VS.
- *Video Highlight Detection (3.9K)*: For the video highlight detection task, we use VTG-IT-VHD.

The VTG-IT-X annotations for the four tasks are re-annotated using 16K videos from the YT-Temporal-180M dataset (Zellers et al. 2021).

Data formatting. All tasks are structured as QA pairs, with varied questions designed to help models comprehend human intent. The answers are formatted to facilitate knowledge transfer between different tasks. Detailed examples are provided in the Appendix (Guo et al. 2024).

¹The details of the annotation process can be found in Appendix (Guo et al. 2024).

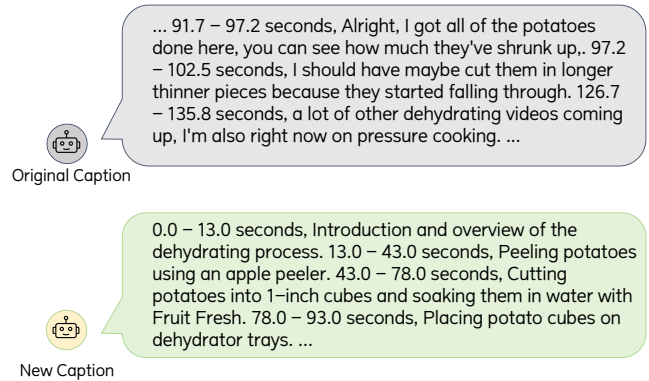


Figure 3: Example of new annotations.

Experiments

In the experimental section, we primarily aim to address the following questions²:

- Q1. Will VTG-LLM achieve better zero-shot performance compared to SOTA video LLMs?
- Q2. Is it necessary to use time embeddings, and what will happen if time embeddings are randomly initialized?
- Q3. How will the special time tokens affect the model's performance?
- Q4. Will the slot-based compression method outperform other compression methods?

Experiment Settings

Models and training configuration. Unless otherwise specified, we employ ViT-G/14 from EVA-CLIP (Sun et al. 2023) and the Qformer from InstructBLIP (Dai et al. 2024) as the visual backbone. The vision projection layer and the weights of sequence embedding are initialized using the Video-LLaMA checkpoint (Zhang, Li, and Bing 2023). For the language model backbone, we utilize LLaMA-2-7B (Touvron et al. 2023). The number of slots is set to 256. The models are trained on the VTG-IT-120k and a randomly sampled subset (97k) from the Valley dataset (Luo et al. 2023b). For the training process, we initially divide the videos uniformly into 96 segments, from each of which we then randomly select one frame. For the testing process, we uniformly select 96 frames from the entire video. Training is carried out using 16 ATN 910B NPUs with a batch size of 64. The fine-tuning also uses 16 NPUs and set the batch size to 16. We set the learning rate to 3e-5 initially, and train the models for 10 epochs by default.

²Fine-tuned performance, results on ActivityNet Captions, more ablation studies, qualitative comparisons, and case studies can be found in Appendix (Guo et al. 2024).

Model	Youcook2			Charades-STA		QVHighlights	
	SODA_c	CIDEr	F1 Score	R@1 _(IOU=0.5)	R@1 _(IOU=0.7)	mAP	HIT@1
Traditional Video LLMs							
Valley (7B)	0.1	0.0	1.5	4.7	1.6	10.9	15.2
VideoChat (7B)	0.2	0.6	3.4	3.2	1.4	13.1	18.1
Video-LLaMA (7B)	0.0	0.0	0.1	2.7	1.2	11.3	15.6
Video-ChatGPT (7B)				7.7	1.7	3.8	
Temporal Grounding Video LLMs							
TimeChat (7B)	1.2	3.4	12.6	32.2	13.4	14.5	23.9
VTimeLLM (7B)	1.0	3.6	9.1	27.5	11.4		
Momentor (7B)				26.6	11.6	7.6	
HawkEye (7B)				31.4	14.5		
VTG-LLM (7B)	1.5	5.0	17.5	33.8	15.7	16.5	33.5

Table 1: Zero-shot performance of algorithms over various tasks.

Evaluation datasets, metrics, and baseline models. We evaluate the model performance on three different tasks:

- *dense video captioning.* We employ Youcook2 (Zhou, Xu, and Corso 2018) as the test dataset, and following the evaluation settings in TimeChat (Ren et al. 2023). Metrics including CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), which assessing the quality of the captions; (2) SODA_c (Fujita et al. 2020) for story-level evaluation; (3) F1 score to measure the model’s ability to accurately locate timestamps;
- *Moment retrieval.* We utilize test set of Charades-STA (Gao et al. 2017) for the moment retrieval task and report the recall at IOU thresholds of 0.5 and 0.7.
- *Video highlight detection.* We employ the validation set of the QVHighlights dataset (Lei, Berg, and Bansal 2021) and report the mean average precision (mAP) with IOU thresholds of 0.5 and 0.75, as well as the HIT@1, which represents the hit ratio of the highest scored clip.

For baseline models, we select Valley (Luo et al. 2023b), VideoChat (Li et al. 2023a), Video-ChatGPT (Maaz et al. 2023), and Video-LLaMA (Zhang, Li, and Bing 2023) as examples of traditional video LLMs. For video LLMs specifically designed for VTG tasks, we choose TimeChat (Ren et al. 2023), VTimeLLM (Huang et al. 2023), Momentor (Qian et al. 2024), and HawkEye (Wang et al. 2024a)³.

Numerical Results

R1. VTG-LLM achieves superior zero-shot performance on various VTG tasks. In Table 1, we evaluate the performance of the VTG-LLM on three VTG tasks, including dense video captioning (Youcook2), moment retrieval (Charades-STA), and video highlight detection (QVHighlights). The results indicate that: (1) The VTG-LLM model outperforms all existing 7B video LLM models on the zero-shot setting across all three datasets; (2) The improvement of the VTG-LLM on metrics that evaluate timestamp location

³Although related, LITA (Huang et al. 2024) employs self-built evaluation sets, making fair comparison difficult.

accuracy, such as the F1 score for the Youcook2 dataset and Recall for the Charades-STA dataset, is more pronounced. This suggests that the performance gain of the VTG-LLM primarily stems from more accurate timestamp location. Additionally, performance on other metrics like CIDEr and SODA_c also improves. (3) As shown in Table 3, the performance of VTG-LLM remains relatively robust across different video durations.

R2. Time embedding boost the model performance when using zero initialization technique. In the “Ablation Studies on STE” section of Table 2, we show the performance of VTG-LLM when only using sequence embedding (“SE Only”) and using sequence-time embedding but randomly initializing the weights of time-embedding (“TE Random Initialize”). The results indicate that: (1) the time-embedding is necessary for VTG-LLM to achieve the best result. Without the time-embedding, the performance of VTG-LLM on all datasets decreases; (2) Randomly initializing the time embedding significantly hurts the performance of VTG-LLM.

R3. Special time tokens significantly boost temporal grounding capacity with the cost of slightly caption quality reduction. In the “Ablation Studies on Time Token” section of Table 2, we show the performance of VTG-LLM when not using special time tokens (“No Time Token”), when not formatting the length of timestamps to six time tokens (“Time Token not Formatted”), and when do not initialize the token embedding and LLM head for time tokens (“Token Embedding not Initialized”). The results indicate that

- Using time tokens significantly boosts the model’s capacity to accurately locate the timestamps, improving the score on metrics like F1 score, R@1_(IOU=0.5), R@1_(IOU=0.7), and mAP. However, the caption-quality related metrics like SODA_c and CIDEr slightly decrease. This suggests that there exists a trade-off between caption quality and timestamp accuracy when using time token.

Model	Youcook2			Charades-STA		QVHighlights	
	SODA_c	CIDEr	F1 Score	R@1 _(IOU=0.5)	R@1 _(IOU=0.7)	mAP	HIT@1
<i>Ablation Studies on data</i>							
VTG-LLM (TimeIT)	1.0	2.8	12.0	35.1	14.8	14.9	19.1
<i>Ablation Studies on STE</i>							
SE Only	1.4	4.4	17.2	<u>33.6</u>	<u>14.8</u>	<u>16.4</u>	32.8
SE + TE (Random Initialize)	1.3	4.2	16.8	21.4	10.5	14.2	22.5
<i>Ablation Studies on Time Token</i>							
No Time Token	<u>1.7</u>	<u>5.5</u>	17.4	29.5	13.6	16.2	<u>33.9</u>
Time Token not Formatted	1.5	5.1	16.8	27.0	12.4	15.0	28.8
Token Embedding not Initialized	1.4	4.8	16.1	15.6	6.5	14.0	21.9
<i>Ablation Studies on Token Compression</i>							
Entropy Sampling	1.3	4.2	16.8	21.1	9.6	14.0	20.1
Diverse Sampling	1.3	4.4	16.8	22.4	10.5	14.2	22.7
Cross Attention	1.3	4.2	<u>17.6</u>	19.9	8.8	13.7	21.0
Original VTG-LLM	1.5	5.0	17.5	33.8	15.7	16.5	33.5

Table 2: Ablation studies on VTG-LLM.

YouCook2	CIDEr	METEOR	F1	SODA_c
[0s, 180s)	6.3	2.3	20.0	1.9
[180s, 240s)	4.6	1.8	18.8	1.4
[240s, 300s)	5.0	1.9	16.6	1.6
[300s, 420s)	5.6	2.1	17.9	1.5
[420s, inf)	4.1	1.6	15.1	1.3

Table 3: Performance with different video durations.

- We found that the "No Time Token" setting tends to predict fewer timestamps on QVHighlights dataset, which may lead to better HIT@1 performance but a worse mAP score.
- Formatting the timestamps using the same number of time tokens significantly enhances the model performance ("Time Token not Formatted"). We believe that this is because the formatted setting simplifies the task, making the models needless to identify the number of tokens in timestamps.
- Initialize the weights of token embedding and LLM prediction head is essential for time token to work well.

In summary, we recommend using special time tokens to achieve significant improvements in the accuracy of locating timestamps, which is the primary goal of this paper. Moreover, investigating the trade-off between caption quality and timestamp location accuracy in future work would be interesting.

R4. Slot-based token compression outperform naive compression baselines. In the "Ablation Studies on Token Compression" section of Table 2, we present the performance of VTG-LLM using different token compression methods and include some naive baselines. The "Entropy Sampling" method involves sampling the number of tokens

to 256 by selecting tokens with the maximum entropy, where the entropy is estimated using k-nearest neighbor distances (Kozachenko and Leonenko 1987). The "Diverse Sampling" method involves sampling tokens to 256 using k-means++ (Arthur, Vassilvitskii et al. 2007) to select tokens with the maximum distance. The "Cross Attention" method involves using a cross-attention layer to compress the number of tokens. The results indicate that: (1) The slot-based sampling achieves better performance among all compression methods; (2) Sampling-based methods perform poor compared to slot-based token compression method. We conjecture that this is because the sampling methods drop too much necessary information; (3) The cross-attention method performs even worse than sampling-based methods, which might be attributed to the attention mechanism requiring a large amount of data to perform well.

Conclusion, Limitation, and Future Works

In this paper, we propose the VTG-LLM, an improved video LLM model for integrating knowledge about timestamps, enhancing the zero shot performance on four downstream VTG tasks. Extensive numerical results demonstrate the superior zero-shot performance of VTG-LLM over state-of-the-art video LLM models on Charades-STA, QVHighlights, and Youcook2 datasets, with each proposed component contributing to individual performance gains.

Nonetheless, there are certain limitations in the VTG-LLM that necessitate additional research. For example, while this study mainly concentrates on the accuracy of timestamp locations, it would be advantageous to explore methods for improving other aspects, such as the precision of salient scores and the quality of captions. Furthermore, we did not incorporate the audio components of the videos in this study. Moreover, evaluating the performance of slot-based compression on a more diverse set of tasks is also worth investigating.

Acknowledgments

This work is supported in part by the funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society, in part by the Shenzhen Key Lab of Crowd Intelligence Empowered Low-Carbon Energy Network (Grant No. ZDSYS20220606100601002), in part by Shenzhen Stability Science Program 2023, and in part by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001).

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Arthur, D.; Vassilvitskii, S.; et al. 2007. k-means++: The advantages of careful seeding. In *Soda*, volume 7, 1027–1035.
- Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.
- Chen, S.; Li, H.; Wang, Q.; Zhao, Z.; Sun, M.; Zhu, X.; and Liu, J. 2024. Vast: A vision-audio-subtitle-text omnimodality foundation model and dataset. *Advances in Neural Information Processing Systems*, 36.
- Cheng, Z.; Leng, S.; Zhang, H.; Xin, Y.; Li, X.; Chen, G.; Zhu, Y.; Zhang, W.; Luo, Z.; Zhao, D.; et al. 2024. VideoLLaMA 2: Advancing Spatial-Temporal Modeling and Audio Understanding in Video-LLMs. *arXiv preprint arXiv:2406.07476*.
- Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.
- Fabian Caba Heilbron, B. G., Victor Escorcia; and Niebles, J. C. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 961–970.
- Fujita, S.; Hirao, T.; Kamigaito, H.; Okumura, M.; and Nagata, M. 2020. SODA: Story oriented dense video captioning evaluation framework. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 517–531. Springer.
- Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, 5267–5275.
- Ghosal, D.; Majumder, N.; Mehrish, A.; and Poria, S. 2023. Text-to-audio generation using instruction-tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*.
- Guo, Y.; Liu, J.; Li, M.; Tang, X.; Chen, X.; and Zhao, B. 2024. VTG-LLM: Integrating Timestamp Knowledge into Video LLMs for Enhanced Video Temporal Grounding. *arXiv preprint arXiv:2405.13382*.
- Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating summaries from user videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VII 13*, 505–520. Springer.
- Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2018. Localizing Moments in Video with Temporal Language. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Hua, H.; Tang, Y.; Xu, C.; and Luo, J. 2024. V2Xum-LLM: Cross-Modal Video Summarization with Temporal Prompt Instruction Tuning. *arXiv preprint arXiv:2404.12353*.
- Huang, B.; Wang, X.; Chen, H.; Song, Z.; and Zhu, W. 2023. Vtimellm: Empower llm to grasp video moments. *arXiv preprint arXiv:2311.18445*, 2(3): 9.
- Huang, D.-A.; Liao, S.; Radhakrishnan, S.; Yin, H.; Molchanov, P.; Yu, Z.; and Kautz, J. 2024. LITA: Language Instructed Temporal-Localization Assistant. *arXiv preprint arXiv:2403.19046*.
- Kozachenko, L. F.; and Leonenko, N. N. 1987. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2): 9–16.
- Lei, J.; Berg, T. L.; and Bansal, M. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858.
- Li, K.; He, Y.; Wang, Y.; Li, Y.; Wang, W.; Luo, P.; Wang, Y.; Wang, L.; and Qiao, Y. 2023a. VideoChat: Chat-Centric Video Understanding. *arXiv preprint arXiv:2305.06355*.
- Li, K.; Wang, Y.; Li, Y.; Wang, Y.; He, Y.; Wang, L.; and Qiao, Y. 2023b. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19948–19960.
- Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023a. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.
- Lin, K. Q.; Zhang, P.; Chen, J.; Pramanick, S.; Gao, D.; Wang, A. J.; Yan, R.; and Shou, M. Z. 2023b. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2794–2804.
- Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.
- Luo, D.; Huang, J.; Gong, S.; Jin, H.; and Liu, Y. 2023a. Towards generalisable video moment retrieval: Visual-dynamic injection to image-text pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23045–23055.
- Luo, R.; Zhao, Z.; Yang, M.; Dong, J.; Qiu, M.; Lu, P.; Wang, T.; and Wei, Z. 2023b. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.

- Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. S. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.
- Moreno-Torres, J. G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N. V.; and Herrera, F. 2012. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1): 521–530.
- Oncescu, A.-M.; Henriques, J. F.; Liu, Y.; Zisserman, A.; and Albanie, S. 2021. Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2265–2269. IEEE.
- Puigcerver, J.; Riquelme, C.; Mustafa, B.; and Hounsby, N. 2023. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*.
- Qian, L.; Li, J.; Wu, Y.; Ye, Y.; Fei, H.; Chua, T.-S.; Zhuang, Y.; and Tang, S. 2024. Momentor: Advancing Video Large Language Model with Fine-Grained Temporal Reasoning. *arXiv:2402.11435*.
- Ren, S.; Yao, L.; Li, S.; Sun, X.; and Hou, L. 2023. TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video Understanding. *arXiv preprint arXiv:2312.02051*.
- Song, E.; Chai, W.; Wang, G.; Zhang, Y.; Zhou, H.; Wu, F.; Chi, H.; Guo, X.; Ye, T.; Zhang, Y.; et al. 2024a. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18221–18232.
- Song, E.; Chai, W.; Ye, T.; Hwang, J.-N.; Li, X.; and Wang, G. 2024b. MovieChat+: Question-aware Sparse Memory for Long Video Question Answering. *arXiv preprint arXiv:2404.17176*.
- Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5179–5187.
- Sun, Q.; Fang, Y.; Wu, L.; Wang, X.; and Cao, Y. 2023. Evalclip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.
- Tang, Y.; Ding, D.; Rao, Y.; Zheng, Y.; Zhang, D.; Zhao, L.; Lu, J.; and Zhou, J. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1207–1216.
- Tong, Z.; Song, Y.; Wang, J.; and Wang, L. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35: 10078–10093.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4566–4575.
- Wang, Y.; Li, K.; Li, Y.; He, Y.; Huang, B.; Zhao, Z.; Zhang, H.; Xu, J.; Liu, Y.; Wang, Z.; et al. 2022. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*.
- Wang, Y.; Meng, X.; Liang, J.; Wang, Y.; Liu, Q.; and Zhao, D. 2024a. HawkEye: Training Video-Text LLMs for Grounding Text in Videos. *arXiv preprint arXiv:2403.10228*.
- Wang, Y.; Wang, Y.; Wu, P.; Liang, J.; Zhao, D.; Liu, Y.; and Zheng, Z. 2024b. Efficient Temporal Extrapolation of Multimodal Large Language Models with Temporal Grounding Bridge. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 9972–9987.
- Wu, Y.; Hu, X.; Sun, Y.; Zhou, Y.; Zhu, W.; Rao, F.; Schiele, B.; and Yang, X. 2024. Number it: Temporal Grounding Videos like Flipping Manga. *arXiv preprint arXiv:2411.10332*.
- Xiao, Y.; Luo, Z.; Liu, Y.; Ma, Y.; Bian, H.; Ji, Y.; Yang, Y.; and Li, X. 2023. Bridging the Gap: A Unified Video Comprehension Framework for Moment Retrieval and Highlight Detection. *arXiv preprint arXiv:2311.16464*.
- Xu, H.; Ghosh, G.; Huang, P.-Y.; Okhonko, D.; Aghajanyan, A.; Metze, F.; Zettlemoyer, L.; and Feichtenhofer, C. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*.
- Yang, A.; Nagrani, A.; Seo, P. H.; Miech, A.; Pont-Tuset, J.; Laptev, I.; Sivic, J.; and Schmid, C. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10714–10726.
- Zala, A.; Cho, J.; Kottur, S.; Chen, X.; Oguz, B.; Mehdad, Y.; and Bansal, M. 2023a. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23056–23065.
- Zala, A.; Cho, J.; Kottur, S.; Chen, X.; Oğuz, B.; Mehdad, Y.; and Bansal, M. 2023b. Hierarchical Video-Moment Retrieval and Step-Captioning. In *CVPR*.
- Zellers, R.; Lu, X.; Hessel, J.; Yu, Y.; Park, J. S.; Cao, J.; Farhadi, A.; and Choi, Y. 2021. MERLOT: Multimodal Neural Script Knowledge Models. In *Advances in Neural Information Processing Systems 34*.
- Zhang, H.; Li, X.; and Bing, L. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.
- Zhao, L.; Gundavarapu, N. B.; Yuan, L.; Zhou, H.; Yan, S.; Sun, J. J.; Friedman, L.; Qian, R.; Weyand, T.; Zhao, Y.; et al. 2024. VideoPrism: A Foundational Visual Encoder for Video Understanding. *arXiv preprint arXiv:2402.13217*.
- Zhou, L.; Xu, C.; and Corso, J. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. Minigt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.