Grounded-VideoLLM: Sharpening Fine-grained Temporal Grounding in Video Large Language Models

Anonymous ACL submission

Abstract

Despite their impressive performance in coarse-grained video understanding, Video Large Language Models (Video-LLMs) still face challenges in fine-grained temporal grounding, including ineffective temporal modeling and inadequate timestamp representations. In this work, we introduce Grounded-VideoLLM, a novel Video-LLM designed to perceive and reason over specific video moments with fine-grained temporal precision. Our model features (1) a two-stream encoder that explicitly captures inter-frame relationships while preserving intra-frame visual details and (2) discrete temporal tokens enriched with structured time knowledge for timestamp representation. Besides, we propose a multi-stage training strategy tailored to such grounding-specific architecture. The model is initially trained on simple video-caption tasks and progressively introduced to complex video temporal grounding tasks, ensuring a smooth learning curve and temporal alignment. We further strengthen Grounded-VideoLLM's temporal reasoning by constructing a VideoQA dataset with grounded information using an automated annotation pipeline. Extensive experiments demonstrate that Grounded-VideoLLM not only surpasses existing models in fine-grained grounding tasks but also exhibits strong potential as a general video understanding assistant.

1 Introduction

Multi-modal Large Language Models (MLLMs) have made remarkable progress in image-level understanding (Liu et al., 2023; Dai et al., 2023; Li et al., 2023b). However, extending their capabilities to the video domain poses distinct challenges. Unlike static images, the temporal nature of videos challenges models to process not only visual content but also the sequence and timing of events. While current Video-LLMs (Xu et al., 2024a; Li et al., 2024; Zhang et al., 2023b; Lin et al., 2023) are capable of capturing global visual semantics and generating coarse-grained captions for short clips, they struggle with fine-grained video understanding (Liu et al., 2024c; Wang et al., 2024d), which requires decomposing the video along the temporal axis to accurately perceive and reason over specific moments, such as subtle actions, transitions, and events that unfold over time.

Previous research efforts (Ren et al., 2024; Huang et al., 2024a; Qian et al., 2024a; Huang et al., 2024b; Guo et al., 2024) have explored *temporal grounding* to improve fine-grained video understanding. However, two main challenges impede their potential for achieving effective *fine-grained temporal grounding*: **(1)** Models like Video-ChatGPT (Maaz et al., 2024b), P-LLaVA (Xu et al., 2024a), and Video-LLAMA (Zhang et al., 2023b) typically sample multiple frames from a video and encode each frame independently using an image encoder, followed by a feature projector (e.g., sliding Q-former (Ren et al., 2024), visual adapter (Huang et al., 2024a)). This focuses primarily on spatial details while potentially neglecting the temporal relationships between frames since their visual encoders are solely trained on images. **(2)** Current models also struggle with timestamp representation, which is crucial for pinpointing specific moments in time for fine-grained understanding. Models such as TimeChat (Ren et al., 2024) and VTimeLLM (Huang et al., 2024a) represent timestamps as plain texts, for example, `["from 102.3 to 120.1 seconds"]`. Despite being straightforward, this needs to tokenize continuous floating-point values, which is inefficient for LLMs since their next-token prediction paradigm struggles with handling numerical data (Schwartz et al., 2024; Frieder et al., 2023). Although there have been some previous works (Yang et al., 2023; Huang et al., 2024b; Qian et al., 2024a) using special tokens to represent time positions, Vid2Seq (Yang et al., 2023) relies heavily on large-scale

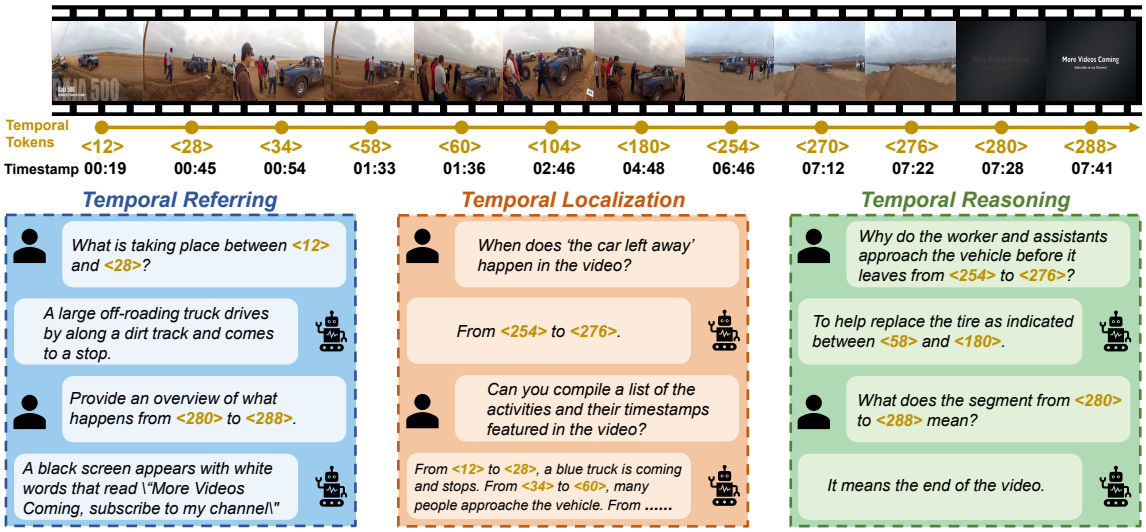

Figure 1: *Grounded-VideoLLM* enables Temporal Referring/Localizing/Reasoning for MLLMs.

pre-training from scratch using noisy transcribed speech and is limited to the task of dense video captioning, while LITA (Huang et al., 2024b) and Momentor (Qian et al., 2024a) only align these tokens with simple fine-tuning stage, which proves to be insufficient in our experiments (Table 4).

To further improve video understanding, we propose to sharpen the model with *fine-grained temporal grounding*, allowing the model to recognize not only *what* happens but pinpoint *when* it happens with finer granularity, as illustrated in Figure 1. Targeting these goals, we introduce *Grounded-VideoLLM*, a novel Video-LLM that can perceive and reason over specific video moments with fine-grained precision. **From the perspective of model architecture**, *Grounded-VideoLLM* is built upon two key innovations: (1) Two-Stream Encoding: We decompose each segment of the video into spatial and temporal components and encode each with an expert encoder. The temporal stream extracts motion representations from dense frames and complements the spatial stream, which captures appearance representations. This dual-stream approach forms comprehensive video representations enriched with both temporal and spatial information. (2) Temporal Tokens: We extend the LLM's vocabulary by introducing discrete tokens crafted to denote relative time positions and share a unified embedding space with the LLM, allowing *Grounded-VideoLLM* to avoid the inefficiency of tokenizing numerical text and seamlessly predict both timestamps and textual outputs in a single sequence of discrete tokens. **From the perspective of training**, we start with an image MLLM (Microsoft, 2024) as the foundation and adopt a three-stage training strategy. We meticulously select different tasks for each stage, and progressively refine the model in a "coarse-to-fine" manner, transitioning from image understanding to video comprehension, and ultimately to fine-grained temporal grounding. This scheme ensures the introduced temporal tokens align closely with the video timelines and LLM's semantic space, distinguishing our method from previous studies (Huang et al., 2024b; Qian et al., 2024a; Yang et al., 2023). Furthermore, we enhance the model's temporal reasoning by curating 17K grounded VideoQA (Xiao et al., 2024) samples with the assistance of GPT-4 (Achiam et al., 2023). Extensive experiments demonstrate that *Grounded-VideoLLM* shows promising results over existing Video-LLMs not only in traditional video temporal grounding tasks but also in general video understanding benchmarks.

2 Related Work

Video Large Language Models have caught a growing interest for general video understanding (Zhang et al., 2023b; Lin et al., 2023). However, they struggle with temporal perception (Liu et al., 2024c) and exhibit hallucination (Wang et al., 2024d) when asked about specific moments. They typically encode each frame independently using an image encoder to create video embeddings, resulting in video representations lacking inherent temporal information and heavily relying on the position embeddings of LLM for temporal understanding, limiting the model's capability to perform

fine-grained temporal grounding. In contrast, we employ a two-stream architecture that integrates a video expert to extract motion features to complement the appearance features during the early encoding process. This is different from traditional two-stream networks (Simonyan and Zisserman, 2014; Feichtenhofer et al., 2016) since we don't rely on heavy extraction of optical flows. Additionally, we employ a progressive training strategy that gradually adapts an image-based MLLM for fine-grained video understanding. Although concurrent works such as SlowFast-LLaVA (Xu et al., 2024b) and VideoGPT+ (Maaz et al., 2024a) also introduce an additional stream, SlowFast-LLaVA relies on a single image encoder to process each video frame without training, missing crucial temporal relationships between frames. VideoGPT+ merely arranges video tokens as a prefix to image tokens using sparse frames. Instead, our approach is specifically designed for fine-grained temporal grounding, leveraging a unique encoding, pooling, and training strategy tailored for dense frames, along with a dedicated grounding mechanism.

Video Temporal Grounding (VTG) tasks usually include Temporal Sentence Grounding (Gao et al., 2017; Hendricks et al., 2018), Dense Video Captioning (Caba Heilbron et al., 2015; Zhou et al., 2018), and Grounded VideoQA (Xiao et al., 2024). Given the emerging capabilities of Video-LLMs, many studies have investigated how to adapt them for VTG tasks. For example, TimeChat (Ren et al., 2024) and VTimeLLM (Huang et al., 2024a) perform temporal grounding using a fully text-to-text approach through instruction-tuning datasets. Momentor (Qian et al., 2024a) introduces a temporal perception module to address the quantization errors, and VTG-LLM (Guo et al., 2024) incorporates absolute-time tokens to handle timestamps. Compared to these, we avoid textual representation of timestamps and instead introduce discrete temporal tokens for timestamp encoding. Different from previous methods that also use special tokens (Huang et al., 2024b; Qian et al., 2024a; Yang et al., 2023; Peng et al., 2023), our model is more efficient by continuing training based on an established image MLLM with a two-stream architecture and a progressive training strategy.

3 Model Architecture

Given that current MLLMs already exhibit strong image-understanding capabilities, our architecture aims to sharpen temporal awareness by capturing motion dynamics across frames, which serve as a vital supplement to spatial content. As shown in Figure 2, we develop *Grounded-VideoLLM* upon a well-established MLLM for spatial comprehension and integrate an expert video encoder for temporal comprehension. Additionally, to avoid tokenizing numerical texts, we incorporate temporal tokens into the LLM's vocabulary for efficient and unified timestamp representation.

3.1 Two-Stream Encoding

Given a video $\mathcal{V}$ with T frames, we divide it into K segments and employ a segment-wise encoding strategy. Due to the inherent redundancy of consecutive frames, each segment can be naturally represented from two perspectives: spatial and temporal. The spatial representation of each segment is derived from an individual keyframe, capturing the primary appearance semantics, while the temporal representation is learned from multiple frames depicting the motion evolution within the segment.

Spatial Stream. We sample the middle frame from each segment as the keyframe and extract its spatial features using the original image encoder from the MLLM (Radford et al., 2021), resulting in spatial features $\mathbf{F}_S \in \mathbb{R}^{H_S \times W_S \times D_S}$, where H_S, W_S, D_S denote the height, width and dimension of the spatial features. Since dense frames are crucial for fine-grained temporal grounding, an appropriate pooling strategy is required to reduce token length. As indicated by (Xu et al., 2024a) and (Yao et al., 2024) that a 2D average pooling is both efficient and robust for spatial downsampling, we employ a 2D pooling kernel with a size $\sigma_S \times \sigma_S$ over the feature map and gets $\mathbf{F}_S \in \mathbb{R}^{N_S \times D_S}$ as the feature for spatial stream, where $N_S = \frac{H_S}{\sigma_S} \times \frac{W_S}{\sigma_S}$.

Temporal Stream. Traditional two-stream networks typically encode the optical flow as the temporal stream. However, given the scale of data and parameters of MLLMs, extracting optical flow is computationally expensive and impractical. Consequently, we resort to a strong and well pre-trained video encoder to extract motion representations for each segment, using a lower resolution but more frames. We input each segment, containing $\frac{T}{K}$ frames, into the video encoder to obtain the segment-level features $\mathbf{F}_T \in \mathbb{R}^{\frac{T}{K} \times H_T \times W_T \times D_T}$, where H_T, W_T, D_T denote the height, width and dimension of each frame feature. Similar to the spatial stream, we apply a 2D average pooling strat-

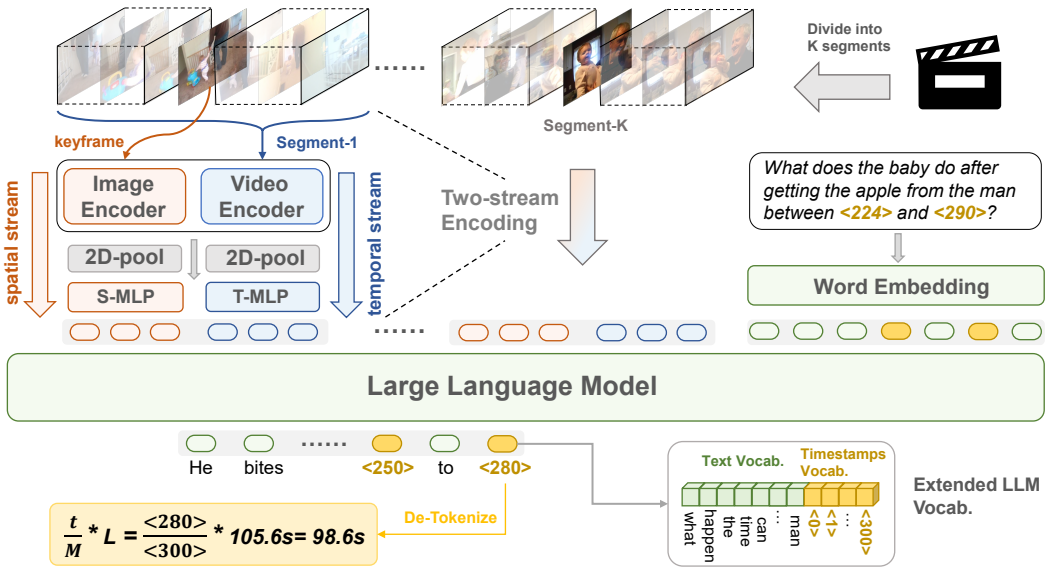

Figure 2: Overview of *Grounded-VideoLLM*. For temporal modeling, we employ a segment-wise encoding strategy by decomposing each segment into a spatial part and a temporal part and encoding each respectively. For timestamp representation, we introduce additional special temporal tokens sharing a unified embedding space with LLM.

egy to downsample $\mathbf{F}_T$. However, as the temporal stream focuses primarily on temporal modeling, we retain the complete temporal information by only pooling along the spatial dimensions. Specifically, we aggressively downsample $\mathbf{F}_T$ using a kernel with a larger size of $\sigma_T \times \sigma_T$, resulting in the compressed $\mathbf{F}_T \in \mathbb{R}^{\frac{T}{K} \times N_T \times D_T}$ for temporal stream, where $N_T = \frac{H_T}{\sigma_T} \times \frac{W_T}{\sigma_T}$. To get the complete segment-level representation, we flatten the features of both stream and concat them together :

$$\mathbf{F}_{Seg} = \text{Concat}\left[f(\mathbf{F}_S); g(\mathbf{F}_T)\right] \quad (1)$$

where $\mathbf{F}_{Seg} \in \mathbb{R}^{(N_S + \frac{T}{K} \cdot N_T) \times D}$, $f(\cdot)$ and $g(\cdot)$ are two MLPs that project the visual features to LLM's dimension D. The final video representation is formed by concatenating the K segment-level representations $\mathbf{F}_{Seg}$, resulting in $\mathbf{F}_{Vid} \in \mathbb{R}^{K \cdot (N_S + \frac{T}{K} \cdot N_T) \times D}$. This representation retains detailed spatial information across all segments along with their global temporal contexts, while maintaining a manageable token length. The combined video representation $\mathbf{F}_{Vid}$ is then fed into the LLM serving as soft prompts, alongside the word embeddings of the instruction text $\mathbf{F}_{Text}$ to generate the target response $\mathcal{A}$. The model is trained using the standard cross-entropy loss function for LLM.

3.2 Unified Temporal Tokens

Given a text depicting a video and its associated timestamps, we employ a relative time representation that converts continuous timestamps into a sequence of discrete tokens. For a video $\mathcal{V}$ with a duration of L seconds, we evenly divide $\mathcal{V}$ into M equal-length chunks, and then define $M+1$ anchor points (ranging from <0> to <M>) across $\mathcal{V}$, denoting relative temporal positions. Each anchor point corresponds to a specific timestamp and is encoded as a temporal token. For instance, <0> denotes the very start of $\mathcal{V}$ while <M> indicates the end. These $M+1$ tokens are added to the LLM's vocabulary to enable unified modeling alongside text. A specific continuous timestamp τ can be easily converted to a temporal token <t> and vice verse:

$$t = \text{Round}(M \cdot \frac{\tau}{L}), \quad \tau = L \cdot \frac{t}{M} \quad (2)$$

While this may introduce minor quantization errors, these can be minimized by selecting an appropriate M or an interpolation expansion. We then organize the text span and its corresponding temporal tokens in a unified format. Both text tokens and temporal tokens are mapped to embeddings through the extended word embedding layer of LLM.

This strategy avoids the need to tokenize and process numerical values, which has been identified as a limitation of LLMs (Schwartz et al., 2024). Notably, special tokens in LLMs are widely used in various domains. For example, Pix2Seq (Chen et al., 2021) leverages special tokens to represent spatial grounding in images, Open-VLA (Kim et al., 2024) and RT-2 (Brohan et al., 2023) utilize them for encoding robot action spaces, Yo'LLaVA

(Nguyen et al., 2024) use special tokens to refer to personalized subjects, and, most relevant to our work, Vid2Seq employs special tokens for temporal grounding in videos. However, the aforementioned works *lack effective strategies to align these tokens with both LLM's semantic meanings and specific functionalities*. For example, Vid2Seq (Yang et al., 2023) simply appends temporal tokens as a prefix to the caption and trains the entire T5 model from scratch using noised transcribed speech, which disrupts the language model's original semantic embedding. In contrast, we introduce a temporal token alignment training stage in Sec. 4 to mitigate this issue. Instead of training the entire model, we update only the word embeddings of temporal tokens and the final logit head, with carefully curated grounding-specific datasets. This ensures that temporal tokens are aligned with both the video timeline and the LLM's semantic space, and timestamps and text can be jointly decoded as a single sequence while maintaining the general video understanding ability.

4 Progressive Training

Different from previous methods (Zhang et al., 2023b; Lin et al., 2023) that train models from scratch using mixed image and video datasets, we start with a pre-trained image-based MLLM (Microsoft, 2024) and progressively enhance its fine-grained temporal grounding capabilities. This strategy can be applied to any off-the-shelf MLLM and is more efficient. Appendix Table 7 enumerates the datasets used at different stages.

Stage-1: Video-Caption Alignment. We leverage approximately 1.28 million video-text pairs sampled from diverse sources (Wang et al., 2024a; Bain et al., 2021; Chen et al., 2024b) to align video encoder's features with the MLLM. This alignment allows the MLLM, which was pre-trained solely on images, to gain a foundational understanding of videos. Since 2D down-sampling has been applied to the visual features, only the two projectors $f(\cdot)$ and $g(\cdot)$ are trainable, while the video encoder, image encoder, and LLM remain frozen. As this stage does not involve any temporal grounding tasks, the temporal tokens in Sec.3.2 are not yet incorporated.

Stage-2: Temporal Token Alignment. While Video-Caption Alignment connects videos and the MLLM at a coarse level, a gap persists between this alignment and fine-grained temporal grounding. To address this, we introduce the temporal tokens described in Sec.3.2 and continue pre-training the model on a diverse range of grounding datasets (Huang et al., 2024a; Qian et al., 2024a; Wang et al., 2024c), focusing on tasks such as Temporal Sentence Grounding, Dense Video Captioning, and Temporal Referring, which enables the model to refer to and localize temporal information effectively. Since new tokens are introduced, we additionally make the word embedding matrix and the final classifier head of the LLM trainable. This stage enhances the model's ability to comprehend multiple events and aligns the temporal tokens with both the video timelines and the LLM's semantic space as shown in Figure 3 and Table 4, distinguishing us from previous works (Huang et al., 2024b; Yang et al., 2023; Qian et al., 2024a).

Stage-3: Multi-Task Instruction Tuning. Following the initial two stages, the model has developed a basic understanding of video content and can locate specific timestamps. In this stage, we further enhance the model's fine-grained temporal grounding while improving its responsiveness to diverse instructions. To achieve this, we gather two types of datasets: (1) We compile a wide range of datasets for temporal grounding tasks, similar to Time-IT (Ren et al., 2024) and VTG-IT (Guo et al., 2024), which include tasks of dense video captioning (remove ActivityNet (Caba Heilbron et al., 2015)), temporal sentence grounding, and grounded VideoQA. (2) We incorporate a subset of instructional datasets from VideoChat2 (Li et al., 2024) and ShareGPT-4Video (Chen et al., 2024a) to further enhance the model's ability to generate detailed video captions. By utilizing these diverse datasets, which encompass both temporal grounding and video instruction tasks, *Grounded-VideoLLM* excels in temporal referring, localization, reasoning, and general comprehension of video content. In this stage, the trainable parameters remain the same as in Stage 2, with the addition of LoRA parameters (Hu et al., 2022) for the LLM.

5 Grounded VideoQA Dataset Generation

Grounded VideoQA requires the model to not only answer questions but identify timestamps that support predicted answers, demonstrating the temporal reasoning abilities. NExT-GQA (Xiao et al., 2024) was manually developed by extending NExT-QA (Xiao et al., 2021) with temporal labels for start and end timestamps. However, annotating these labels is labor-intensive and time-consuming, limit-

Model	LLM Scale	Charades-STA				ActivityNet-Grounding				ActivityNet-Captions	
		R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	SODA_c	METEOR
Video-LLaMA (Zhang et al., 2023b)	7B	25.2	10.6	3.4	16.8	21.9	10.8	4.9	16.5	1.9	1.9
SeViLA (Yu et al., 2023)	3B	27.0	15.0	5.8	18.3	31.6	19.0	10.1	23.0	-	-
Video-ChatGPT (Maaz et al., 2024b)	7B	27.2	6.2	1.9	19.7	19.5	10.6	4.8	14.2	1.9	2.1
Valley (Luo et al., 2023)	7B	28.4	1.8	0.3	21.4	30.6	13.7	8.1	21.9	0.3	0.8
VideoChat (Li et al., 2023c)	7B	32.8	8.6	0.0	25.9	23.5	12.6	6.0	17.4	0.9	0.9
Momenter (Qian et al., 2024a)	7B	42.6	26.6	11.6	28.5	42.9	23.0	12.4	29.3	2.3	4.7
VTimeLLM (Huang et al., 2024a)	7B	51.0	27.5	11.4	31.2	44.0	27.8	14.3	30.4	5.8	**6.8**
TimeChat (Ren et al., 2024)	7B	-	32.2	13.4	-	-	-	-	-	-	-
VTG-LLM (Guo et al., 2024)	7B	-	33.8	15.7	-	-	-	-	-	-	-
HawkEye (Wang et al., 2024c)	7B	50.6	31.4	14.5	33.7	**49.1**	29.3	10.7	32.7	-	-
Grounded-VideoLLM (Vicuna)	7B	51.8	34.3	18.3	34.7	43.9	29.1	18.3	34.5	**6.2**	6.4
Grounded-VideoLLM (Phi3.5)	4B	**54.2**	**36.4**	**19.7**	**36.8**	46.2	**30.3**	**19.0**	**36.1**	6.0	**6.8**

Table 1: Zero shot results on temporal sentence grounding and dense video captioning tasks.

Model	Acc@GQA	mIoP	IoP@0.5	mIoU	IoU@0.5
VIOLETv2 (Fu et al., 2023)	12.8	23.6	23.3	3.1	1.3
Temp[CLIP] NG+ (Xiao et al., 2024)	16.0	25.7	25.5	12.1	8.9
SeViLA (Yu et al., 2023)	16.6	29.5	22.9	**21.7**	13.8
LangRepo (Kahatapitiya et al., 2024)	17.1	31.3	28.7	18.5	12.2
FrozenBiLM NG+ (Yang et al., 2022)	17.5	24.2	23.7	9.6	6.1
VideoStreaming (Qian et al., 2024b)	17.8	32.2	31.0	19.3	13.3
LLoVi (Zhang et al., 2023a)	24.3	**37.3**	**36.9**	20.0	15.3
Grounded-VideoLLM (Vicuna)	24.0	32.2	31.2	20.8	16.9
Grounded-VideoLLM (Phi3.5)	**26.7**	34.5	34.4	21.1	**18.0**

Table 2: Results on NExT-GQA. Acc@GQA is defined as the percentage of questions that are both correctly answered and visually grounded with IoP ≥ 0.5.

ing NExT-GQA only to QA pairs for the validation and test sets. To create a scalable training dataset, we utilized OpenAI GPT-4 (Achiam et al., 2023) to assist in constructing training sets for the grounded VideoQA task. These sets were built on public datasets that already contain temporal labels, such as QVHighlights (Lei et al., 2021). We framed the task as a multiple-choice VideoQA using a two-round conversational format, as depicted in Appendix Figure 4 and detailed in Appendix A.3.

6 Experiments

Implementation Details. We select Phi3.5-V-Instruct-3.8B (Microsoft, 2024) as the base MLLM of *Grounded-VideoLLM*. We also build another *Grounded-VideoLLM* based on LLaVA-1.5-7B (Liu et al., 2024a) using Vicuna-1.5 (Chiang et al., 2023) as the LLM for fair comparison. For temporal stream, we adopt InternVideo2-1B (Wang et al., 2024b) as the video encoder. Each video is sampled as a sequence of $T = 96$ frames, which are evenly divided into $K = 12$ segments. We set the pooling size $\sigma_S = 2$ for the spatial stream ($N_S = 144$ tokens per frame) while $\sigma_T = 4$ ($N_T = 16$ tokens per frame) for the temporal stream. Moreover, we introduce $M = 300$ temporal tokens into the LLM's vocabulary for timestamp representation. We use the Phi3.5 version for ablations in Section 6.2. More details are in Appendix A.1.

Tasks and Benchmarks. We assess *Grounded-VideoLLM* across three video temporal grounding tasks: *Temporal Sentence Grounding*, *Dense Video Captioning*, and *Grounded VideoQA*, utilizing benchmarks such as **Charades-STA** (Gao et al., 2017), **ActivityNet-Captions** (Caba Heilbron et al., 2015), and **NExT-GQA** (Xiao et al., 2024). We also show its reasoning capability by the task of Open-Ended VideoQA with benchmarks including **MSVD-QA**, **MSRVTT-QA** (Xu et al., 2017), and **ActicityNet-QA** (Yu et al., 2019). Additionally, to evaluate the model's general video understanding capabilities, we benchmark *Grounded-VideoLLM* against existing models using **VCG-Bench** (Maaz et al., 2024b) and **MVBench** (Li et al., 2024). The evaluation details are in Appendix A.4.

6.1 Main Results

Temporal Sentence Grounding. As shown in Table 1, *Grounded-VideoLLM* (Phi3.5) achieves "mIoU" scores of 36.8 on Charades-STA and 36.1 on ANet-Grounding, respectively. This performance surpasses previous SoTA end-to-end Video-LLMs, such as HawkEye (Wang et al., 2024c), by a substantial margin of +3.4. It is worth emphasizing that this promising "mIoU" performance is largely attributed to significant gains in the "R@0.7" metric compared to other thresholds, demonstrating that *Grounded-VideoLLM* is more advanced in localizing specific moments with finer granularity. Interestingly, although *Grounded-VideoLLM* (Vicuna) may be larger than the Phi3.5 version in terms of model parameters, its overall performance is slightly lower. This is because the base Vicuna model is inherently weaker than Phi3.5. Nevertheless, when using the same LLM as the base model, *Grounded-VideoLLM* (Vicuna) still outperforms other models like TimeChat and VTimeLLM.

Dense Video Captioning. We evaluated *Grounded-VideoLLM* on the ANet-Captions, and the results in Table 1 show that *Grounded-VideoLLM* (Phi3.5) achieves the highest SODA_c score of 6.0, which demonstrates that, thanks to the Temporal Token Alignment training stage, *Grounded-VideoLLM* is highly effective in identifying the multi-event structure of the video and capturing complete storylines. The highest METEOR score (6.8) also indicates that *Grounded-VideoLLM* provides more detailed event descriptions compared with other Video-LLMs.

NExT-GQA (Xiao et al., 2024) requires the model to both correctly answer questions and provide timestamps that support the answers, highlighting the temporal reasoning capability. According to Table 2, *Grounded-VideoLLM* achieves the highest Acc@GQA (26.7, +2.4) and delivers comparable IoU and IoP scores to models such as SeViLA (Yu et al., 2023) and LLoVi (Zhang et al., 2023a), which use specialized grounding modules or rely on proprietary large language models (Achiam et al., 2023). The highest Acc@GQA score further demonstrates *Grounded-VideoLLM*'s capability in both fine-grained temporal grounding and high-level reasoning.

Open-Ended VideoQA. As shown in Table 3, *Grounded-VideoLLM* achieves state-of-the-art or comparative performance across MSVD-QA, MSRVTT-QA (Xu et al., 2017), and ActivityNet-QA (Yu et al., 2019), highlighting its advancements in general video question answering.

General Video-LLM Benchmarks. While *Grounded-VideoLLM* excels in fine-grained temporal grounding, we should ensure that it maintains performance in general video understanding. Therefore, we conducted a comprehensive evaluation using VCG-Bench (Maaz et al., 2024b) and MVBench (Li et al., 2024). As shown in Table 3, *Grounded-VideoLLM* achieves promising results in VCG-Bench, with an average score of 3.24, outperforming other Video-LLMs *with temporal grounding capabilities* (e.g., LITA, VTimeLLM). Notably, *Grounded-VideoLLM* surpasses all other Video-LLMs on the TU (Temporal Understanding) task (see Appendix Table 8), with a score of 3.12 (+7%), demonstrating its superior temporal understanding, which can be attributed to the two-stream architecture that can capture motion dynamics. For MVBench which provides 4,000 QA pairs spanning a wide range of scenes categorized into 20 fine-grained tasks, the results, presented in Table 3, show that *Grounded-VideoLLM* achieves an average score of 60.0, surpassing other Video-LLMs.

Model	MSVD-QA		MSRVTT-QA		ANet-QA		VCG-Bench	MVBench
	Acc.	Score	Acc.	Score	Acc.	Score	Avg.	Avg.
Video-LLMs w/o temporal grounding capability.								
Video-LLaMA (Zhang et al., 2023b)	51.6	2.5	29.6	1.8	12.4	1.1	1.98	34.1
Video-ChatGPT (Maaz et al., 2024b)	64.9	3.3	49.3	2.8	35.2	2.7	2.42	32.7
Video-LLaVA (Lin et al., 2023)	70.7	3.9	59.2	3.5	45.3	3.3	-	43.0
Vista-LLaMA (Ma et al., 2024)	65.3	3.6	60.5	3.3	48.3	3.3	2.57	-
MovieChat (Song et al., 2024)	75.2	3.8	52.7	2.6	45.7	3.4	2.67	-
LongVLM (Weng et al., 2024)	70.0	3.8	59.8	3.3	47.6	3.3	2.89	-
VideoChat2 (Li et al., 2024)	70.0	3.9	54.1	3.3	49.1	3.3	2.98	51.1
Chat-UniVi (Jin et al., 2024)	65.0	3.6	54.6	3.1	45.8	3.2	2.99	-
P-LLaVA-7B (Xu et al., 2024a)	**76.6**	**4.1**	62.0	3.5	56.3	**3.5**	3.12	46.6
ST-LLM (Liu et al., 2024b)	74.6	3.9	**63.2**	3.4	50.9	3.3	3.15	54.9
VideoGPT+ (Maaz et al., 2024a)	-	-	-	-	-	-	**3.28**	58.7
Video-LLMs w/ temporal grounding capability.								
TimeChat (Huang et al., 2024b)	-	-	-	-	-	-	-	38.5
Momentor (Qian et al., 2024a)	68.9	3.6	55.6	3.0	40.8	3.2	-	
VTimeLLM (Huang et al., 2024a)	-	-	-	-	-	-	2.85	
LITA (Huang et al., 2024b)	-	-	-	-	-	-	3.04	
Grounded-VideoLLM (Vicuna)	74.7	3.9	61.9	**3.6**	55.7	3.4	3.26	58.1
Grounded-VideoLLM (Phi3.5)	76.3	**4.1**	60.3	**3.6**	**56.8**	**3.5**	3.24	**60.0**

Table 3: Results on VideoQA, VCG-Bench and MVBench. Refer to Appendix Table 9, 8 for details.

Model	# of video tokens	C-STA	ANet-G	ANet-Cap	MVBench
		mIoU	mIoU	SODA_c	Avg.
Grounded-VideoLLM	3264	**36.8**	**36.1**	**6.0**	**60.0**
w/o temporal-stream (sparse)	3456	30.4	28.0	4.9	58.5
w/o temporal-stream (dense)	3456	34.3	29.2	5.4	53.2
w/o spatial-stream	3584	33.5	28.7	5.5	57.7
w/o temporal token alignment	3264	27.5	23.1	4.7	58.9

Table 4: Impact of two-stream and token alignment.

6.2 In-Depth Analysis

Two-stream Encoding. We conduct ablations to our two-stream encoding. Specifically, we set three variants by removing the temporal stream or spatial stream respectively: (1) *w/o temporal-stream (dense)* feeds $T = 96$ frames with a pooling size $\sigma_S = 4$ (36 tokens per frame). (2) *w/o temporal-stream (sparse)* feeds $T = 24$ frames with a pooling size $\sigma_S = 2$ (144 tokens per frame). (3) *w/o spatial-stream* feeds $T = 14$ frames without pooling (256 tokens per frame). All these variants have a close number of video tokens compared to our two-stream encoding ($12 \times (144 + \frac{96}{12} \times 16) = 3264$ tokens for $\mathbf{F}_{Vid} \in \mathbb{R}^{K \cdot (N_S + \frac{T}{K} \cdot N_T) \times D}$) for a fair comparison. Table 4 shows that, interestingly, the dense frame variant performs slightly better in temporal grounding tasks while worse in general benchmarks than the sparse frame variant. This can be attributed to that the videos in MVBench are much shorter and emphasize spatial details more. Our two-stream architecture strikes a balance by capturing dense motion dynamics while maintaining essential appearance details.

Temporal Tokens. We conducted ablations on the temporal tokens to study how they will affect the grounding performance. The models in Table 5 were trained using only the first two stages and evaluated on grounding tasks. Specifically, 'w/o tem-

Figure 3: Attention weights of the LLM when generating the temporal tokens and 3D-PCA of embeddings.

Setting	C-STA (mIoU)	ANet-G (mIoU)	ANet-Cap (SODA_c)
w/o temporal tokens	32.3	29.6	5.6
w/ 100 temporal tokens	32.2	29.1	5.2
w/ 200 temporal tokens	32.9	30.1	5.5
w/ 300 temporal tokens	**33.8**	**33.1**	**5.7**

Table 5: Impact of temporal tokens.

Model	NExT-GQA		
	Acc@GQA	mIoP	mIoU
Grounded-VideoLLM	26.7	34.5	21.1
w/o grounded VideoQA	18.1 (↓ 8.6)	22.2 (↓ 12.3)	12.9 (↓ 8.2)

Table 6: Impact of grounded VideoQA dataset.

poral tokens' refers to directly using plain text to represent absolute timestamps. The results in Table 5 show that while the performance of plain text and 100 temporal tokens is comparable, both are outperformed by 300 tokens. Furthermore, the results reveal a consistent improvement in performance with an increasing number of temporal tokens, especially for longer videos (ANet), highlighting the benefit of finer-grained time representations.

Alignment Training Stage. We investigate temporal tokens' role by ablating the 2nd training stage of Temporal Token Alignment. Quantitative results in Table 4 reveal a performance drop across all tasks, particularly in temporal sentence grounding. Upon analyzing the outputs, we found that the model often produces time intervals spanning nearly the entire video (e.g., from `<0> to <300>`), neglecting the alignment between specific moments and temporal tokens. Qualitatively, we visualize the attention weights of the LLM to demonstrate how temporal tokens attend to corresponding video moments. Details of generating visualizations are provided in Appendix A.5. As shown in Figure 3 (a), when generating the temporal token, e.g. <241> or <271>, the attention weights are higher and more focused on their corresponding video moments. Conversely, in Figure 3 (b), when the model is trained without the alignment stage, the attention weights of temporal tokens become significantly dispersed across irrelevant moments. This illustrates the necessity of our multi-stage training strategy for temporal alignment. We also visualize the embedding distribution of temporal tokens with PCA in Figure 3 (c), revealing that temporal tokens with similar indices tend to cluster together, exhibiting a continuous transition from tokens with smaller indices to larger ones.

Grounded VideoQA Dataset. We validate the role of our constructed grounded-VideoQA by removing it from the stage-3 data. Since the model without training on our dataset usually generates free-form texts when asked to output the timestamps supporting the answer, we reformulate it as a temporal sentence grounding task, where we combine the predicted answer and question into a single sentence and ask the model to localize its timestamps. Table 6 suggests that there is a significant performance decrease with regard to Acc@GQA (↓ 8.6), mIoP (↓ 12.3), and mIoU (↓ 8.2), from which we can conclude that our Grounded VideoQA dataset is essential for the model's temporal reasoning capability.

7 Conclusion

We present *Grounded-VideoLLM*, a novel Video-LLM that incorporates a two-stream encoding for temporal modeling, along with the temporal tokens for timestamp representation. We employ a multi-stage training scheme, starting with an image-based MLLM and progressively equipping it with fine-grained temporal grounding capabilities. Additionally, we curate a grounded-VideoQA dataset to further enhance the model's temporal reasoning ability. Extensive experiments demonstrate that *Grounded-VideoLLM* not only excels in video temporal grounding tasks but also performs strongly on general video understanding benchmarks.

8 Limitations

While Grounded-VideoLLM demonstrates superiority in handling fine-grained temporal grounding, but it still has some inherent limitations for future works. (1) Timestamp Quantization Error: Although discrete temporal tokens are introduced to represent timestamps and accuracy is improved by increasing the number of tokens, minor quantization errors may still be introduced when converting continuous time into discrete tokens. (2) Computational Resource Requirements: The training and inference processes of the model, especially the parts involving two-stream encoding and large-scale language models, may require more computational resources than single-stream.

References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

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

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. 2024a. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. 2021. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and 1 others. 2024b. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, pages 13320–13331.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical capabilities of chatgpt. In *NeurIPS*.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. 2023. An empirical study of end-to-end video-language transformers with masked visual modeling. In *CVPR*.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Soda: Story oriented dense video captioning evaluation framework. In *ECCV*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*, pages 5267–5275.

Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. 2024. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv preprint arXiv:2405.13382*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2018. Localizing moments in video with temporal language. In *EMNLP*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. 2024a. Vtimellm: Empower llm to grasp video moments. In *CVPR*.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. 2024b. Lita: Language instructed temporal-localization assistant. *arXiv preprint arXiv:2403.19046*.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. 2024. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*.

Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. 2024. Language repository for long video understanding. *arXiv preprint arXiv:2403.14622*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, pages 11846–11858.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2023a. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023c. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

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

Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. 2024b. St-llm: Large language models are effective temporal learners. In *ECCV*.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. 2024c. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. 2023. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.

Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. 2024. Vista-llama: Reducing hallucination in video language models via equal distance to visual tokens. In *CVPR*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024a. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arXiv:2406.09418*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024b. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*.

Microsoft. 2024. microsoft/phi-3-vision-128k-instruct.

Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. 2024. Yo'llava: Your personalized language and vision assistant. *arXiv preprint arXiv:2406.09400*.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. 2024a. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *ICML*.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2024b. Streaming long video understanding with large language models. *arXiv preprint arXiv:2405.16009*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*.

Eli Schwartz, Leshem Choshen, Joseph Shtok, Sivan Doveh, Leonid Karlinsky, and Assaf Arbelle. 2024. Numerologic: Number encoding for enhanced llms' numerical reasoning. *arXiv preprint arXiv:2404.00459*.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, and 1 others. 2024a. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, and 1 others. 2024b. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*.

Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. 2024c. Hawkeye: Training video-text llms for grounding text in videos. *arXiv preprint arXiv:2403.10228*.

Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. 2024d. Videohallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. In *ECCV*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. 2024. Can i trust your answer? visually grounded video question answering. In *CVPR*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024a. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. 2024b. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, pages 10714–10726.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. 2024. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, and 1 others. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. 2023. Self-chained image-language model for video localization and question answering. In *NeurIPS*.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2023a. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*.

Hang Zhang, Xin Li, and Lidong Bing. 2023b. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI*, volume 32.

A Appendix

A.1 More Implementation Details

Phi3.5-Vison-Instruct (Microsoft, 2024) consists of a CLIP style ViT image encoder (Radford et al., 2021), an MLP projector $f(\cdot)$, and the large language model Phi3.5-mini-3.8B (Abdin et al., 2024). Each video is sampled as a sequence of $T = 96$ frames, which are evenly divided into $K = 12$ segments. For the spatial stream encoded by the ViT in Phi3.5-V, we adopt a higher resolution 336×336, but a lower resolution 224×224 for the temporal stream encoded by InternVideo-2. We set the pooling size σ_S to be 2 while σ_T to be 4 respectively. For the spatial stream, each frame takes up $24 \times 24 = 576$ tokens before while $12 \times 12 = 144$ tokens after pooling. For the temporal stream, each frame takes up $16 \times 16 = 256$ tokens before while $4 \times 4 = 16$ tokens after pooling. Therefore, we have an overall of $K \times (144 + \frac{T}{K} \times 16) = 3264$ tokens in total. During training, we use the AdamW optimizer with a cosine learning rate decay and set the learning rate as 2e-5 and 1e-3 for projector $f(\cdot)$ and $g(\cdot)$ in stage-1. During stage-2 and stage-3, we set the learning rate for both projectors and word embeddings as 2e-5, while 2e-4 for LoRA parameters (r = 128 and α = 256). All experiments are conducted on NVIDIA A100/H800 GPUs.

A.2 Instructions for Each Task

The quality and diversity of instructions are essential in the training process. We manually write well-designed instructions for some tasks, combined with some templates in Time-IT (Ren et al., 2024). Table 13 lists the prompts for different tasks.

A.3 Grouned-VideoQA Dataset Generation

Specifically, we input event descriptions along with their timestamps into GPT-4 and prompted it to generate corresponding question-answer pairs, as shown in Table 12. To create distractor options for the multiple-choice questions, we retrieved the top 50 questions most similar to the generated question, based on cosine similarity using an embedding model (Reimers, 2019). The answers to these 50 retrieved questions served as candidates for distractors. From this pool, we randomly sampled four distractors with cosine similarities to the correct answer ranging from 0.2 to 0.9, ensuring that the distractors were contextually relevant but not overly similar to the correct answer. The ground-truth timestamps for answering each question were derived from the timestamps of the associated event descriptions. The constructed dataset comprises 17K samples, which have been incorporated into the training sets for Stage 3, further enhancing the model's temporal reasoning performance.

A.4 Evaluation Process

For temporal sentence grounding, we report the metric of Intersection over Union (**IoU**) (Gao et al., 2017) between the timestamps predicted by the model and the ground truth, including **Recall at IoU** thresholds of {0.3, 0.5, 0.7} and their **mean IoU**. For dense video captioning, we use metrics including **SODA_c** (Fujita et al., 2020) which is specifically tailored for the video's storyline, and **METEOR** score (Banerjee and Lavie, 2005), which is the average of traditional METEOR scores that are calculated based on matched pairs between generated events and the ground truth across IoU thresholds of {0.3, 0.5, 0.7, 0.9}. For Visually-grounded VideoQA, we calculate both the Intersection of Prediction (**IoP**) (Xiao et al., 2024) and Intersection of Union (**IoU**), and use **Acc@GQA** (Xiao et al., 2024) to measure the percentage of questions that are both correctly answered and visually grounded with IoP ≥ 0.5. The responses of Open-Ended VideoQA and VCG-Bench are evaluated by GPT-3.5 with the prompts introduced by Video-ChatGPT (Maaz et al., 2024b).

For the evaluation of the temporal sentence grounding task, we directly input the prompt ["`At which time interval in the video can we see` $< query >$ `occurring?`"] in Table 13 to get the response ["`From` $< start >$ `to` $< end >$"], and then calculate the predicted timestamps with the Equation (2) to get the IoU metrics.

For the evaluation of the dense video captioning task, we directly input the prompt ["`Detect and report the start and end timestamps of activity events in the video, along with descriptions.`"] in Table 13 to get the response ["`From` $< start_1 >$ `to` $< end_1 >$, $< caption_1 >$. `From` $< start_2 >$ `to` $< end_2 >$, $< caption_2 >$. $\cdots$"], and then calculate the SODA_c (Fujita et al., 2020) and Meteor scores (Banerjee and Lavie, 2005).

For the evaluation of the visually-grounded VideoQA task, we adopt a two-round conversation evaluation:

Training Stage	Task	# of Samples	Datasets
Video-Caption Alignment	Video Captioning	1.28M	WebVid-10M, Panda-70M, InternVid-10M
Temporal Token Alignment	Temporal Sentence Grounding	149K	VTimeLLM-Stage2
	Dense Video Captioning	92K	VTimeLLM-Stage2, Moment-10M, InternVid-G
	Temporal Referring	95K	VTimeLLM-Stage2, InternVid-G
Multi-Task Instruction Tuning	Grounded Conversation	442K	RTL, Moment-10M
	Temporal Sentence Grounding	84K	DiDeMo, HiREST, QuerYD, VTG-IT
	Dense Video Caption	41K	COIN, ViTT, YouCook2, VTG-IT
	Grounded VideoQA	17K	Self Collected
	Converstation	233K	VCG-Plus-112K, Videochatgpt-100K, Videochat2-Conv
	VideoQA	282K	EgoQA, NExT-QA, Intent-QA, STAR, CLEVRER, WebVid-QA
	Classification	66K	SthSthV2, Kinetics
	Video Captioning	136K	TextVR, YouCook2, WebVid, ShareGPT4Video

Table 7: Datasets at three training stages. Tasks with gray consist of datasets regarding temporal grounding.

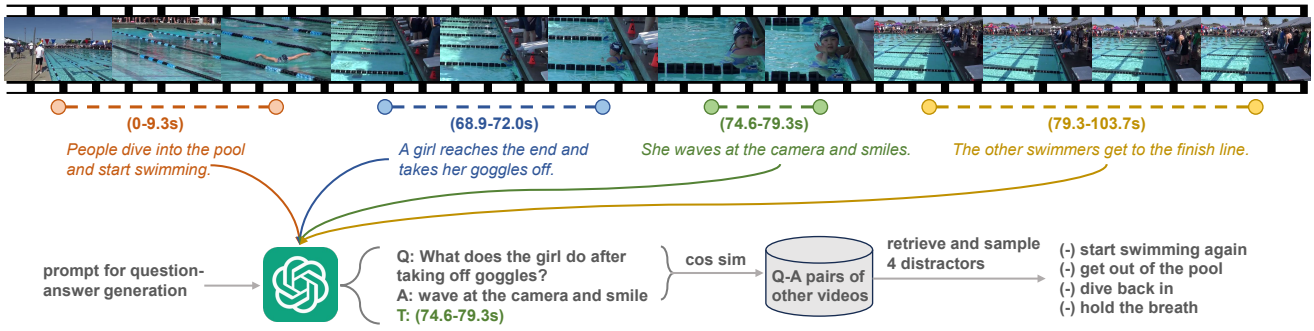

Figure 4: Examples of annotation pipeline and generated data for Grounded VideoQA.

Model	VCG-Bench					
	CI	DO	CU	TU	CO	Avg.
Video-LLaMA (Zhang et al., 2023b)	1.96	2.18	2.16	1.82	1.79	1.98
Video-ChatGPT (Maaz et al., 2024b)	2.50	2.57	2.69	2.16	2.20	2.42
Vista-LLaMA (Ma et al., 2024)	2.44	2.64	3.18	2.26	2.31	2.57
MovieChat (Song et al., 2024)	2.76	2.93	3.01	2.24	2.42	2.67
LongVLM (Weng et al., 2024)	2.76	2.86	3.34	2.39	3.11	2.89
VideoChat2 (Li et al., 2024)	3.02	2.88	3.51	2.66	2.81	2.98
Chat-UniVi (Jin et al., 2024)	2.89	2.91	3.46	2.89	2.81	2.99
P-LLaVA-7B (Xu et al., 2024a)	3.21	2.86	3.62	2.33	2.93	3.12
ST-LLM (Liu et al., 2024b)	3.23	3.05	**3.74**	2.93	2.81	3.15
VideoGPT+ (Maaz et al., 2024a)	3.27	**3.18**	**3.74**	2.83	**3.39**	**3.28**
VTimeLLM (Huang et al., 2024a)	2.78	3.10	3.40	2.49	2.47	2.85
LITA (Huang et al., 2024b)	2.94	2.98	3.43	2.68	3.19	3.04
Grounded-VideoLLM	**3.34**	2.94	3.66	**3.12**	3.14	3.24

Table 8: Results on VCG-Bench. VCG-Bench contains five aspects: Correctness of Information (CI), Detail Orientation (DO), Contextual Understanding (CU), Temporal Understanding (TU), and Consistency (CO).

Round-1:
USER: $<$ *question* $>$. $<$ *options* $>$.
ASSISTANT: Answer: $<$ *answer* $>$.
Round-2:
USER: Provide the timestamps that correspond to your answer.
ASSISTANT: From $<$ *start* $>$ to $<$ *end* $>$.

In the first round, we input the question and options into the model and get the answer. In the second round, we input the question, options, and predicted answer as the contexts, together with the prompt ["Provide the timestamps that correspond to your answer."], into the model to get the predicted timestamps. With both the predicted answers and timestamps, we can calculate the metrics including IoU, IoP, and Acc@GQA (Xiao et al., 2024).

For the evaluation of the Open-ended VideoQA and VCG-Bench, we employed GPT-3.5 turbo to juxtapose model outputs with ground truth data as introduced by Video-ChatGPT (Maaz et al., 2024b), subsequently computing both accuracy and a score. To ensure a fair and consistent comparison with the results presented in Video-ChatGPT, we adopted the same prompt for our evaluations. For MVBench, we directly compute the accuracy of multiple-choice questions.

A.5 Visualization Process

We visualize the attention weights from the last layer of the LLM during the generation of a new temporal token. Since the full video representation consists of a total of $K \times (N_S + \frac{T}{K} \times N_T)$ tokens—where T, K, N_S, and N_T denote the number of frames, number of segments, number of tokens per frame for the spatial stream, and number of tokens per frame for the temporal stream,

Model	Avg.	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI
Otter-V (Li et al., 2023a)	26.8	23.0	23.0	27.5	27.0	29.5	53.0	28.0	33.0	24.5	23.5	27.5	26.0	28.5	18.0	38.5	22.0	22.0	23.5	19.0	19.5
mPLUG-Owl-V (Ye et al., 2023)	29.7	22.0	28.0	34.0	29.0	29.0	40.5	27.0	31.5	27.0	23.0	29.0	31.5	27.0	40.0	44.0	24.0	31.0	26.0	20.5	29.5
VideoChatGPT (Maaz et al., 2024b)	32.7	23.5	26.0	62.0	22.5	26.5	54.0	28.0	40.0	23.0	20.0	31.0	30.5	25.5	39.5	48.5	29.0	33.0	29.5	26.0	35.5
VideoLLaMA (Zhang et al., 2023b)	34.1	27.5	25.5	51.0	29.0	39.0	48.0	40.5	38.0	22.5	22.5	43.0	34.0	22.5	32.5	45.5	32.5	40.0	30.0	21.0	37.0
VideoChat (Li et al., 2023c)	35.5	33.5	26.5	56.0	33.5	40.5	53.0	40.5	30.0	25.5	27.0	48.5	35.0	20.5	42.5	46.0	26.5	41.0	23.5	23.5	36.0
TimeChat (Ren et al., 2024)	38.5	40.5	36.0	61.0	32.5	53.0	53.5	41.5	29.0	19.5	26.5	66.5	34.0	20.0	43.5	42.0	36.5	36.0	29.0	35.0	35.0
Video-LLaVA (Lin et al., 2023)	43.0	46.0	42.5	56.5	39.0	53.5	53.0	48.0	41.0	29.0	31.5	82.5	45.0	26.0	53.0	41.5	33.5	41.5	27.5	38.5	31.5
P-LLaVA-7B (Xu et al., 2024a)	46.6	58.0	49.0	55.5	41.0	61.0	56.0	61.0	36.0	23.5	26.0	82.0	39.5	42.0	52.0	45.0	42.0	53.5	30.5	48.0	31.0
VideoChat2 (Li et al., 2024)	51.1	66.0	47.5	83.5	49.5	60.0	58.0	71.5	42.5	23.0	23.0	88.5	39.0	42.0	58.5	44.0	49.0	36.5	35.0	40.5	65.5
ShareGPT4Video (Chen et al., 2024a)	51.2	49.5	39.5	79.5	40.0	54.5	82.5	54.5	32.5	50.5	41.5	84.5	35.5	62.5	75.0	51.0	25.5	46.5	28.5	39.0	51.5
ST-LLM (Liu et al., 2024b)	54.9	66.0	53.5	84.0	44.0	58.5	80.5	73.5	38.5	42.5	31.0	86.5	36.5	56.5	78.5	43.0	44.5	46.5	34.5	41.5	58.5
VideoGPT+ (Maaz et al., 2024a)	58.7	69.0	60.0	83.0	48.5	66.5	85.5	75.5	36.0	44.0	34.0	89.5	39.5	71.0	90.5	45.0	53.0	50.0	29.5	44.0	60.0
Grounded-VideoLLM	60.0	75.0	75.5	83.0	50.0	63.0	88.0	77.0	37.0	41.5	50.0	91.5	45.0	57.5	82.0	49.5	55.0	45.5	32.0	42.0	59.0

Table 9: Results on MVBench. MVBench contains 20 aspects: Action Sequence (AS), Action Prediction (AP), Action Antonym (AA), Fine-grained Action (FA), Unexpected Action (UA), Object Existence (OE), Object Interaction (OI), Object Shuffle (OS), Moving Direction (MD), Action Localization (AL), Scene Transition (ST), Action Count (AC), Moving Count (MC), Moving Attribute (MA), State Change (SC), Fine-grained Pose (FP), Character Order (CO), Egocentric Navigation (EN), Episodic Reasoning (ER), Counterfactual Inference (CI), and the average of all 20 metrics (Avg).

Model	LLM	Charades-STA				ActivityNet-Grounding				ActivityNet-Captions	
	Scale	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	SODA_c	METEOR
Grounded-VideoLLM	4B	70.2	55.9	33.2	49.4	64.9	47.8	30.4	47.2	6.6	6.5

Table 10: More results on temporal sentence grounding and dense video captioning tasks. We incorporate a subset of Charades-STA and ActivityNet-Captions into the 3rd training stage and achieve better performance.

respectively—we obtain an attention weight vector with the shape $[K \times (N_S + \frac{T}{K} \times N_T), 1]$. First, we discard the spatial stream portion, focusing only on the temporal information, which results in a new vector with the shape $[K \times \frac{T}{K} \times N_T, 1]$. We then reshape this vector to the form $[T, N_T, 1]$ and average it along the spatial dimension, yielding $[T, 1]$, which represents the final attention weights corresponding to each frame when generating a new token.

A.6 Distribution of Temporal Tokens

We visualize the embeddings of $M = 300$ temporal tokens to investigate their distribution in embedding space. We employ PCA (Abdi and Williams, 2010) to reduce the dimensionality of the temporal tokens to 1D, 2D, and 3D representations. For all reductions, we use the reduced values as coordinates, incorporating a gradient color scheme in which the color of the data points changes progressively with the token index, as illustrated in Figure 5. Our observations reveal that temporal tokens with similar indices tend to cluster together, exhibiting a continuous transition from tokens with smaller indices (light colors) to those with larger indices (darker colors).

A.7 More experiment results

As shown in Table 10, we incorporate a subset of the training sets of Charades-STA and ActivityNet-Captions into the 3rd training stage and re-train the model from the checkpoint of the 2nd training stage, which achieves better performance. This also greatly improves the performance on NExT-GQA as illustrated in Table 11.

Model	Acc@GQA	mIoP	IoP@0.3	IoP@0.5	mIoU	IoU@0.3	IoU@0.5
Grounded-VideoLLM	29.4	37.4	48.0	36.5	27.0	41.0	25.8

Table 11: Results on NExT-GQA. We incorporate a subset of Charades-STA and ActivityNet-Captions into the 3rd training stage and achieve better performance.

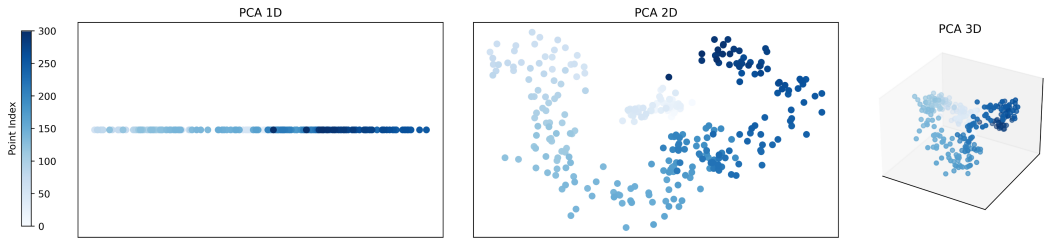

Figure 5: Visualization of temporal tokens with PCA.

System:
You are a good question generator. I need your help in generating question-answer pairs pertaining to the visual event descriptions. I have a video and I will provide you with descriptions of certain segments and their corresponding timestamps within this video. You need to consider these segments comprehensively based on the given description and timestamps and select one segment which you think can provide a HIGH-QUALITY QUESTION. Based on the description of that segment, ask a question related to that segment, as well as one correct answer. Both the proposed answer and question should be consistent with the content of the give description. BE CAREFUL! Your proposed questions and answers should follow these rules:
(0) Avoid choosing the segment spanning across the whole video.
(1) The question you raised should include causal and temporal relationships as much as possible. Question types should be diverse including WHY, HOW, WHAT, WHERE, etc.
(2) NEVER involve anything that is not covered in the given descriptions.
(3) The answer should NEVER appear in your question.
(4) Your answer should be a phrase no more than 7 words. Keep your answers concise and accurate.

Demonstrations:

User:
video duration: 82.73 seconds
segment-1: [0.83, 19.86] A young woman is seen standing in a room and leads into her dancing.
segment-2: [17.37, 60.81] The girl dances around the room while the camera captures her movements.
segment-3: [56.26, 79.42] She continues dancing around the room and ends by laying on the floor.
Response:
chosen segment: segment-3
segment timestamps: [56.26, 79.42]
question: What did the girl do after she ended dancing?
answer: lay on the floor

. . .
(other in context demonstrations)
. . .

Table 12: Prompts used to generate visually-grounded VideoQA samples with GPT-4.

Prompts for Video-Caption Alignment:
(1) "Describe the following video concisely.",
(2) "Provide a brief description of the given video clip.",
(3) "Offer a succinct explanation of the footage presented.",
(4) "Summarize the visual content of the following video.",
(5) "Give a short and clear explanation of the subsequent video clip.",
(6) "Share a concise interpretation of the video provided.",
(7) "Present a compact description of the clip's key features.",
(8) "Relay a brief, clear account of the video shown.",
(9) "Render a clear and concise summary of the video below.",
(10) "Write a terse but informative summary of the following video clip."

Prompts for Temporal Sentence Grounding:
(1) "When does $< query >$ happen in the video?",
(2) "At what time does the occurrence $< query >$ take place in the video?",
(3) "During which part of the video does $< query >$ occur?",
(4) "When in the video does the $< query >$ incident occur?",
(5) "At which moment does $< query >$ take place in the video?",
(6) "During which phase of the video does $< query >$ happen?",
(7) "When does the $< query >$ event occur in the video?",
(8) "At what time does $< query >$ occur in the video sequence?",
(9) "When does the $< query >$ situation take place in the video?",
(10) "At which time interval in the video can we see $< query >$ occurring?"

Prompts for Dense Video Captioning:
(1) "Localize a series of activity events in the video, output the start and end timestamp for each event, and describe each event with sentences.",
(2) "Detect and report the start and end timestamps of activity events in the video, along with descriptions.",
(3) "Pinpoint the time intervals of activity events in the video, and provide descriptions for each event.",
(4) "Can you compile a list of the activities and their timestamps featured in the video?",
(5) "I need you to scrutinize the video and catalog every event it contains, along with the timestamps."

Prompts for Temporal Referring:
(1) "What is happening from $< start >$ to $< end >$?",
(2) "What is taking place between $< start >$ and $< end >$?",
(3) "What events unfold between $< start >$ and $< end >$?",
(4) "What is happening during the period from $< start >$ to $< end >$?",
(5) "What occurs between $< start >$ and $< end >$?",
(6) "What is going on from $< start >$ to $< end >$?",
(7) "How do things progress from $< start >$ to $< end >$?",
(8) "Can you describe what happens from $< start >$ to $< end >$?",
(9) "Describe the events occurring between $< start >$ and $< end >$.",
(10) "Narrate the actions that unfold from $< start >$ to $< end >$."

Table 13: Prompts used for different tasks.