SlowFocus: Enhancing Fine-grained Temporal Understanding in Video LLM

Ming Nie[1] **Dan Ding**[1] **Chunwei Wang**[2] **Yuanfan Guo**[2] **Jianhua Han**[2] **Hang Xu**[2] **Li Zhang**[1*]
[1]School of Data Science, Fudan University [2]Noah's Ark Lab, Huawei
https://github.com/fudan-zvg/SlowFocus

Abstract

Large language models (LLMs) have demonstrated exceptional capabilities in text understanding, which has paved the way for their expansion into video LLMs (Vid-LLMs) to analyze video data. However, current Vid-LLMs struggle to simultaneously retain high-quality frame-level semantic information (*i.e.*, a sufficient number of tokens per frame) and comprehensive video-level temporal information (*i.e.*, an adequate number of sampled frames per video). This limitation hinders the advancement of Vid-LLMs towards fine-grained video understanding. To address this issue, we introduce the SlowFocus mechanism, which significantly enhances the equivalent sampling frequency without compromising the quality of frame-level visual tokens. SlowFocus begins by identifying the query-related temporal segment based on the posed question, then performs dense sampling on this segment to extract local high-frequency features. A multi-frequency mixing attention module is further leveraged to aggregate these local high-frequency details with global low-frequency contexts for enhanced temporal comprehension. Additionally, to tailor Vid-LLMs to this innovative mechanism, we introduce a set of training strategies aimed at bolstering both temporal grounding and detailed temporal reasoning capabilities. Furthermore, we establish FineAction-CGR, a benchmark specifically devised to assess the ability of Vid-LLMs to process fine-grained temporal understanding tasks. Comprehensive experiments demonstrate the superiority of our mechanism across both existing public video understanding benchmarks and our proposed FineAction-CGR.

1 Introduction

Large language models (LLMs) have garnered significant attention due to their exceptional text understanding capabilities. Building on the strengths of LLMs, video large language models (Vid-LLMs) [15, 18, 42] adapt them to the video modality, extending their reasoning and interactive skills to video data. By training on video-level tasks like captioning and question answering, they establish coarse-grained video-language correspondence and acquire the capabilities of video understanding.

While Vid-LLMs have demonstrated the promising performance in video understanding, they still face a significant challenge. To embed video features into LLMs under computing resource constraints, Vid-LLMs typically need to sparsely sample the original video (*e.g.*, retaining one frame every second) into a collection of low-frequency frames. Subsequently, the token number of each frame are also compressed through a visual adapter like average pooling [16] or Q-former [14]. This leads to a dilemma: Vid-LLMs have to choose between compromised frame-level features and video-level features, each resulting in a reduction of video information. As illustrated in Figure 1 (a), under the premise of a constant total token number, we investigate the performance of Vid-LLMs by

*Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

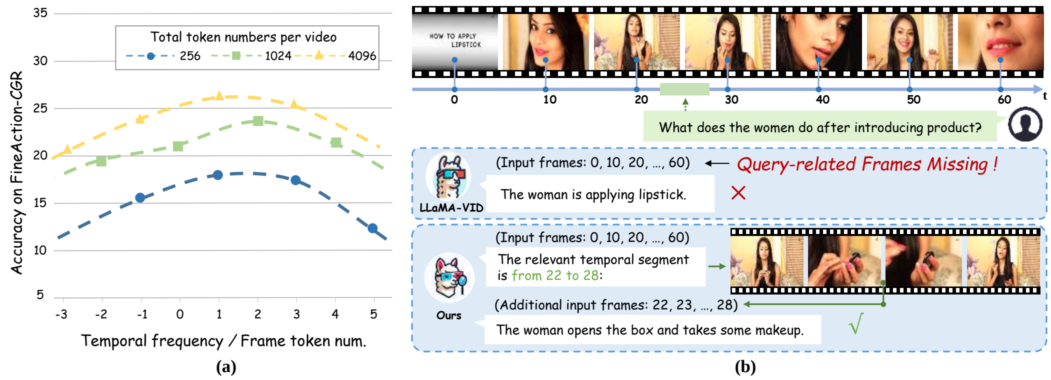

Figure 1: (a) Trade-off between video sampling frequency and frame token number. The horizontal axis represents the ratio (log-transformed) of these two factors. Each curve corresponds to a fixed total number of tokens (*e.g.*, 256 for a 1-minute video). (b) Deficiency of existing Vid-LLMs, such as LLaMA-VID, when facing fine-grained video understanding, and the efficacy of our approach.

dynamically adjusting the sampling frequency and frame token number. The results clearly show that both low sampling frequency and low frame token number lead to performance degradation. Low-frequency sampling results in a sparse image collection as input, omitting crucial temporal details. On the other hand, excessive frame-feature compression degrades the semantic and spatial contexts of each frame. When confronted with fine-grained video understanding tasks, this limitation becomes significantly evident, as depicted in Figure 1 (b). Existing models (*e.g.*, LLaMA-VID [17]) often overlook crucial details early in the input due to low-frequency sampling, leading to inaccuracy.

To address this challenge, we introduce SlowFocus, which is designed to pinpoint relevant temporal segments in response to questions, and subsequently maintains high-quality temporal details to enrich fine-grained video comprehension. For video understanding tasks, we assume that the relevant details are concentrated within one or several clips. SlowFocus initially identifies these segments based on the provided questions. It then densely samples the segmented temporal clips at a high-frequency to extract local temporal features highly pertinent to the questions. To effectively model the temporal relationships between frames and capture inter-frame contexts, we propose a specialized temporal encoder and a multi-frequency mixing attention module for enhanced temporal comprehension.

To enhance Vid-LLMs' ability to perform SlowFocus based on mixed frequencies, we propose a set of training and inference strategies to improve their temporal localization and fine-grained temporal reasoning capability. Following VTimeLLM [10], we fine-tune our Vid-LLM in the second stage on dense video captioning and temporal grounding tasks. This process enhances the model's ability to predict discrete bins that define the relevant temporal segments. In the third stage, we adapt the Vid-LLM to the SlowFocus mechanism for high-quality, fine-grained temporal-related tasks, which enables our model to reason precisely based on high-frequency temporal details.

In addition to the scarcity of methods for fine-grained video understanding, existing benchmarks [37, 40] also fall short in providing adequate challenges for specific temporal-related tasks. To bridge this gap and evaluate our proposed framework, we introduce FineAction-CGR, a newly dedicated benchmark that focuses on fine-grained video understanding, especially reasoning tasks based on temporal details. Our method demonstrates superior performance on the proposed benchmark, offering a promising solution to high-quality video understanding.

The contributions of this paper are summarized as follows: **(i)** We introduce SlowFocus, a straightforward yet effective framework designed to resolve the prevalent trade-off in existing Vid-LLMs between capturing limited frame-level details and overarching video-level contexts. SlowFocus adeptly maintains high-frequency local details alongside low-frequency global contexts, facilitating the identification of pertinent temporal segments and precise reasoning on video contents. **(ii)** We present a novel training strategy specifically designed to enhance the temporal localization abilities of Vid-LLMs, and seamlessly adapt them to our newly proposed SlowFocus approach. **(iii)** We establish a comprehensive new benchmark and carry out extensive experiments to rigorously assess the fine-grained video understanding capabilities of Vid-LLMs. The empirical evidence strongly indicates that our SlowFocus approach significantly outperforms existing models, particularly in tasks requiring detailed temporal understanding and reasoning within videos.

2 Related works

Vision large language models. Researchers have made significant efforts to enable Large Language Models(LLMs) to comprehend visual information. BLIP-2 [14] aligns vision-language representation with a lightweight Querying Transformer by concept of Q-Former. MiniGPT-4 [45] aligns detailed image descriptions with advanced LLM which significantly enhances its multi-modal abilities. Exploring diverse multi-modal instruction-following data, LLaVA [20] has demonstrated impressive multi-model chat abilities. Recent LLMs, like Kosmos-2 [26] and VisionLLM [33], probed into more specific aspects of image comprehension, including referring and grounding [27], remarkably enhancing the capability to describe intricate image details.

Video large language models. The exploration of LLMs' potential has been extended from images to videos, which contributes to emergence of Video LLMs. VideoChat [15] combines fine-tuning and LLM-based video agents and is fine-tuned using a specially designed video-centric instruction dataset. Video-ChatGPT [23] proposes a novel human assisted and semi-automatic annotation framework for generation high quality instruction data for videos. Video-LLaMA [42], trained on vision-language branch and audio-language branch separately with same visual data and process, has demonstrated impressive abilities in understanding both visual and auditory content in videos. Video-LLaVA [18] learns united visual representation by alignment before projection and conducts joint training on images and videos simultaneously. With efforts, Video LLMs have exhibited powerful task-handling capabilities in downstream tasks, like text-video retrieval and video captioning. However, these models remain limited in ability to comprehend fine-grained content. To solve this problem, we introduce a model with powerful capability of fine-grained video understanding.

Fine-grained video understanding. Comprehending videos in fine-grained aspects requires precisely locating and understanding specific events within a video. It is roughly divided by previous works [10] into temporal grounding [2, 7] and dense video captioning [13, 31] tasks. Temporal grounding demands the model to precisely identify start and end timestamps of video segment according to a given text query. Dense video captioning requires both temporal localization and captioning for all events within an untrimmed video. Both tasks are constrained by the labor-intensive nature of annotation, resulting in relatively small datasets. Moreover, fine-grained video understanding demands a deep comprehension of how objects change and interact over time, particularly at a detailed level. This complexity necessitates temporal reasoning tasks to effectively analyze these dynamics. However, current research in this field falls short of adequately addressing these requirements. To solve this problem, our proposed large-scale dataset provides mass temporal annotations and captions in different dimensions, supplying a wealth of information for training in temporal video grounding and reasoning tasks.

3 Method

In this section, we begin with a preliminary overview of video LLMs (Vid-LLMs) in Section 3.1. Following that, we offer a comprehensive explanation of our proposed SlowFocus in Section 3.2, followed by a detailed introduction to our innovative training strategy in Section 3.3.

3.1 Preliminary

Contemporary Vid-LLMs typically feature a modular architecture, which includes a visual encoder E_V, a series of visual adapters Q, and a large language model L. For a given video $V = \{V(t) \in \mathbb{R}^{H\times W\times 3} | t = 0, ..., T\}$ that consists of T frames, along with its associated question q, Vid-LLMs generally perform downsampling on the original video V at a fixed interval M_L. This process results in sparsely distributed frames V_L:

$$V_L = \{V(M_L t) \in \mathbb{R}^{H\times W\times 3} | t = 0, ..., T\}, \tag{1}$$

where $M_L \gg 1$. Consequently, only a very limited number of frames ($T/M_L \ll T$) are selected as the actual input for the Vid-LLM, a technique we refer to as low-frequency sampling in this study.

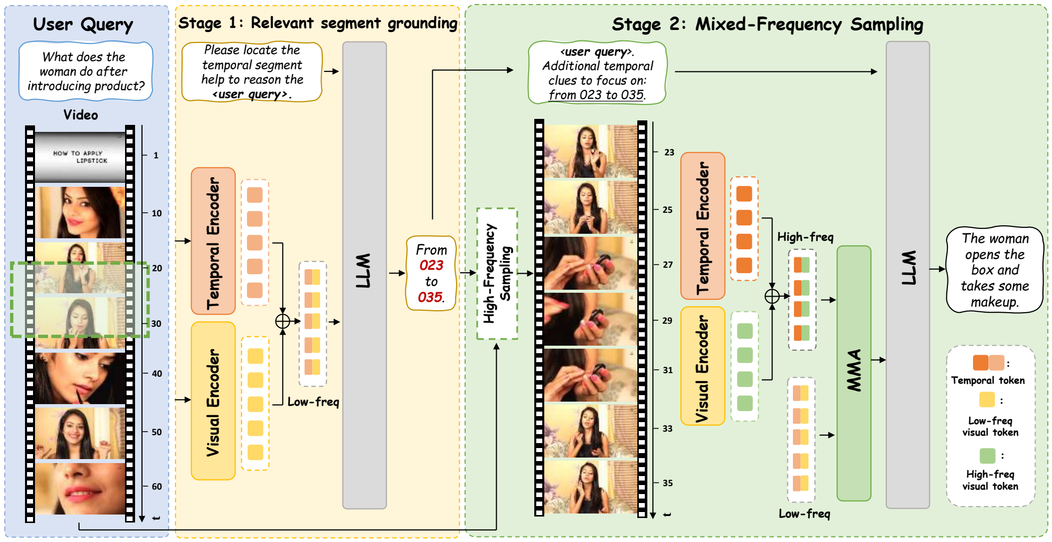

Figure 2: The framework of SlowFocus. We initially identify the relevant temporal segments based on the given query. Subsequently the high-frequency sampling is performed on these segmented clips. Combined with low-frequency sampling across the entire video, our SlowFocus mechanism maintains mixed-frequency visual tokens to accurately answer the query.

Subsequently, the visual encoder processes the downsampled frames V_L and encodes them into a series of visual tokens denoted as $z_L = E_V(V_L)$. These visual tokens are then transformed to align with the embedding space of the language model through the visual adapter Q, resulting in aligned visual tokens $h_L = Q(z_L)$. Concurrently, the input text query q is encoded into linguistic tokens h_q by the textual encoder. These visual and text tokens are concatenated into a unified sequence $[h_L, h_q]$, which then serves as the input for the decoder component of the large language model L. The model leverages this combined representation to infer the appropriate answer $ans = L([h_L, h_q])$, showcasing its ability to perform cross-modal reasoning and respond to human queries.

Although this paradigm is well developed in the field of video understanding, it encounters significant limitations in tasks requiring fine-grained temporal reasoning. By retaining only a small, discontinuous subset of frames (t, where $t \ll T$), substantial information loss can occur, as depicted in Figure 1 (b). Fine-grained temporal video understanding demands that the model concentrate on one or more specific temporal intervals, which could be potentially brief. However, the low-frequency sampling method predominantly captures overarching information while neglecting crucial local details, leading to inaccuracies such as incorrect responses or hallucinations. To address these shortcomings, we develop the SlowFocus strategy.

3.2 SlowFocus

To improve Vid-LLMs with the SlowFocus mechanism, we first expand the traditional training and inference paradigm by introducing Mixed-Frequency Sampling (MFS). Subsequently, we explicitly model the temporal relationships among the sampled frames. Finally, we enhance the visual tokens with our newly proposed multiple-frequency mixing attention, designed to capture long-term contexts. We now provide a detailed elaboration on each of these components.

Relevant segment grounding. To better mimic human cognition in our Vid-LLM, we transform its inference paradigm into a multi-round dialogue format that integrates both the query and temporal awareness elements. The methodology of this transformation is illustrated in Figure 2. Initially, we sample the entire video at a fixed interval to capture the low-frequency frames V_L, which are then encoded into visual embeddings h_L. Following that, we reformat the original question q into temporal grounding questions q_1, such as "*<video>\nPlease provide the temporal segment help to reason the question: <question>*", where *<question>* refers to the original question q.

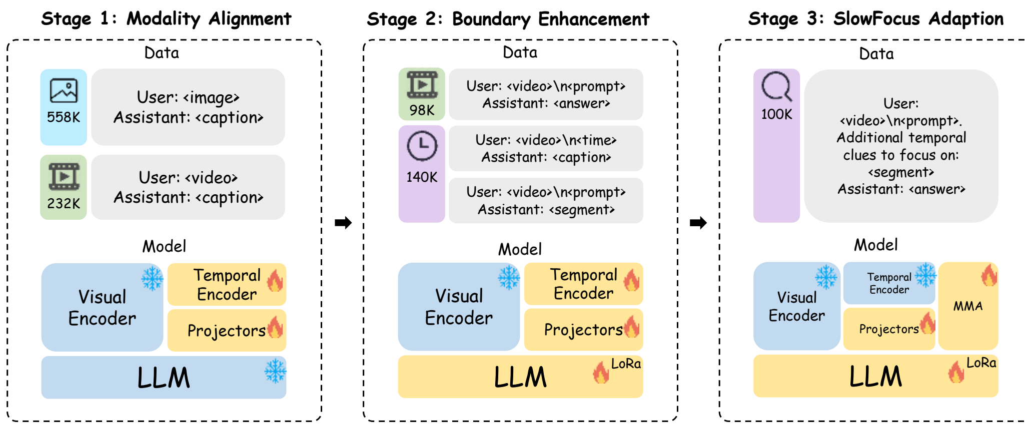

Figure 3: The training strategy of SlowFocus, including data distribution and parameter updating in each stage. `<image>` and `<video>` denote the tokens for image and video, respectively.

With the augmented question q_1, we direct the Vid-LLM to identify the relevant temporal segments τ within the video that are pertinent to the original question q. This is accomplished by providing the model with low-frequency frames containing global information:

$$\tau = L([h_L, h_{q_1}]), \tag{2}$$

where h_{q_1} represents the textual embedding of question q_1 and $\tau \subset [0, T]$. The identified segment τ is designed to encompass query-related details that are crucial for addressing the question q.

Mixed-frequency sampling. Subsequently, we perform dense sampling on the identified segment τ to obtain high-frequency frames V_H:

$$V_H = \{V(M_H t) \in \mathbb{R}^{H \times W \times 3} | t \in \tau\}, M_H = max(\frac{|\tau|}{N_H}, 1). \tag{3}$$

We dynamically adjust the sampling interval M_H based on the temporal length of segment $|\tau|$ and to ensure an adequate number of samples N_H are taken from the relevant segment. The densely sampled frames V_H are then encoded into high-frequency visual tokens h_H by visual encoder.

With the inclusion of local details from high-frequency frames, we augment the initial question q with prompts, such as "*Additional temporal clues to focus on: ...*", to form q_2, as illustrated in Figure 2. We then combine the mixed-frequency visual tokens to predict the final answer:

$$ans = L([h_L, \pi(h_L, h_H), h_{q_2}]), \tag{4}$$

where π is the multiple-frequency mixing attention, which will be detailed subsequently.

Multiple-frequency mixing attention. Because of the roughness of the low-frequency and the detailed specificity of the high-frequency visual features, simply concatenating these two directly into the language model may not yield optimal results. Fine-grained temporal understanding often requires modeling relationships between multiple events, necessitating the integration of global contexts into high-frequency local features. From this perspective, we introduce the Multiple-frequency Mixing Attention (MMA):

$$\pi(h_L, h_H) = softmax(\frac{h_H h_L^T}{\sqrt{d}}) h_L, \tag{5}$$

where, for simplicity, we omit the process of applying FFN to h_L and h_H. The resulting output is then fed into LLM to predict the textual response, as described in Equation 4.

Temporal relationship modeling. While LLM can implicitly capture temporal relationships among input visual embeddings based on their sequential positioning, it faces difficulties when the visual tokens are not uniformly distributed along the timeline. To address this challenge, we propose a temporal encoder that encodes the relative positions of frames into a set of discretized temporal

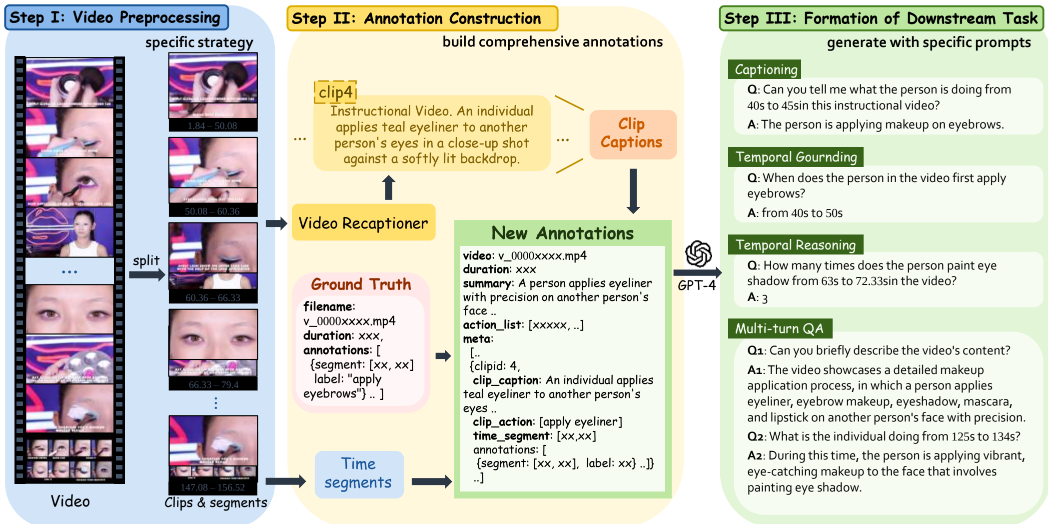

Figure 4: Pipeline of instruction-following data generation. Split the filtered FineAction videos into clips and extract time segments. Use the fine-tuned Video Recaptioner Model and GPT-4V to generate captions for both video clips and full videos. Integrate the ground truth data, new captions, and time segments to create comprehensive annotations. Finally, generate QA pairs for various tasks using different prompts via GPT-4.

tokens $\mathcal{E} \in \mathbb{R}^{N \times C}$, where N represents the discrete temporal space. For frames sampled at multiple frequencies $V(t)$, the corresponding temporal token is ϵ_i, where $i = \lfloor N * t/T \rfloor$. These temporal embeddings are then incorporated into the visual tokens by directly adding them to the frame features.

3.3 Training strategy

Considering training efficiency, we introduce a novel training procedure which is structured into three distinct phases in this work: (i) pre-training for modality alignment, (ii) fine-tuning for enhancing temporal grounding, and (iii) SlowFocus adaption, as illustrated in Figure 3. We now provide detailed elaborations on the training strategies and datasets employed for each of these stages.

Modality alignment. In the pre-training stage, our primary focus is to optimize the visual adapters and temporal encoders, while the visual encoder and language model are frozen to ensure that the visual features align effectively with the language space. Following the approach used in LLaMA-VID [16], we utilize the image-text LCS-558K dataset from LLaVA [20], along with 232K video-caption samples from the WebVid 2.5M [3].

Boundary enhancement. After the pre-training stage, the Vid-LLM gains proficiency in processing visual information. During the fine-tuning stage, we focus on enhancing the model's ability to comprehend sequences of video frames, thereby improving its temporal localization capabilities. In alignment with practices from VTimeLLM [10], we employ the InternVid-10M-FLT dataset [34], which is specifically designed for temporal-awareness training. The tasks within this dataset include: (i) dense video captioning, which requires detailed descriptions of all events along with their corresponding timestamps, and (ii) segment captioning and temporal video grounding, which involve generating descriptions based on timestamps or determining timestamps based on descriptions. Throughout this stage, we continue to train the visual adapters and temporal encoders. Additionally, the LLM is further trained using LoRA [9] to refine its capabilities.

SlowFocus adaption. Following the fine-tuning stage, our model has demonstrated the ability to comprehensively understand activities within the video and accurately align them with their respective timestamps. In the final stage, to further enhance multi-modality comprehension and integration with the MMF mechanism, we construct an instruction-tuning dataset using samples from ActivityNet Captions [13] and FineAction [21]. We transform the annotations into over 100K high-quality QA dialogues, which will be detailed in the subsequent section. In alignment with our SlowFocus mechanism, during training with fine-grained captioning and reasoning tasks, the model is provided with the ground-truth temporal segments τ_{GT} and subjected to mixed-frequency sampling. In this

Method	LLM	LoRA	Temporal grounding				Temporal captioning				Temporal reasoning	
			mIoU	R@0.3	R@0.5	R@0.7	B	M	R	C	Acc	Score
VideoLLaMA [42]	Vicuna-7B	✗	0.17	0.25	0.11	0.00	0.06	0.09	0.08	0.13	8.75	0.53
Video-ChatGPT [23]	Vicuna-7B	✗	0.06	0.10	0.01	0.00	0.15	0.12	0.10	0.21	13.93	0.79
LLaMA-VID [16]	Vicuna-7B	✗	0.35	0.52	0.17	0.03	0.16	0.12	0.11	0.23	15.65	0.87
VTimeLLM [10]	Vicuna-7B	✓	27.69	32.83	24.26	21.87	0.05	0.09	0.08	0.12	9.96	0.54
LLaMA-VID†	Vicuna-7B	✓	22.38	26.17	19.47	17.53	0.23	0.20	0.37	1.03	24.81	1.26
Ours	Vicuna-7B	✓	**66.68**	**85.80**	**73.01**	**56.25**	**0.66**	**0.41**	**0.70**	**3.27**	**53.10**	**2.78**

Table 1: Main results on FineAction-CGR benchmark. The column *LoRA* represents whether the LLM is fine-tuned fully or using LoRA. †: Model is re-trained on the stage 3's data. B: B@4. M: METEOR. R: ROUGE. C: CIDEr.

Method	LLM	LoRA	MSVD-QA		MSRVTT-QA		ActivityNet-QA		Video-based generative performance				
			Acc	Score	Acc	Score	Acc	Score	Correctness	Detail	Context	Temporal	Consistency
FrozenBiLM [38]	DeBERTa-V2	✗	32.2	-	16.8	-	24.7	-	-	-	-	-	-
VideoLLaMA [42]	Vicuna-7B	✗	51.6	2.5	29.6	1.8	12.4	1.1	1.96	2.18	2.16	1.82	1.79
LLaMA-Adapter [43]	LLaMA-7B	✗	54.9	3.1	43.8	2.7	34.2	2.7	2.03	2.32	2.30	1.98	2.15
VideoChat [15]	Vicuna-7B	✗	56.3	2.8	45.0	2.5	26.5	2.2	2.23	2.50	2.53	1.94	2.24
Video-ChatGPT [23]	Vicuna-7B	✗	64.9	3.3	49.3	2.8	35.2	2.7	2.40	2.52	2.62	1.98	2.37
LLaMA-VID [16]	Vicuna-7B	✗	69.7	3.7	57.7	3.2	47.4	3.3	**2.96**	3.00	3.53	2.46	2.51
LLaMA-VID	Vicuna-7B	✓	69.2	3.4	57.1	2.9	45.6	3.3	2.87	2.89	3.28	2.13	2.47
Ours	Vicuna-7B	✓	**70.1**	**3.9**	**58.3**	**3.5**	**48.4**	**3.6**	2.95	**3.03**	**3.61**	**2.54**	**2.60**

Table 2: Comparison with existing methods on coarse-grained video understanding benchmarks. Our method achieve on par performance with state-of-the-art models.

stage, we freeze all the visual adapters and attention modules, only training the language model as well as the MMA module. The parameters of visual encoder are frozen all over the stages.

4 FineAction-CGR benchmark

In addition to the scarcity of previous methods for fine-grained video understanding, existing benchmarks [37, 40] do not adequately challenge specific temporal-related tasks. To address this gap and evaluate our proposed framework, we introduce FineAction-CGR, a comprehensive and high-quality instruction-following benchmark designed for evaluating the capabilities of Video LLMs in fine-grained video understanding. FineAction [21] is a large-scale and fine-grained video dataset encompassing diverse action categories with detailed annotations of action instances and time segments. For the purpose of SlowFocus adaption and model evluation, we divide the FineAction dataset into training and testing sets based on videos, allocating 75% to the training set and 25% to the test set, ensuring these is no overlap between them. Building on this foundation, we probe time information and content information from FineAction and expand it into multi-tasks. This section outlines the data construction procedures, which consist of three main steps: (i) video preprocessing, (ii) annotation construction, and (iii) formation of downstream tasks. More details are included in the appendix.

Video preprocessing. As depicted in Step I of Figure 4, we employ a modified two-stage splitting algorithm from Panda70M [5] to segment videos into clips, thereby providing raw materials for fine-grained information at the clip level. This preprocessing step yields 62,912 video clips.

Annotation construction. Step II in Figure 4 outlines our approach to creating new annotations, which include time segments, captions, and action labels. Given the high cost of using GPT-4V [1] for generating large-scale, clip-level captions, we generate detailed captions for entire videos with GPT-4V and then use a fine-tuned Video Recaptioner Model to produce large-scale, clip-level captions. By integrating action labels and time segments from the ground truth with clip segments and the generated captions, we create comprehensive new annotations.

Formation of downstream tasks. Utilizing the new annotations, we design four types of video instruction-following tasks, as illustrated in Step III of Figure 4: (i) segmented captioning, (ii) temporal video grounding, (iii) temporal video reasoning and (iv) multi-turn QA. Detailed information on construction procedures, task formats, prompts, and case studies can be found in Appendix C- E.

Method	LLM	LoRA	Global mode Acc	Global mode Score	Breakpoint mode Acc	Breakpoint mode Score
VideoChat [15]	Vicuna-7B	✗	57.8	3.00	46.1	2.29
VideoLLaMA [42]	Vicuna-7B	✗	51.7	2.67	39.1	2.04
Video-ChatGPT [23]	Vicuna-7B	✗	47.6	2.55	48.0	2.45
MovieChat [28]	Vicuna-7B	✗	62.3	3.23	48.3	2.57
Ours	Vicuna-7B	✓	58.6	3.14	48.1	2.53

Table 3: Comparison with existing methods on MovieChat-1K.

Method	LLM	LoRA	Acc
FrozenBiLM [39]	DeBERTa [8]	✗	26.9
VIOLET [6]	-	✗	19.9
InternVideo [35]	-	✗	32.1
LLoVi-7B [41]	Llama2-7B [29]	-	34.0
Vamos [32]	GPT-4	-	36.7
LangRepo-12B [12]	Mixtral [11]	-	41.2
Ours	Vicuna-7B	✓	39.7

Table 4: Comparison with existing methods on EgoSchema.

Sampling strategy	Temporal encoder	MMA	Temporal grounding mIoU	R@0.3	R@0.5	R@0.7	Temporal reasoning Acc	Score
V_L	✗	✗	32.54	41.67	34.52	25.84	30.25	1.57
$V_L + V_H, N_H = 10$	✗	✗	34.81	44.65	36.49	27.90	39.12	2.02
$V_L + V_H, N_H = 10$	✓	✗	57.26	73.47	62.28	47.16	46.37	2.37
$V_L + V_H, N_H = 20$	✓	✗	64.92	83.25	70.19	55.13	51.68	2.66
$V_L + V_H, N_H = 40$	✓	✗	65.03	82.96	70.23	56.01	51.59	2.62
$V_L + V_H, N_H = 20$	✓	✓	66.68	85.80	73.01	56.25	53.10	2.78

Table 5: Components analysis. V_L means only low-frequency frames are sampled. $V_L + V_H$ represents performing mixed-frequency sampling and N_H denotes the number of high-frequency frames.

Evaluation protocol. To assess the capability of video language model in comprehending fine-grained video understanding tasks, we integrate multiple evaluation metrics in comprehensively evaluate the model's performance against our proposed benchmark. For temporal grounding task, we calculate the Intersection over Union (IoU) between the temporal segments generated by the model and the corresponding ground truth, and we report meanIoU (mIoU) metric. For segmented captioning, we employ commonly accepted captioning-based metric, including BLEU [25], METEOR [4], ROUGE [19] and CIDEr [30]. For temporal reasoning task, we utilize GPT-4 [1] to evaluate the accuracy and score of the generated answers.

5 Experiments

5.1 Experiment setup

In our experiments, we implement LLaMA-VID [17] as baseline and utilize Vicuna-7B v1.5 [44] as our foundational LLM. More implementation details are provided in the appendix. We adjust the resolution of input videos to 224 $\times$ 224, and each frame is condensed into 64 tokens. The low-frequency sampling interval M_L is set to match the original video fps, ensuring one frame is sampled every second. We define the dense sampling number N_H as 20 and the size of temporal token space N as 1000. The AdamW [22] optimizer is applied with a cosine learning rate and decay and a warm-up period. We train our Vid-LLM for three stages. During the initial pre-training stage, the learning rate is set to 1×10^{-3}. For the subsequent fine-tuning stages, the learning rate is adjusted to 2×10^{-4}. Additionally, the LoRA parameters are configured with $r = 64$ and $alpha = 128$. All experiments are conducted on 8 V100 GPUs.

5.2 Main results

Results on fine-grained video understanding. We evaluate the performance of existing Vid-LLMs on our proposed fine-grained video understanding benchmark FineAction-CGR, as detailed in Table 1. Further specifics are available in the appendix. Our method significantly outperforms other counterparts, achieving a mIoU of 66.68 for temporal grounding and an accuracy of 53.10% for the reasoning task. Notably, most other models, with the exception of VTimeLLM [10], exhibit subpar performance in both the temporal grounding and reasoning tasks on the benchmark. We propose several possible explanations for this observation: (i) These models lack sensitivity to precise time

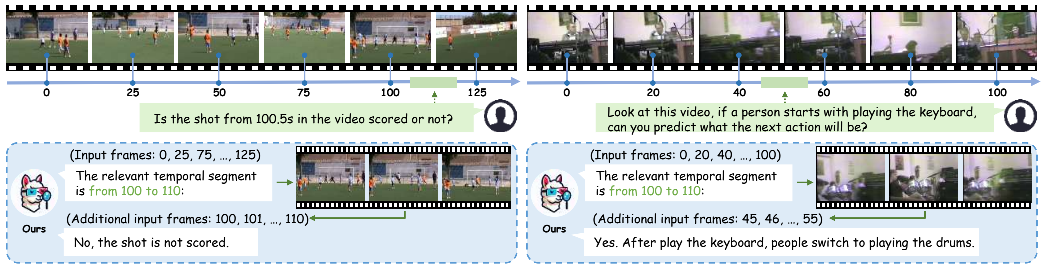

Figure 5: Qualitative examples. Our proposed SlowFocus can effectively leverages the segmented temporal clues to accurately answer the posed question.

boundaries, making it challenging for them to accurately predict temporal segments. (ii) Due to their reliance on low-frequency sampling, these models struggle to capture fine-grained temporal details, which adversely affects their performance on tasks requiring fine-grained temporal reasoning.

We also fine-tune the baseline using stage 3. To ensure fairness, the implementation details for baseline fine-tuning are kept consistent with those of our method. The baseline's performance improves because stage 3 includes tasks focused on temporal grounding and is specifically designed for fine-grained analysis. Furthermore, the remaining performance gap further supports our explanations.

Stages	Temporal grounding mIoU	R@0.3	R@0.5	R@0.7	Temporal reasoning Acc	Score
1	0.11	0.29	0.00	0.00	7.13	0.51
1+2	51.67	63.29	57.84	45.81	20.34	1.04
1+3	28.58	36.72	33.25	24.22	32.27	1.63
1+2+3	66.68	85.80	73.01	56.25	53.10	2.78

Table 6: Ablation of training stages. The first stage itself yields poor result. Integrating these stages together results in the optimal performance.

Fps	N_V	Temporal grounding mIoU	R@0.3	R@0.5	R@0.7	Temporal reasoning Acc	Score
64	1	24.15	33.05	24.84	25.71	34.58	1.80
16	4	45.19	58.67	50.17	38.18	45.36	2.35
4	16	61.01	76.54	65.12	52.63	52.17	2.67
1	64	66.68	85.80	73.01	56.25	53.10	2.78

Table 7: The results of dynamically adjusting sampling frequency (fps) and frame token number N_V when maintaining constant total token number.

Results on coarse-grained video-based benchmarks. In Table 2, we present a comparative evaluation of our method against various state-of-the-art methods across three zero-shot video-QA benchmarks: MSVD-QA [36], MSRVTT-QA [37], and ActivityNet-QA [40]. We also conduct experiments on the video-based generative performance benchmark [23]. The results demonstrate that the proposed SlowFocus mechanism not only enhances fine-grained video understanding but also delivers competitive performance in coarse-grained video understanding tasks, achieving results on par with state-of-the-art models.

Results on long video benchmarks. To further investigate the effectiveness of the proposed method on more challenging scenarios, we provide evaluations on long video benchmarks, including MovieChat-1K [28] and EgoSchema [24]. As shown in Table 3, we evaluate our method on MovieChat-1K. The results show that although our method is not specifically trained on long video benchmarks (in contrast, MovieChat [28] has undergone targeted training for long videos), it still achieves competitive results (58.6% accuracy in global mode and 48.1% in breakpoint mode).

Additionally, we also conduct experiments on EgoSchema [24] benchmark, as detailed in Table 4. The results further demonstrate that, despite not being specifically trained on long video datasets, our method still achieves competitive performance.

Qualitative results. Figure 5 illustrates the qualitative results of our method on different videos. Our SlowFocus comprehensively analyzes the entire video, accurately identifies relevant temporal details based on the posed question, and provides precise answer within the context of videos.

5.3 Ablations

In this section, we provide detailed ablation studies of our method conducted utilizing Vicuna-7B as the foundation model.

Component-wise analysis. We first investigate the impact of each proposed component in Table 5. Importantly, we observe that the baseline model, which relies solely on low-frequency frames (V_L), achieves limited performance, with an mIoU of 32.54 and an accuracy of 30.25. This highlights the

N	Temporal grounding				Temporal captioning				Temporal reasoning	
	mIoU	R@0.3	R@0.5	R@0.7	B	M	R	C	Acc	Score
0.1K	43.81	62.26	51.64	33.18	0.47	0.33	0.49	2.15	39.41	2.20
1K	66.68	85.80	73.01	56.25	0.66	0.41	0.70	3.27	53.10	2.78
10K	63.74	86.19	71.05	55.17	0.66	0.40	0.71	3.24	52.59	2.71

Table 8: Ablation study on temporal token space N.

challenges faced by current Vid-LLMs in addressing fine-grained temporal tasks using only low-frequency sampling. Additionally, the results in the second row indicate that the inclusion of MFS significantly improves reasoning capabilities, resulting in an accuracy increase of +8.87. Furthermore, the temporal encoder, which models temporal relationships effectively, boosts both grounding and reasoning capabilities, with an increase of +22.45 in mIoU and +7.25 in accuracy. Increasing the high-frequency sampling number N_H from 10 to 20 leads to a noticeable performance gain (+7.66 in mIoU and +5.31 in accuracy), although further increasing N_H to 40 offers only marginal benefits. Lastly, the MMA module contribute an accuracy enhancement of +1.42.

Necessity of fine-tuning strategy. We also analyze the impact of each training stage, as reported in Table 6. Directly evaluating the pre-trained model after stage 1 yields poor performance. Based on necessary pre-training stage 1, when only utilizing stage 2 for boundary enhancement, there is a significant improvement by 51.56 in mIoU in temporal grounding ability. Furthermore, implementing only stage 3 for SlowFocus adaptation boosts the performance in temporal reasoning by 25.14 in accuracy. Integrating these stages results in the highest performance in both mIoU and accuracy metrics, highlighting the irreplaceability of our proposed comprehensive training strategy.

What contributes more to fine-grained video understanding? In Table 7, we conduct further experiments to explore how changes in sampling frequency and the number of frame tokens influence our method. For meaningful comparison, we maintain constant total token numbers while gradually decreasing the sampling frequency. The results indicate that within proposed SlowFocus, performance improves with an increase in frame tokens and remains relatively unaffected by a decrease in global sampling frequency. This demonstrates the efficacy of our approach in high-frequency sampling and effectively alleviates the trade-off dilemma between sampling frequency and frame tokens.

Ablation on discretized temporal space. We investigate the influence of temporal token space N in Table 8. The results demonstrate that when the token space increase from 0.1K to 1K, a significant improvement (+22.87 in mIoU and +13.69 in accuracy) occurs. While further increasing N from 1K to 10K, the performance drops instead.

6 Conclusion and limitations

Conclusion. We introduce SlowFocus, a straightforward yet potent mechanism that significantly enhances fine-grained video understanding. Our approach improves the temporal localization capabilities of Vid-LLMs, enabling them to precisely identify relevant temporal segments based on the given query. Moreover, with our newly developed temporal encoder and multi-frequency mixing attention module, our method effectively models the temporal relationships among frames and captures inter-frame context. Demonstrating superior performance on our newly established fine-grained video understanding benchmarks, we hope that our work can propel the development of Vid-LLMs in achieving advanced video understanding.

Limitations. Despite significant advancements made in this work to enhance Vid-LLM's access to high-frequency temporal details and its temporal reasoning capabilities, challenges persist due to the limited existing research on maintaining high resolution in video. Consequently, our method may still encounter inaccurate predictions stemming from the ambiguity of spatial details.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant No. 62106050 and 62376060) and Natural Science Foundation of Shanghai (Grant No. 22ZR1407500).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint*, 2023. 7, 8

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 3

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, 2005. 8

[5] 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, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. *arXiv preprint*, 2024. 7, 14

[6] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint*, 2021. 8

[7] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 3

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020. 8

[9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint*, 2021. 6

[10] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. *arXiv preprint*, 2023. 2, 3, 6, 7, 8, 14

[11] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint*, 2024. 8

[12] Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. *arXiv preprint*, 2024. 8

[13] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 3, 6

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1, 3

[15] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint*, 2023. 1, 3, 7, 8

[16] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint*, 2023. 1, 6, 7

[17] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint*, 2023. 2, 8

[18] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint*, 2023. 1, 3

[19] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 8

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2024. 3, 6, 14

[21] Yi Liu, Limin Wang, Yali Wang, Xiao Ma, and Yu Qiao. Fineaction: A fine-grained video dataset for temporal action localization. *TIP*, 2022. 6, 7

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*, 2017. 8

[23] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint*, 2023. 3, 7, 8, 9

[24] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 2023. 9

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 8

[26] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint*, 2023. 3

[27] Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12977–12987, 2024. 3

[28] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024. 8, 9

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*, 2023. 8

[30] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 8

[31] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018. 3

[32] Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos: Versatile action models for video understanding. *arXiv preprint*, 2023. 8

[33] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 2024. 3

[34] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint*, 2023. 6

[35] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint*, 2022. 8

[36] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, 2017. 9

[37] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2, 7, 9

[38] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 2022. 7

[39] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 2022. 8

[40] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 2, 7, 9

[41] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *arXiv preprint*, 2023. 8

[42] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint*, 2023. 1, 3, 7, 8

[43] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint*, 2023. 7

[44] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 8

[45] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint*, 2023. 3

A Overview

In this appendix we present:

- More implementation details (Section B).
- Data construction details (Section C).
- Data statistics (Section D).
- Prompts used for instruction generation (Section E).
- Discussion on broader impacts (Section F).

B More implementation details

In our work, we employ CLIP-ViT-L-14 for the frozen visual encoder, which accepts image resolutions at 224×224 as input. For visual adapters, we follow LLaVA [20] to train a projector layer, aligning visual features with the pre-trained LLM word embedding.

We also elaborate on how we enable LLM to predict temporal segments during stages 2 and 3 fine-tuning. Following VTimeLLM [10], we utilize the textual format *"from s to e"* to denote a video clip, where *s* and *e* represent the starting and ending points, respectively. These time points are normalized and range from 000 to 999, corresponding to the discretized temporal token space N. During training, Vid-LLM is exposed to supervisory signals as described above, which enable it to develop the capability to accurately locate temporal segments.

C Data construction

Step I: video preprocessing. We split videos into video clips to provide raw materials for fine-grained information in video clip level.

A proper splitting and stitching strategy is necessary. For one thing, the content of each clip should be semantically consistent. Without this consistency, actions within the same video clip obtained by erroneous segmentation can't be inferred reasonably and in chronological order, which negatively impacts the performance of downstream tasks such as action recognition and temporal reasoning. For another, the video clips with insufficient information or that are too short are not suitable for subsequent tasks, such as temporal grounding. Therefore, we adopt a modified two-stage splitting algorithm in Panda70M [5] to obtain semantically consistent video clips.

In first stage, We collect time boundaries of events in the video. Instead of detecting abrupt changes in pixels of adjacent frames, we detect lens switching so that actions in the same video clip can be inferred reasonably. In second stage, we stitch adjacent events based on time boundaries in first stage. Our stitching strategy could be depicted as merging short video clips and that are semantically similar. To be specific, we merge current event with the previous, if one of the following two conditions is met: 1)the video clip duration is less than five seconds, 2) the feature distance between the start frame of the current event and the end frame of the previous event does not exceed 0.1. Finally, we split videos into clips according to time boundaries from second stage.

Our adjusted splitting strategy has two advantages: 1) it preserves the information density of each segment, ensuring that subsequently generated captions are reasonable; 2) it maintains logical temporal consistency of actions within each segment, which facilitates the generation of question-answer pairs in downstream tasks. Following the preprocessing step, we obtained 62,912 video clips.

Step II: annotation construction. As Figure 4 illustrates, new annotations we construct consist of time segments, captions and actions. (1) Action labels are from FineAction ground truth. (2) Time segments cover clip's time segments from above segmentation step and the action time segments from ground truth. (3) Captions cover video captions and clip captions. We first utilzie GPT-4V to generate detailed captions for entire videos. However, due to the cost of GPT-4V for generating large-scale, clip-level video captions, we fine tune a Video Recaptioner Model based on the captions generated by GPT-4V, which is then utilized to generate large-scale, clip-level captions. With actions and time segments from ground truth, we integrate clip segments from splitting stage and captions generated by GPT-4V and Video Recaptioner Model to get new, comprehensive annotations.

Step III: formation of downstream tasks. We design various types of instruction-following tasks with different prompts to comprehensively train and evaluate Vid-LLMs. FineAction-CGR encompasses 4 video tasks: (1) captioning, (2) temporal video grounding, (3) temporal video reasoning and (4) multi-turn QA. A few examples from different tasks are shown in Figure 8-9.

Captioning task needs model to detects what actions occur in given period.

- Action recognition: recognize action in level of clip which may contain a kind of action or several kinds of actions.

Temporal grounding tasks aims to predict the boundary in the video given an instruction in which the start and end time of clip are mentioned to attract model's attention to target segment. We design two types of question to achieve fine-grained temporal localization.

- First/last time grounding: identify the fist/last time a certain action appears.

Temporal reasoning tasks requires model to understand relationship between actions.

- Action sequence reasoning: infer possible action before/after an action over a period of time.

- Count of times: counts times a certain action appears in a clip.

Multi-turn QA task contains multiple rounds of dialogues involving the above three tasks. The problems are progressive from the entire video to the sub-segment.

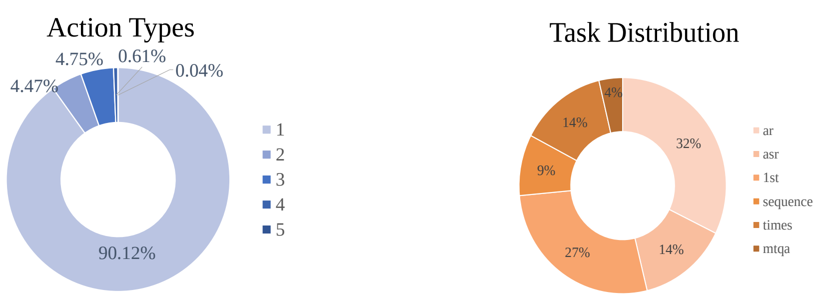

Figure 6: Video Statistics in FineAction-CGR. It contains a diverse distribution of action types and tasks

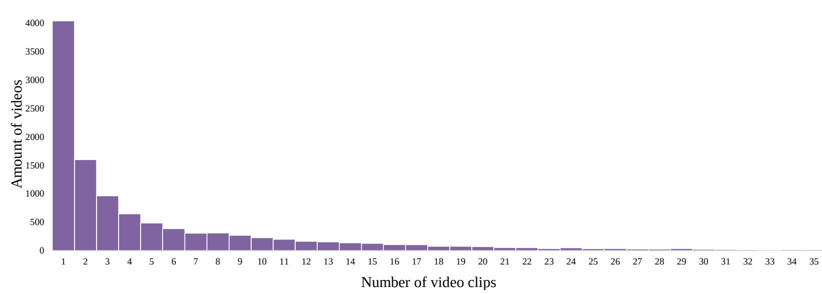

Figure 7: Distribution of clips number in FineAction-CGR

D Data statistics

After filtering, we obtained 11,188 videos with different number of action types. The distribution is illustrated in Figure 6 left. After splitting, videos are segmented into clips. The amount of videos with different clip number is presented in Figure 7, exhibiting a long-tailed distribution. The largest

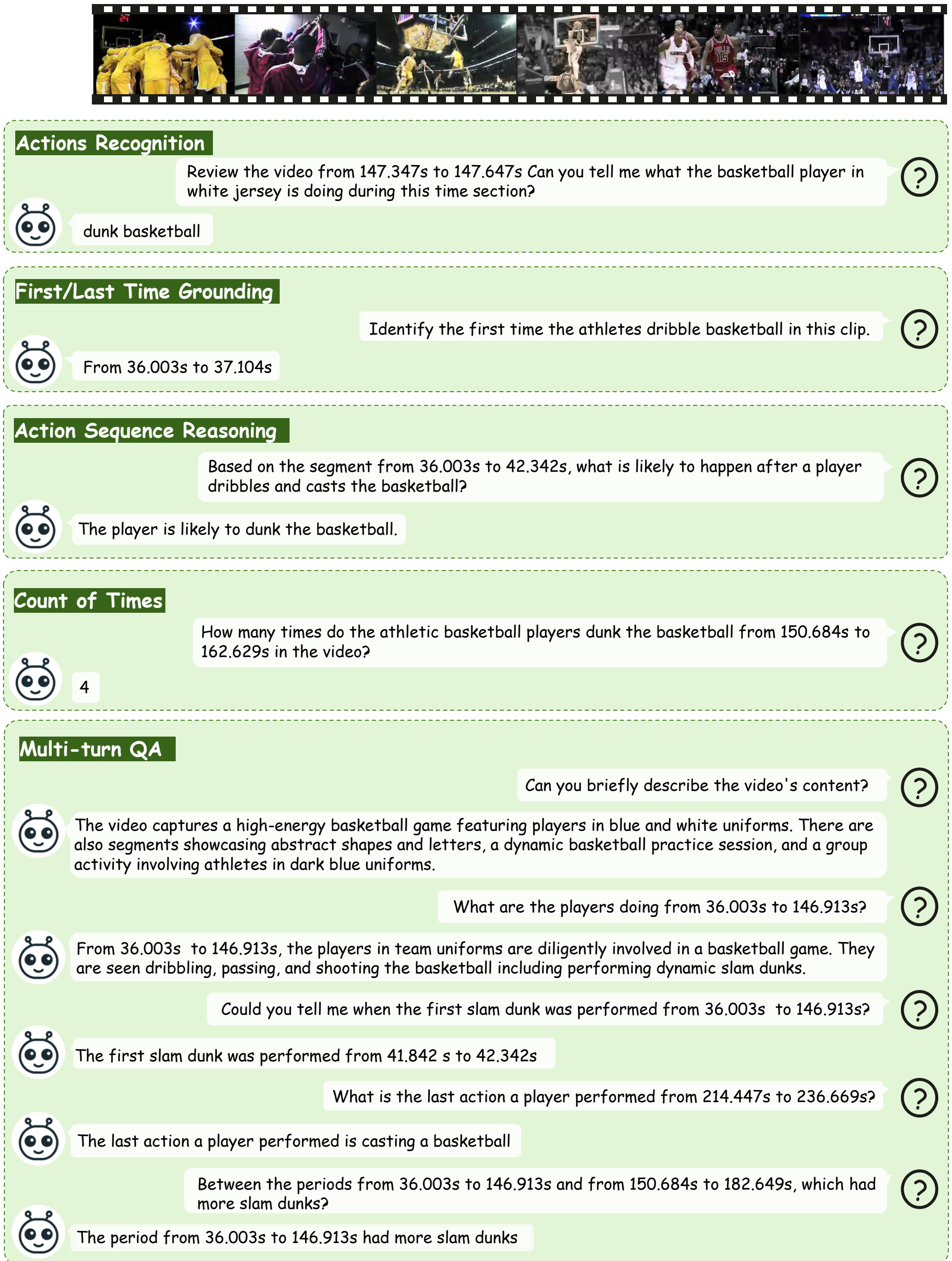

Figure 8: Example of different tasks in FineAction-CGR

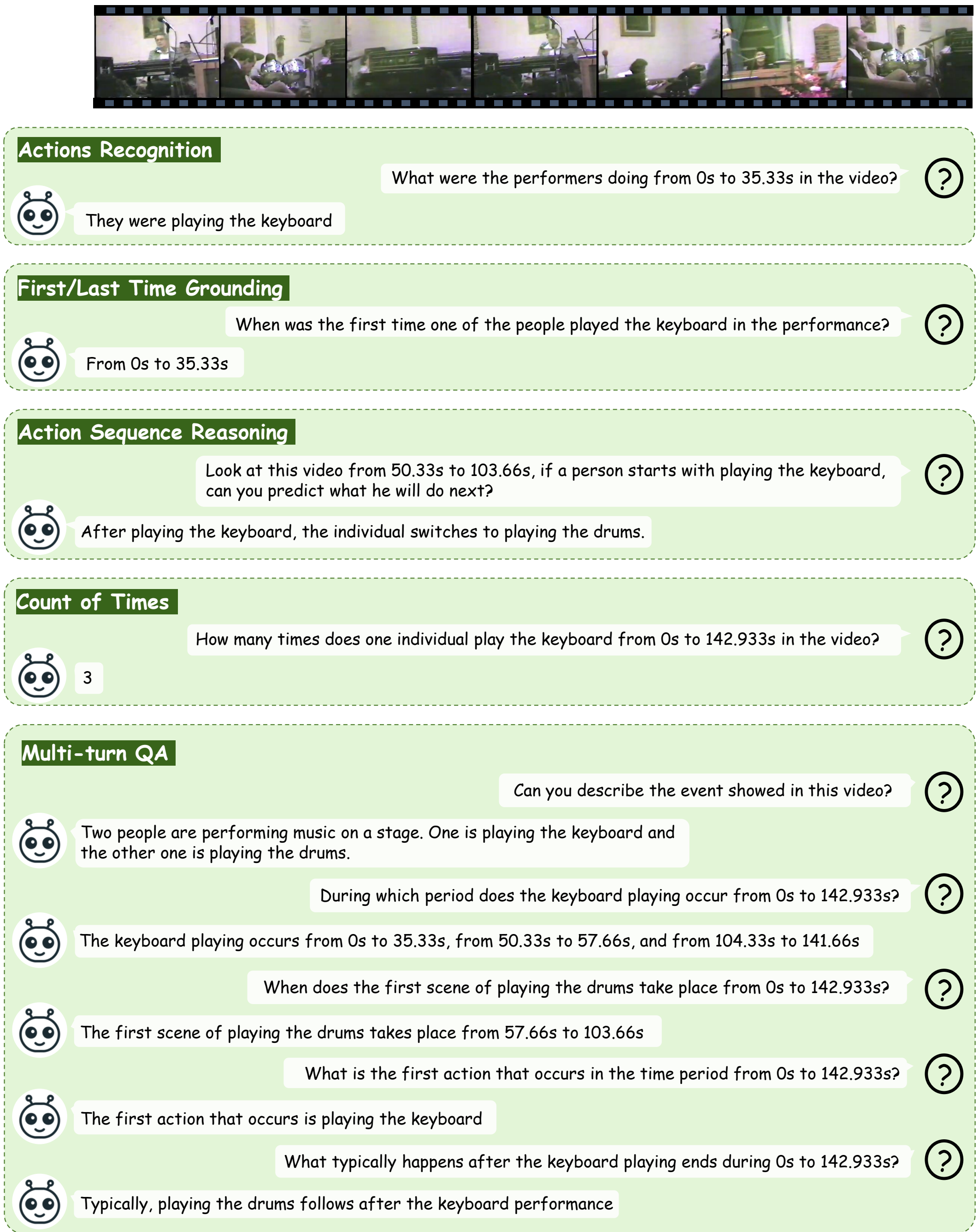

Figure 9: Example of different tasks in FineAction-CGR

clip number is more than 68. The horizontal coordinate in the figure is only displayed to 35. After generation step, we obtain 131,984 pairs of QA. The distribution in different tasks are illustrated in Figure 6 right. Explanation of the legend 1) 'ar': Captioning task where the model needs to recognize a single action in a clip, 2) 'asr': Captioning task where the model needs to recognize multiple actions in a clip, 3) '1st': Temporal grounding task where the model needs to localize the first/last occurrence of a given action in a clip, 4) 'sequence': Temporal reasoning task where the model needs to identify what action happens before or after a given action in a clip, 5) 'times': Temporal reasoning task where the model needs to count the occurrences of a given action in a clip, 6) 'mtqa': Multi-turn QA task where the model engages in a multi-turn question and answer dialogue about the entire video.

E Prompt

Prompts used for generation of different kinds of instruction data are shown in Figure 10-13. Due to page length constraints, we have omitted some in-context examples in certain tasks.

F Broader impacts

The proposed SlowFocus has the potential to greatly enhance the capability of Video LLMs for video content analysis. Therefore it could be beneficial for applications such as video analytics, surveillance and automated content moderation. It could also be used in educational settings to analyze educational videos to provide a better understanding of complex contents in videos for educational purposes.

As for the potential negative impacts, there is a risk that the technology could be misused to analyze private videos and spreading disinformation.

Prompts for Action Recognition

Generate a concise dialogue with questions and answers based on the following video caption, clip captions, action label and time segments with sub segments. All the sub-segments in a segment represent all the time periods in which an action occurs. Below are a few examples:

<Example 1>

[Video Caption]

{A spirited basketball game unfolds on an indoor court, where teams in contrasting uniforms compete, witnessed by an excited audience under the glare of bright lights.}

[Time]

["0:00:00.000", "0:00:20.520"]

[Clip Caption]

{Players in yellow and purple jerseys engage in an intense basketball game, running, passing, and dribbling on a well-lit court, captured in a wide-angle static shot.}

[Action Label]

dribble basketball

[Time Segments with Sub-segments]

["0:00:00.000", "0:00:20.520"]: {[5.014, 5.983], [6.953, 7.989], [9.025, 9.794], [10.997, 11.966], [15.008, 16.245], [17.382, 19.454]}

[Q & A]

[{"from":"human",

"value":"Look at from 15.008 to 16.245 of the video. What the player is doing?"},

{"from":"gpt",

"value":"dribble basketball"}]

Now given the following video caption, clip captions, action label and time segments with sub segments, please generate a question to test whether the deep learning model can give an action label based on the information provided by asking a question like "**What does the person do from a to b in this video?**" or "**What do the people do from c to d in this video?**".

{video_caption}

{time_segment}

{clip_caption}

{action_label}

{time_segment : sub_segments}

The people mentioned must be from [Video Caption] and [Clip Caption]. And the time mentioned should refer to [Time]. The answer must be [Action Label].

Just return the list as your response, like following format:

[{"from":"human", "value":"... from 0 to 20 ..."},

{"from":"gpt", "value":"..."}]

Just return the list.

Figure 10: Prompt for Action Recognition

Prompts for Temporal Grounding

Generate a concise dialogue with questions, answers based on the following clip caption, action labels and their time sgements. The action labels are in chronological order. Below are a few examples:

<Example 1>

[Action Label]

dribble basketball, cast basketball, dunk basketball

[Clip Caption and Time Segment]

["0:00:00.000", "0:00:20.520"] Players in yellow and purple jerseys engage in an intense basketball game, running, passing, and dribbling on a well-lit court, captured in a wide-angle static shot.

[Action Labels and Time Segments]

[0.33, 1.301] dribble basketball,

[1.668, 2.636] cast basketball,

[4.33, 4.638] dribble basketball,

[5.0, 5.639] cast basketball,

[11.011, 12.679] dribble basketball,

[13.013, 13.981] dunk basketball

[Q & A]

[{"from":"human", "value":"Identify the first time the player dribbles basketball in this clip."},

{"from":"gpt", "value":"from 0.33 to 1.301"},{"from":"human", "value":"Identify the last time a player casts basketball in this clip."},

{"from":"gpt", "value":"from 5 to 5.639"}]

Now given the following segment caption, action labels and their time sgements, please generate questions and their answers, please generate three questions related to time and action to test whether the deep learning model can recognition fine-grained action based on the information provided by asking question like "**identify the first/last time the person/people does/do a certain action during this clip**."

{action_label}

{clip_caption} : {time_segment}

{actions_and_segments_list}

Don't mention time segments in questions. The people/person in questions must refer to [Clip Caption]. The actions mentioned in the questions are selected from [Action Label]. The answers should be directly inferred from [Action Labels and Time Segments]. Find the time period of the first/last occurrence of an action based on the time given in the question. Keep the answers brief, with no more than 4 words each.

Just return the list as your response, the format must be like following example:

[{"from":"human", "value":"..."},

{"from":"gpt", "value":"from <s3> to <e3>"},{"from":"human", "value":"..."},

{"from":"gpt", "value":"from <s4> to <e4>"},{"from":"human", "value":"..."},

{"from":"gpt", "value":"from <s5> to <e5>"}]

The value from 'human' is question. The value from 'gpt' is answer. Just return the list.

Figure 11: Prompt for Temporal Grounding

Prompts for Temporal Reasoning

Generate a concise dialogue with questions and answers based on the following clip caption, action labels and time sgements. The action labels are in chronological order. Below is an example:

<Example 1>

[Clip Caption and Time Segment]

["0:00:00.000", "0:00:20.520"] Players in yellow and purple jerseys engage in an intense basketball game, running, passing, and dribbling on a well-lit court, captured in a wide-angle static shot.

[Action Label]

dribble basketball, cast basketball, dunk basketball

[Action Labels and Time Segments]

[0.33, 1.301] dribble basketball,

…

[13.013, 13.981] dunk basketball

[Q & A]

[{"from":"human", "value":"infer from 0.33 to 2.636, what the player would most possibly do after dribble basketball?"},{"from":"gpt", "value":"Cast basketball"},

{"from":"human", "value":"Look at this clip from 0 to 20, what players usually do in a basketball game?"},{"from":"gpt", "value":"Players in a basketball game usually dribble basketball and cast basketball."},

{"from":"human", "value":"Look at this clip from 11.011 to 13.981, a player dribbles basketball. Guess what would he do after that?"},{"from":"gpt", "value":"The player may dunk basketball."},

{"answer_segment":[[1.668, 2.636],[0,20],[13.013.13.981]]}]

Now given the following clip caption, action labels and time segments, please generate questions and their answers, please generate three questions related to time and action to test whether the deep learning model can recognition fine-grained action based on the information provided by asking question like "**Based on one sub-segment of this segment and infer what will most possibly happen after an action**." or "**Look this video from a to b, What's the most possible action before/after this step?**"

time_segment: clip_caption

action_list

actions_segments_list

The actions mentioned in the questions are selected from [Action Label]. The answers should be inferred from the provided time and action sequence in [Action Labels and Time Segments].

Just return the list as your response, please follow below format:

[{"from":"human", "value":" ..."},{"from":"gpt", "value":"..."},{"answer_segment":[<s3>, <e3>]}]

The value from 'human' is question. The value from 'gpt' is answer. The time in every question must be in format 'from ... to ...' and be the maximum period containing question's and answer's actions' but not others. "answer_segment" is a list of time segments pointing to minimum time priod of the answer's action and each is a subset of time in its corrresponding question. Dont't write what you imagine. Just return the list.

Figure 12: Prompt for Temporal Reasoning

Prompts for Multi-turn QA

You are an excellent assistant for generating multi-round conversations. I need you to generate a dialogue with multi-turn questions and answers based on the information of a video. The information is a complete annotation of a video which contains one or more clips in chronological order. It mainly has captions, time segments and actions. Below is an example:

<Example 1>

<Information>

{“summary”: “A fast-paced basketball game … ”, “action_list”: [“cast basketball”, ..],

“meta”: [{ “clipid”: 0, “clip_caption”: “An intense basketball game .. indoor court ..”,

“clip_action”: […], “time_segment”: […],

“annotations”: [{“segment”: […], “label”: “…”},…]} ..]}

[Q & A]

[{"from":"human", "value":"Can you briefly describe the video's content?"}, {"from":"gpt", "value":"The well-lit indoor arena hosts an energetic basketball game..."},

{"from":"human", "value":"What the players is doing from 0 to 11.878?"}, {"from":"gpt", "value":"The players are competing. They dribble and cast basketball on the court."},

{"from":"human", "value":"Please capture the first time a player is dribbling basketball from 0 to 11.878."}, {"from":"gpt", "value":"from 0.033 to 1.969."},

{"from":"human", "value":"What is the last action before dunking basketball from 52.686 to 52.986?"}, {"from":"gpt", "value":"An player casts basketball."},

{"from":"human", "value":"Please infer what players usually do in a basketball game."},{"from":"gpt", "value":"They usually dribble basketball or cast basketball."}, "task":['cp','cp','grd','rsn','rsn']]

Now given the following <Information>, please **generate a dialogue with five pairs of questions and answers to test whether the deep learning model has capability of captioning, temporal grounding and especially temporal reasoning based on the information provided**.

Captioning task contains video captioning, clip captioning and action recognition during a time peiord. **Temporal grounding task** needs to find out target time period according to requirements whose answer is a time period**. Temporal reasoning task** needs to logically infer behavior motive from video, such as time sequence between actions, action sequence in an activity or reasons why people in video do such an action. Questions must be designed according to information provided and must contain all three tasks. You can design temporal reasoning questions different from examples I give you, but don't write what you imagine.

{all_information}

Just return the list as your response, following is format example:

[{"from":"human", "value":"Can you briefly describe ...?"},{"from":"gpt", "value":"..."},

{"from":"human", "value":"... from ... to ...?"},{"from":"gpt", "value":"..."},

{"from":"human", "value":"... from ... to ...."},{"from":"gpt", "value":"from ... to ... ."},

{"from":"human", "value":"... from ... to ...?"},{"from":"gpt", "value":"..."},

{"from":"human", "value":"..."},{"from":"gpt", "value":"..."},"task":['..','..','..','..','..']]

The value from 'human' is question. The value from 'gpt' is answer. 'task' is a list of string in which 'cp' means captioning, 'grd' means grounding, 'rsn' means reasoning. The length of 'task' list are equal to the number of questions. Just return the list.

Figure 13: Prompt for Multi-turn QA

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We make the main claims clearly in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation of this paper in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We discuss the experimental setup in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and benchmark would be released after submission and review. It does include a new open-source benchmark.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details of training and testing are specified in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Experiments in this paper are conducted in multiple iterations, yielding stable results. We reported the average of these results for consistency and reliability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational cost and resources are specified in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Broader Impacts are discussed in Appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We follow the safeguards of base LLM Vicuna-7B v1.5.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models and datasets used in this paper have been properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Benchmark would be released after submission and review. The paper does provide well-structured details of the dataset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.