Dispider: Enabling Video LLMs with Active Real-Time Interaction via Disentangled Perception, Decision, and Reaction

Rui Qian[1*] **Shuangrui Ding**[1*] **Xiaoyi Dong**[1,2] **Pan Zhang**[2]
Yuhang Zang[2] **Yuhang Cao**[2] **Dahua Lin**[1,2] **Jiaqi Wang**[2]
[1] The Chinese University of Hong Kong
[2] Shanghai AI Laboratory
{qr021, ds023}@ie.cuhk.edu.hk

Abstract

Active Real-time interaction with video LLMs introduces a new paradigm for human-computer interaction, where the model not only understands user intent but also responds while continuously processing streaming video on the fly. Unlike offline video LLMs, which analyze the entire video before answering questions, active real-time interaction requires three capabilities: 1) Perception: real-time video monitoring and interaction capturing. 2) Decision: raising proactive interaction in proper situations, 3) Reaction: continuous interaction with users. However, inherent conflicts exist among the desired capabilities. The Decision and Reaction require a contrary Perception scale and grain, and the autoregressive decoding blocks the real-time Perception and Decision during the Reaction. To unify the conflicted capabilities within a harmonious system, we present Dispider, a system that disentangles Perception, Decision, and Reaction. Dispider features a lightweight proactive streaming video processing module that tracks the video stream and identifies optimal moments for interaction. Once the interaction is triggered, an asynchronous interaction module provides detailed responses, while the processing module continues to monitor the video in the meantime. Our disentangled and asynchronous design ensures timely, contextually accurate, and computationally efficient responses, making Dispider ideal for active real-time interaction for long-duration video streams. Experiments show that Dispider not only maintains strong performance in conventional video QA tasks, but also significantly surpasses previous online models in streaming scenario responses, thereby validating the effectiveness of our architecture. The code and model are released at https://github.com/Mark12Ding/Dispider.

*Equal Contribution

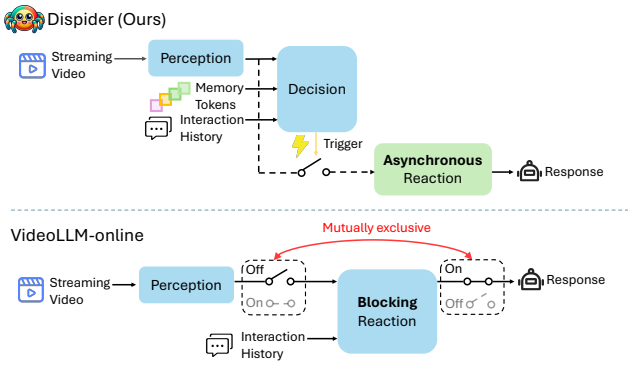

Figure 1. Schematic comparison between Dispider and VideoLLM-online [5]. Our Dispider introduces a disentangled paradigm for active real-time interaction with streaming video. It features a lightweight perception module for continuous monitoring, a decision module to determine when to trigger system interactions, and an asynchronous reaction module for generating detailed responses. In contrast, VideoLLM-online is unable to simultaneously perform streaming perception and response generation because it relies on a single LLM to handle both functions.

1. Introduction

The recent surge in multimodal large language models (MLLMs) has brought considerable attention to video understanding tasks. Given a video, these models aim to accurately comprehend its content and provide responses aligned with human expectations. However, the majority of current video LLMs are designed around an offline setting, where models are required to view the entire video before generating a single response. While effective for certain applications, this offline approach is impractical for real-time, interactive scenarios where users expect continuous feedback as video streams.

VideoLLM-online [5] has been a pioneering effort in ad-

dressing this issue. By leveraging the LLaMA model [33], they enable proactive response updates during streaming video inputs. However, it has a critical limitation: since it uses a single LLM for video processing and response generation, it cannot perform perception and answer reaction simultaneously. The autoregressive nature of next-token prediction in LLMs forces VideoLLM-online to alternate between perception (processing frames) and reaction (generating response), rather than handling both in parallel. This blocking reaction delays the video inputs during responding, reduces responsiveness and hinders its ability to engage in real-time interactions effectively, especially during long-duration video streams that require continuous, uninterrupted processing.

To overcome these limitations, we introduce Dispider, a novel system designed for active, real-time interaction with streaming videos. Dispider disentangles **perception**, **decision**, and **reaction** into asynchronous modules that operate in parallel. As illustrated in Figure 1, the perception module continuously processes streaming video inputs. Meanwhile, the decision module integrates the full interaction history including previous decision tokens and visual information from the perception module. This determines whether to trigger the reaction module, thereby ensuring that detailed, timely responses can be delivered. Unlike VideoLLM-online, Dispider's decision process remains uninterrupted by the asynchronous reaction step, ensuring a fluid and continuous decision-making flow.

Specifically, we design Dispider with a scene-based perception module, a real-time response decision module, and an asynchronous interaction module. In the scene-based perception module, the system dynamically segments the video stream into non-uniform clips based on scene boundaries, ensuring that each segment captures meaningful changes in the visual content. Subsequently, we integrate scene-based features, historical responses, and previous decision tokens into an interleaved sequence. This sequence is then processed by the real-time response decision module, which determines whether the model should generate a response or continue waiting for additional video content.

When an interaction is triggered, the asynchronous interaction module generates context-aware responses without disrupting the ongoing video processing. This asynchronous approach ensures that video analysis and response generation occur in parallel, maintaining the system's real-time performance.

Furthermore, Dispider is trained on a specialized streaming QA dataset designed to simulate real-time interaction scenarios. This training recipe enables the model to effectively handle both instances requiring responses and situations where silence is more appropriate. Consequently, Dispider enhances its ability to interact appropriately across diverse contexts, ensuring timely and relevant responses in dynamic streaming environments.

This disentangled design ensures that the Dispider can provide timely, accurate, and computationally efficient responses, even for long-duration video streams. We evaluate Dispider's performance in real-time video stream interactions (StreamingBench [36]) and show that it significantly outperforms VideoLLM-online [5] in temporal grounding, proactive response generation, and multi-step reasoning. Furthermore, it outperforms conventional offline Video LLMs across long-video benchmarks (EgoSchema [44], VideoMME [20], MLVU [75]) and time-sensitive tasks (ET-Bench [42]). Dispider demonstrated exceptional performance, particularly excelling in temporal reasoning and effectively handling diverse video lengths.

2. Related Work

2.1. Large Language Models

Large Language Models (LLMs) have revolutionized natural language processing, achieving remarkable performance across a wide range of tasks. Early models such as BERT [13] and T5 [52] used masked language modeling for pre-training. The shift to decoder-only models like GPT [50] introduced scalable architectures that enhanced language generation. Recent models, including PaLM [11], LLaMA [56], and GPT-4 [45], continue to push the boundaries with massive parameters and extensive training data. Techniques such as supervised fine-tuning and reinforcement learning from human feedback [2–4, 10, 22, 23, 27, 33, 46, 65] have further improved these models' ability to generate coherent, contextually relevant responses. Inspired by the reasoning capabilities of LLMs, we extend these models to the domain of streaming video understanding, where real-time interaction presents unique challenges.

2.2. Video Large Language Models

The progress of multi-modal LLMs [16, 30, 39, 47, 57, 61, 70] has been significant, particularly with image-based models like InstructBLIP [12], LLaVA [38–40], and Flamingo [1], which integrate vision and language models. Extending this to video introduces challenges in managing both frame sequences and the context length of LLMs. Recent works on video LLMs [6–9, 15, 17, 19, 21, 34, 37, 41, 43, 55, 58–60, 62, 64, 67, 68, 71, 72] present novel strategies for processing long-form videos while maintaining effective reasoning. Notable approaches include TimeChat [53], which emphasizes temporal relationships across video frames, and MovieChat [55], which uses sparse memory to handle long videos. VideoChat [31] adopts a chat-centric approach, integrating video foundation models with LLMs to excel at spatiotemporal reasoning, event localization, and causal inference. Building on

these video LLMs, our work introduces a pipeline designed to process streaming video inputs and generate real-time outputs.

2.3. Streaming Video Understanding

In practice, users typically expect models to operate online and interactively in real-time. This is known as streaming video understanding, which involves processing continuous video streams while ensuring long-term temporal consistency and enabling interactive responses. Despite its practical significance, only a few works have explored this area. VideoLLM-online [5] introduces the LIVE framework for streaming dialogue, but it lacks an efficient module for handling long-term video inputs over extended periods and does not prioritize real-time interactivity. VideoStream [48] proposes memory-propagated encoding for long videos, yet its focus is on offline processing rather than real-time streaming. Similarly, Flash-VStream [69] addresses inference latency and memory efficiency for long video streams but overlooks real-time user interaction. In contrast, our approach centers on developing a real-time visual assistant for streaming video, emphasizing both long-context handling and interactive, real-time responses, thereby bridging the gap left by previous methods.

3. Method

3.1. Problem Formulation

In contrast to previous offline video LLMs, which generate responses only after processing the entire video, our approach operates in real-time by simultaneously processing the video and providing continuous, interactive responses. Specifically, the model actively determines when sufficient information has been observed to provide a complete response, allowing it to produce answers promptly without waiting for the entire video to finish. Following the formulation of VideoLLM-online [5], we formulate this new setting as below.

Given a continuous video stream $V = \{v_1, v_2, \ldots, v_T\}$, where v_i represents the i-th video clip, and a context sequence C_t up to a specific time t (which includes prior vision-language interactions such as historical user queries and assistant responses), our goal is to generate timely and accurate dialogue responses R_t without processing the entire video sequence.

At each time t, the model observes C_t and the video frames up to that time $V_{[0,t]} = \{v_1, v_2, \ldots, v_t\}$. The model must decide whether it has sufficient information to generate a response. We define a decision function π and a reaction function f:

$$a_t = \pi(C_t, V_{[0,t]}) \in \{\text{wait}, \text{respond}\}.$$

If $a_t = \text{respond}$, the model generates a response:

$$R_t = f(C_t, V_{[0,t]})$$

Otherwise, the model keeps silent for more information.

In this work, we implement these functions using a disentangled framework composed of three distinct modules: Perception, Decision, and Reaction. These modules are designed to handle the unique challenges of real-time video understanding and dialogue generation. The Perception module focuses on continuous video monitoring, while the Decision module assesses when to act, and the Reaction module generates responses without waiting for the entire video sequence to finish. This approach allows the system to remain responsive to new information while ensuring that the generated answers are based on the most relevant and up-to-date context.

3.2. Proactive Streaming Video Processing

To enable active real-time response, we propose a proactive streaming video processing approach that dynamically segments the video stream and evaluates whether to generate a response. The system is composed of two key modules: the Scene-based Perception Module and the Real-time Response Decision Module.

Scene-based Perception Module. To ensure efficient processing of long video streams, we begin by adaptively segmenting the video into non-uniform clips based on scene boundaries. This segmentation strategy preserves the structural information of the video, allowing the model to focus on the most informative parts while removing redundancy and maintaining context.

We begin by sampling frames at a regular interval and extract L2-normalized feature embeddings using the pretrained SigLip model [66]. By computing the cosine similarity between these embeddings, we can identify significant visual changes, which indicate potential scene boundaries. These boundaries help divide the video into meaningful segments. To prevent excessively short clips, we introduce an exclusion window around the boundaries, ensuring that the resulting clips are of sufficient length to contain relevant information.

Each clip v_i is then processed by the video encoder to produce the clip-wise feature representation F_i, along with a special clip indicator $\hat{F}_i$. These clip features are used in the subsequent decision-making process to determine if enough information has been gathered to respond.

Real-time Response Decision Module. Based on the video content observed so far and the historical context, the real-time response decision module evaluates whether the model should generate a response or continue waiting for more video content. We illustrate this whole process in Figure 2. To effectively combine these multi-modal inputs, we use an interleaved sequence format, which integrates video features, question information, and decision tokens.

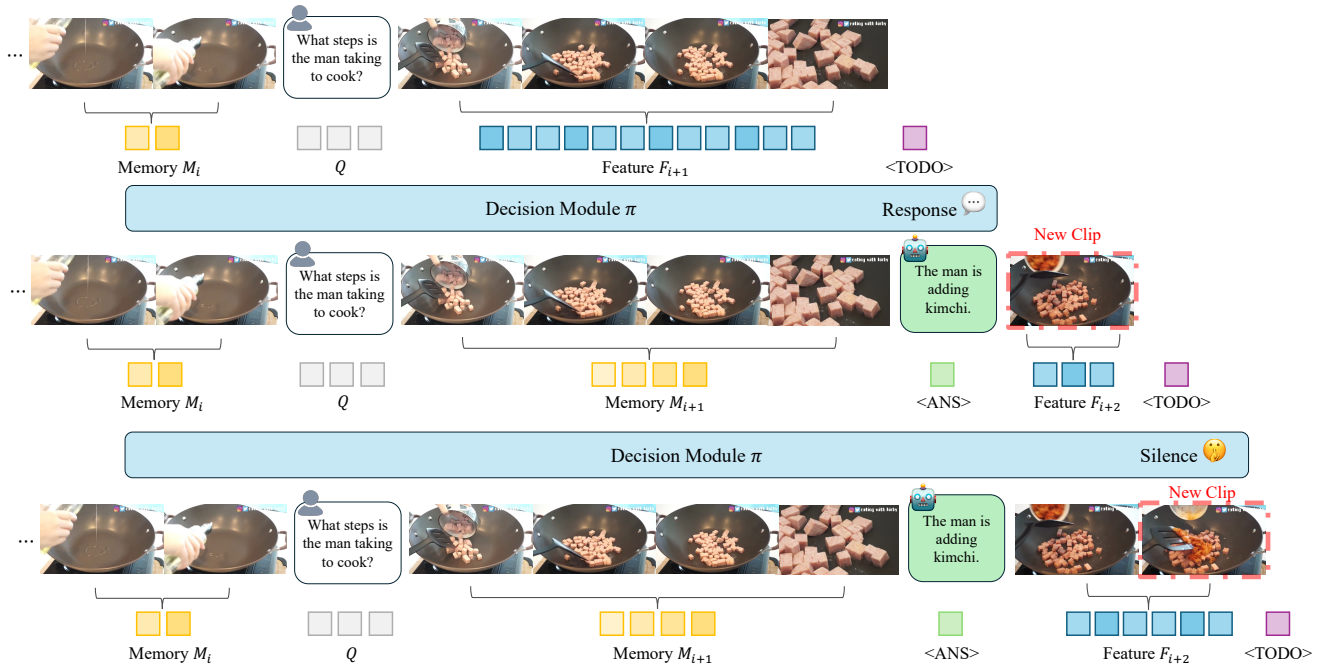

Figure 2. Illustration of the Response Decision Module in a proactive streaming video processing system. The module dynamically determines when to respond during video streaming by segmenting the video into non-uniform clips and utilizing historical memory to capture context. The module then combines memory features, clip features, question text, and special tokens, ⟨TODO⟩ and ⟨ANS⟩, to decide on response timing.

The process begins with the sequence of video clip representations up to the point of the user's question, followed by the question Q and a special token, ⟨TODO⟩, which indicates that a response decision is yet to be made. Formally, the input sequence at this stage is:

$$F[0 : q_{\text{pos}}] \oplus Q \oplus \langle\text{TODO}\rangle,$$

where q_{pos} is the timestamp of the question, and $\oplus$ denotes concatenation.

After this initial stage, the model aggregates the features up to the question timestamp into a global memory M through pooling. Then, we extend the sequence to include this global memory that captures historical context, the question texts, and the clip features from the question timestamp to the current point, ending with ⟨TODO⟩:

$$M[0 : q_{\text{pos}}] \oplus Q \oplus F[q_{\text{pos}} :] \oplus \langle\text{TODO}\rangle.$$

When the model decides to respond, the ⟨ANS⟩ token is used to mark the response action. Specifically, with the video continuing to stream, we further integrate global memory from the question timestamp to the position of the first answer, insert the special answer token ⟨ANS⟩, and include later clips for potential subsequent responses. This sequence concludes with the current clip memory segment and a ⟨TODO⟩ token. Thus, the ultimate input format with multiple answers can be written as:

$$\begin{aligned} &M[0 : q_{\text{pos}}] \oplus Q \oplus \\ &M[q_{\text{pos}} : a^1_{\text{pos}}] \oplus \langle\text{ANS}\rangle \oplus \\ &M[a^1_{\text{pos}} : a^2_{\text{pos}}] \oplus \langle\text{ANS}\rangle \oplus \\ &\cdots \\ &F[a^k_{\text{pos}} :] \oplus \langle\text{TODO}\rangle, \end{aligned}$$

where a^k_{pos} stands for the timestamp for the k-th answer.

Importantly, none of the tokens we utilize originate from the responses generated by the Reaction module. This design ensures that the Decision module remains unblocked by the response generation process, allowing it to continuously monitor the video stream.

We feed the interleaved sequence into a compact LLM and adopt a binary classification head on top of the final-layer embedding of the ⟨TODO⟩ token. This head is trained to predict whether the model should respond at each timestamp. We use standard binary cross-entropy loss to supervise the model's decision-making process.

3.3. Asynchronous Interaction

Once the interaction is triggered, a dedicated asynchronous module is employed to generate contextually fine-grained responses tailored to the state of the video stream. Specifically, for the k-th answer output, the interaction module

receives the current query, the $k-1$ previously produced answers, and the grounded clip features extracted at the corresponding timestamp. Relevant historical clips are retrieved by computing their cosine similarity with the embedding of a designated ⟨TODO⟩ token. This procedure supports multi-hop reasoning, where relevant evidence may be distributed across multiple temporal segments.

To train the model's temporal multi-hop retrieval capabilities, we compute the cosine similarity between the ⟨TODO⟩ token embedding and each historical clip indicator $\hat{F}_i$. Applying the softmax function to these similarity scores yields a predicted relevance distribution $\hat{P}_{\text{pred}}(i)$, representing the relevance of i-th historical clip to the current response.

For supervision, the ground-truth temporal relevance is represented by a binary mask over clips known to be relevant to the current question Q. Defining R_Q as the set of ground-truth relevant clips, we construct the true relevance distribution as:

$$\hat{P}_{\text{true}}(i) = \frac{1}{|R_Q|} \quad \text{if} \quad i \in R_Q \quad \text{else} \quad 0,$$

where R_Q is the ground-truth set of relevant clips for particular question Q.

We then calculate the KL divergence loss between the true distribution, $\hat{P}_{\text{true}}$, and the predicted distribution, $\hat{P}_{\text{pred}}$, as:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{|R_Q|} \sum_{i=1}^{|R_Q|} \hat{P}_{\text{true}}(i) \log \hat{P}_{\text{pred}}(i).$$

This KL divergence loss encourages the model to align its predictions with the true temporal relevance of the clips, thereby improving its ability to correctly retrieve and ground the relevant video segments for generating accurate, timely responses.

In addition, the response decision module may occasionally trigger the interaction actually when no response is required. To accommodate this scenario, we introduce both positive (response-required) and negative (no-response-required) samples when training the interaction module, enabling the model to simulate real-time interactive conditions. The model thus learns to either generate incremental, contextually enriched responses based on newly emerging cues in the video stream or produce a special ⟨SILENT⟩ token to indicate silence when appropriate.

This adaptive reasoning approach ensures that the model remains responsive to new information. By focusing on unanswered content and utilizing past interactions, the model delivers timely and relevant responses, enhancing the overall efficiency and user experience in streaming dialogue generation.

In this way, we decouple the response generation from the video stream processing. This separation allows the streaming video to continue being processed in parallel, without waiting for response generation to complete. As a result, the system remains highly efficient and responsive, ensuring that the video content is continuously monitored and processed while still generating contextually accurate responses at the appropriate moments.

4. Experiments

4.1. Implementation Details

Dispider utilizes a compact LLM as the proactive streaming video processor for response decisions, and a larger LLM for the precision interaction module. In practice, the input video frames are resized to 224×224, and a CLIP-L/14 [51] is employed to extract frame-wise features. Following the token compression techniques in VideoStream [48], we concatenate adjacent tokens and use the compact LLM, instantiated as Qwen2-1.5B [49], to produce time-aware compressed clip-wise features along with clip indicators. Next, we reuse this compact LLM to process the sequence consisting of global memory, question texts, and clip features for response decisions. The final LLM, instantiated as Qwen2-7B [49], receives the grounded clips and global memory to generate responses at the necessary timestamps.

We adopt a two-stage training process. In the first stage, we train the streaming video processor and response decision module using a combination of GroundVQA [14] and ET-Instruct [42] with enriched temporal annotations for supervising streaming responses and providing temporal grounding labels. To further enhance basic reasoning capabilities, we add 50K curated implicit QA pairs with time labels. Next, we construct a dataset of 122K streaming video QA pairs, derived from timestamped QA in ET-Instruct [42] and augmented with data from VideoChatGPT [43] and LLaVA-Next-Video [29], to train the reaction module.

In the second training stage, we freeze the video encoder and the compact LLM, then train only the final interaction module. During training, we insert instructions at various timestamps to improve adaptability. For inference on conventional benchmarks, the question is placed at the end of the video to ensure fair comparisons with prior work, whereas for streaming evaluation, it is posed at the beginning to enable proactive responses.

4.2. Benchmarks

For evaluation, we utilize a range of benchmarks tailored to different aspects of long-video QA and streaming video understanding.

StreamingBench. StreamingBench [36] serves as the latest comprehensive benchmark for evaluating streaming video understanding in multimodal large language models (MLLMs). It emphasizes three critical evaluation aspects: Real-time Visual Understanding, Omni-source Understand-

ing, and Contextual Understanding. The benchmark includes a diverse dataset of 900 videos paired with 4,500 human-annotated QA pairs, with five questions per video asked at varying timestamps.

ETBench Subset. In addition to StreamingBench, we construct a streaming video QA benchmark using a subset of ETBench to measure the model's proactive response capability in real-time video interactions. Specifically, we select six subtasks from ETBench suitable that require the model to predict explicit event timestamps: step localization (SLC), dense video captioning (DVC), temporal action localization (TAL), temporal video grounding (TVG), episodic memory (EPM), and video highlight detection (VHD), encompassing a total of 4,460 videos. For this benchmark, we test our model using two versions. In the conventional setting, we pose the instruction at the end of the question. In the streaming setting, the instruction is provided at the beginning of each video, and the model is required to produce the correct responses at appropriate timestamps as the video plays. We report the F1 score for temporal grounding evaluation and the sentence similarity score for caption evaluation.

Long-Video QA Benchmarks. Finally, we adopt several long-video QA benchmarks, including EgoSchema [44], VideoMME [20], and MLVU [75]. EgoSchema consists of over 5K videos, each approximately 3 minutes long, while VideoMME and MLVU include videos of varying lengths, from a few minutes to several hours. We report accuracy on multiple-choice questions for comparison across these benchmarks.

4.3. Streaming Video Understanding

We evaluate Dispider's performance in streaming video interactions, emphasizing its ability to process real-time input and respond dynamically. Questions are posed at the start of the video, and the model generates responses only when relevant clues are detected, remaining silent otherwise for meaningful, context-aware interactions.

As shown in Table 1, Dispider significantly outperforms Flash-VStream [69] and VideoLLM-online [5], particularly in the Proactive Output (PO) task. This task requires the model to determine the precise timing of its response, such as outputting "GOAL" when a goal is scored. Unlike standard input-output tasks, it demands proactive maintaining an internal state to track relevant video frames.

While other streaming models fail in this task, Dispider achieves a competitive score of 25.3. Even compared to offline Video LLMs, where the question is provided after the video has played, Dispider demonstrates superior proactive decision-making capabilities by handling questions posed at the start of the video.

For the ETBench subset in the streaming setting, as shown in the bottom rows of Table 3, our model consistently outperforms VideoLLM-Online [5] across all metrics, with particularly notable improvements in temporal grounding. This demonstrates that our disentangled design equips the model with much stronger temporal awareness, enabling more proactive responses aligned with specific instructions. Notably, on dense video captioning and step localization tasks, our model achieves both more precise temporal grounding and more comprehensive descriptions in streaming mode than the conventional setting. This indicates that our disentangled architecture can effectively monitor the video stream in real-time and proactively generate informative responses according to the instructions. And the streaming interaction has the potential to achieve more powerful video understanding.

Additionally, we present a quantitative comparison between our model and VideoLLM-online in Fig. 3. From this comparison, when faced with questions requiring multi-step reasoning, our model can gradually identify the necessary clues from the video stream and generate informative answers step by step, while VideoLLM-online [5] fails to do so. For example, from the 'thirsty' in the question, our model can associate it with the drinks appearing in the video stream, and infer what actions are needed based on the context. In contrast, VideoLLM-online only gives simple descriptions of the scene or ongoing actions.

A key advantage of our method is the disentangled perception, decision, and reaction architecture, which enables the model to simultaneously process the streaming video input and generate responses in a non-blocking manner. However, VideoLLM-online has to experience interruptions in the video stream during answer generation as shown in Figure 2.

4.4. Conventional Video Understanding

In this section, we compare our model with existing video LLMs on conventional video QA benchmarks, where the model is required to provide one answer after watching the complete video.

We first present a comparative analysis across three challenging long-video benchmarks. As shown in Table 2, we report the accuracy on the EgoSchema full set, MLVU multiple-choice questions, and the VideoMME overall set without subtitles. Generally, our disentangled architecture handles this conventional scenario well and achieves competitive performance. Notably, on EgoSchema, which requires long temporal reasoning, our method achieves a leading performance of 55.6, demonstrating strong temporal perception.

In terms of MLVU and VideoMME, which consist of videos ranging from minutes to hours long, our model's promising results highlight its ability to efficiently process contextual information across varying temporal lengths. This capability is crucial for streaming video interactions,

Model	Params	Frames	Real-Time Visual Understanding											Omni-Source Understanding					Contextual Understanding					Overall
			OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	**All**	ER	SCU	SD	MA	**All**	ACU	MCU	SQA	PO	**All**	
Human																								
Human‡	-	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46	88.00	88.24	93.60	90.27	90.26	88.80	90.40	95.00	100	93.55	91.66
Proprietary MLLMs																								
Gemini 1.5 pro	-	1 fps	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69	46.80	39.60	74.90	80.00	60.22	51.41	40.73	54.80	45.10	48.73	67.07
GPT-4o	-	64	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28	41.20	37.20	43.60	56.00	44.50	41.20	38.40	32.80	56.86	38.70	60.15
Claude 3.5 Sonnet	-	20	80.49	77.34	82.02	81.73	72.33	75.39	61.11	61.79	69.32	43.09	72.44	31.60	34.00	32.80	48.80	36.80	38.40	34.80	34.40	64.71	37.70	57.68
Open-Source Video MLLMs																								
LLaVA-OneVision	7B	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12	40.80	37.20	33.60	44.80	38.40	35.60	36.00	27.27	29.55	32.74	56.36
Qwen2-VL	7B	0.2-1 fps	75.20	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04	41.20	22.00	32.80	43.60	34.90	31.20	26.00	39.60	22.73	31.66	54.14
MiniCPM-V 2.6	8B	32	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44	40.80	24.00	34.00	41.20	35.00	34.00	31.60	41.92	22.22	34.97	53.85
LLaVA-NeXT-Video	32B	64	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96	37.69	24.80	34.40	42.80	34.90	29.20	30.40	35.35	18.18	30.79	52.77
InternVL-V2	8B	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72	37.60	26.40	37.20	42.00	35.80	32.00	31.20	32.32	40.91	32.42	51.40
Kangaroo	7B	64	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60	37.60	31.20	28.80	39.20	34.20	32.80	26.40	33.84	16.00	30.06	51.10
LongVA	7B	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96	39.60	32.40	28.00	41.60	35.40	32.80	29.60	30.30	15.91	29.95	48.66
VILA-1.5	8B	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32	41.60	26.40	28.40	36.00	33.10	26.80	34.00	23.23	17.65	27.35	43.20
Video-CCAM	14B	96	56.40	57.81	65.30	62.75	64.60	51.40	42.59	47.97	49.58	31.61	53.96	33.60	22.00	28.40	34.80	29.70	27.60	24.40	16.67	22.73	22.88	42.53
Video-LLaMA2	7B	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52	30.40	32.40	30.40	36.00	32.40	24.80	26.80	18.67	0.00	21.93	40.40
Streaming MLLMs																								
Flash-VStream	7B	-	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23	25.91	24.90	25.60	28.40	26.00	24.80	25.20	26.80	1.96	24.12	24.04
VideoLLM-online	8B	2 fps	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99	31.20	26.51	24.10	32.00	28.45	24.19	29.20	30.80	3.92	26.55	32.48
Dispider (ours)	7B	1 fps	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63	35.46	25.26	38.57	43.34	35.66	39.62	27.65	34.80	25.34	33.61	53.12

Table 1. Performance comparison on StreamingBench on Omni-source Understanding, Contextual Understanding, and Real-Time Visual Understanding. Omni-source Understanding includes Emotion Recognition (ER), Scene Understanding (SCU), Source Discrimination (SD), and Multimodal Alignment (MA). Contextual Understanding includes Misleading Context Understanding (MCU), Anomaly Context Understanding (ACU), Sequential Question Answering (SQA) and Proactive Output (PO). Real-Time Visual Understanding includes Object Perception (OP), Causal Reasoning (CR), Clips Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT). Results are categorized into Human, Proprietary MLLMs, and Open-Source MLLMs for a comprehensive evaluation.

as it allows the model to accurately summarize context when questions are inserted at arbitrary temporal positions.

Additionally, we adopt a recent time-aware benchmark, ETBench [42], to evaluate temporal awareness. In the conventional setting in Table 3, our method is able to capture the timestamps corresponding to specific questions. Without a specialized design for time tokens, our model achieves the highest F1 score in temporal video grounding and episodic memory subtasks, even surpassing models with specialized temporal tokens. The promising performance on dense video captioning and step localization further demonstrates the model's ability in both accurate temporal grounding and precise visual perception.

4.5. Ablation Study

Clip Segmentation Strategy. We first present an ablation study on clip segmentation. We compare the standard uniform clip segmentation (16 frames per clip sampled at 1 FPS) with our scene-based non-uniform segmentation. We report conventional QA accuracy on MLVU and VideoMME, as well as streaming metrics on temporal video grounding and dense video captioning on ETBench in Table 4. The scene-based segmentation yields superior results in both conventional and streaming settings. This aligns with our motivation that scene-based segmentation preserves more structural video information, which facilitates better model learning and more timely responses.

Special Token Design. We also ablate three special tokens in our architecture, i.e., ⟨ANS⟩ and ⟨TODO⟩ in the streaming video processor, ⟨SILENT⟩ token in the final LLM. These three special tokens mainly impact the streaming video interactions, so we report the streaming evaluation metrics on temporal video grounding and dense video captioning in Table 5. There are three observations. *First*, the absence of ⟨ANS⟩ token can lead the model to be unaware of the timestamps at which a response has been given. As a result, if relevant clues have appeared in the video contexts, the model tends to produce a response, resulting in a high recall but a low accuracy. *Second*, the ⟨TODO⟩ serves as an indicator that reminds the streaming processor to decide whether to respond. The performance slightly degrades without this special token. *Third*, the ⟨SILENT⟩ token in the final LLM serves as a secondary filter for response decision. If the preceding streaming processor incorrectly identifies a timestamp as requiring a response, the ⟨SILENT⟩ token enables the LLM to rethink whether an answer is needed

Method	LLM Size	Frames	EgoSchema	MLVU	VideoMME
Video-LLaVA [35]	7B	8	38.4	47.3	39.9
Chat-UniVi [28]	7B	64	-	-	40.6
LLaMA-VID [34]	7B	1 FPS	38.5	33.2	-
TimeChat [53]	7B	96	33.0	30.9	30.2
MovieChat [55]	7B	2048	53.5	25.8	38.2
Video-LLaMA2 [9]	7B	16	51.7	48.5	47.9
LLaVA-Next-Video [74]	7B	32	43.9	-	46.6
ShareGPT4Video [6]	8B	16	-	46.4	39.9
VideoChat2 [32]	7B	16	54.4	47.9	39.5
LongVA [73]	7B	128	-	56.3	52.6
Kangaroo [41]	8B	64	-	61.0	56.0
Video-CCAM [18]	14B	96	-	63.1	53.2
VideoXL [54]	7B	128	-	64.9	55.5
Dispider (ours)	7B	1 FPS	55.6	61.7	57.2

Table 2. Comparison on long video benchmarks. We report the accuracy on the EgoSchema full set, MLVU multiple-choice questions, and the VideoMME overall set without subtitles. For a fair comparison, we also present the model size of the LLM and the number of sampled frames.

Method	Frames	TVG_{F1}	EPM_{F1}	TAL_{F1}	VHD_{F1}	DVC_{F1}	DVC_{Sim}	SLC_{F1}	SLC_{Sim}
Conventional video QA inference.									
w/ specialized time tokens									
VTG-LLM [24]	96	15.9	3.7	14.4	48.2	40.2	18.6	20.8	14.4
LITA [26]	100	22.2	4.6	18.0	23.9	39.7	17.2	21.0	12.2
ETChat [42]	1 FPS	38.6	10.2	30.8	62.5	38.4	19.7	24.4	14.6
w/o specialized time tokens									
VideoChatGPT [43]	100	7.0	1.3	15.1	28.8	8.8	11.3	5.7	10.2
Video-LLaVA [35]	8	7.0	1.9	15.0	28.9	28.0	15.0	0.9	8.3
LLaMA-VID [34]	1 FPS	5.5	1.2	8.0	30.0	27.1	12.6	5.2	11.1
Video-LLaMA2 [9]	8	0.1	0.0	0.0	1,5	0.6	14.5	0.0	15.2
PLLaVA [63]	16	6.9	1.1	5.7	28.9	13.3	10.6	9.7	11.8
VTimeLLM [25]	100	7.6	1.9	18.2	28.9	12.4	13.1	8.7	6.4
TimeChat [53]	96	26.2	3.9	10.1	40.5	16.6	12.5	5.6	9.2
Dispider (ours)	1 FPS	43.6	17.2	29.9	51.5	31.6	17.8	14.1	11.7
Streaming video QA inference.									
VideoLLM-Online [5]	2 FPS	13.2	3.8	9.1	22.4	24.0	13.4	9.9	10.1
Dispider (ours)	1 FPS	36.1	15.5	27.3	54.2	33.8	18.9	18.8	12.4

Table 3. Comparison on ETBench. We present the results for two different settings. In the conventional video QA setting, the model is required to answer the question after watching the entire video. In the streaming setting, the question is placed at the beginning of the video, and the model is expected to provide real-time responses. We report performance on six subtasks that are suitable for both evaluation settings.

Clip	MLVU	V-MME	TVG_{F1}	DVC_{F1}	DVC_{Sim}
Uniform	59.8	55.4	34.5	33.1	18.1
Scene-based	61.7	57.2	36.1	33.8	18.9

Table 4. Ablation study on the clip segmentation. We compare uniform 16-frame clip segmentation and our scene-based segmentation with SigLip.

referring to the historical QA interaction contexts.

5. Conclusion

In this work, we introduced Dispider, a novel framework designed to enable active real-time interaction with video LLMs. By disentangling the key capabilities of perception, decision, and reaction, and adopting an asynchronous processing approach, Dispider overcomes the inherent conflicts that hinder real-time interaction in traditional video

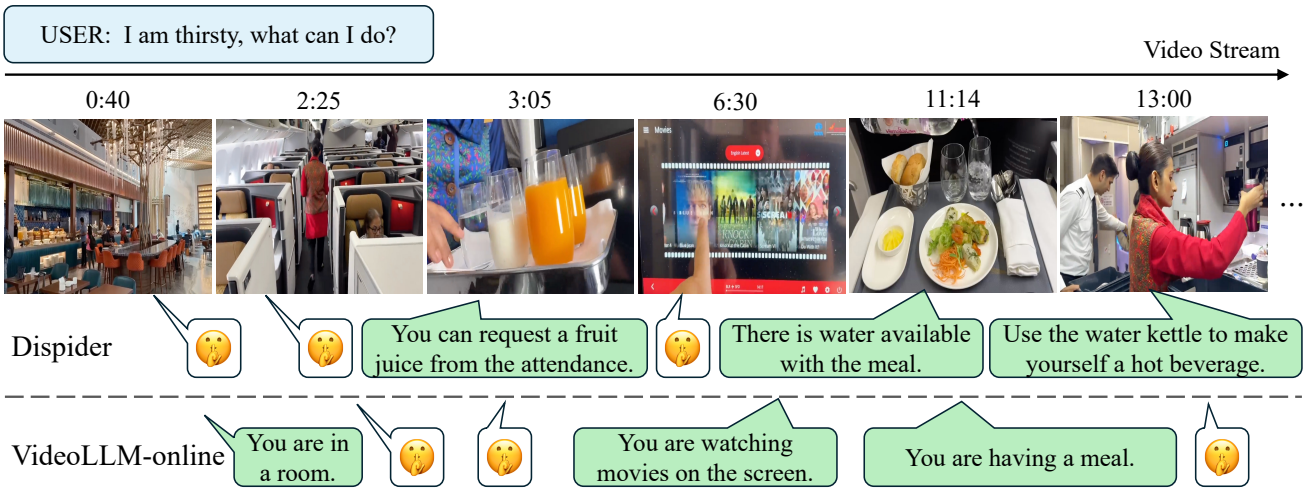

Figure 3. Quantitative comparison between Dispider and VideoLLM-online in streaming video understanding. The question is inserted at the beginning, and we show the model's response in either answer texts or the state of keeping silent.

⟨ANS⟩	⟨TODO⟩	⟨SILENT⟩	TVG_{F1}	DVC_{F1}	DVC_{Sim}
✗	✗	✗	20.1	19.7	12.3
✗	✗	✓	26.3	24.9	13.1
✓	✗	✓	35.2	31.0	17.2
✗	✓	✓	28.7	25.6	14.5
✓	✓	✗	35.5	30.2	16.8
✓	✓	✓	36.1	33.8	18.9

Table 5. Ablation study on the special token designs. We respectively explore the effectiveness of ⟨ANS⟩ and ⟨TODO⟩ in streaming video processor as well as ⟨SILENT⟩ in the final LLM.

LLMs. Our approach ensures that the model can continuously process the video stream while providing timely, contextually accurate, and precise responses to user interactions. The Proactive Streaming Video Processing module optimizes the video analysis by focusing on relevant content, while the Precise Interaction module generates detailed responses asynchronously, allowing for uninterrupted video processing. We demonstrated the effectiveness of Dispider through extensive experiments on both conventional and streaming video QA benchmarks, where it outperformed existing methods in proactive response capabilities, temporal awareness, and computational efficiency. Dispider's disentangled architecture and ability to handle long-duration video streams make it an ideal solution for real-time interactive applications.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 2

[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[4] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 2

[5] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *CVPR*, 2024. 1, 2, 3, 6, 8

[6] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024. 2, 8

[7] Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. Motionllm: Understanding human behaviors from human motions and videos. *arXiv preprint arXiv:2405.20340*, 2024.

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[9] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2, 8

[10] 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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[11] Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2303.12345*, 2023. 2

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[14] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12934–12943, 2024. 5

[15] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 2

[16] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024. 2

[17] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. *arXiv preprint arXiv:2403.11481*, 2024. 2

[18] Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024. 8

[19] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 2

[20] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 2, 6

[21] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Xiong Wang, Di Yin, Long Ma, Xiawu Zheng, Ran He, Rongrong Ji, Yunsheng Wu, Caifeng Shan, and Xing Sun. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*, 2024. 2

[22] Gemini Team Google. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2

[23] Gemini Team Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 2

[24] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv preprint arXiv:2405.13382*, 2024. 8

[25] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024. 8

[26] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2025. 8

[27] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 2

[28] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023. 8

[29] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 5

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[31] 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 arXiv:2305.06355*, 2023. 2

[32] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 8

[33] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 2

[34] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024. 2, 8

[35] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual represen-

tation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 8

[36] Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024. 2, 5

[37] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, et al. Mm-vid: Advancing video understanding with gpt-4v (ision). *arXiv preprint arXiv:2310.19773*, 2023. 2

[38] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 2

[39] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2

[40] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2

[41] Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024. 2, 8

[42] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Chang Wen Chen, and Ying Shan. E.t. bench: Towards open-ended event-level video-language understanding. In *Neural Information Processing Systems (NeurIPS)*, 2024. 2, 5, 7, 8

[43] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 2, 5, 8

[44] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6

[45] OpenAI. Gpt-4 technical report. 2023. 2

[46] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2

[47] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. Gpt4point: A unified framework for point-language understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26417–26427, 2024. 2

[48] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3, 5

[49] Team Qwen. Qwen2 technical report, 2024. 5

[50] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 2

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763, 2021. 5

[52] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 2

[53] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024. 2, 8

[54] Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024. 8

[55] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 2, 8

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[57] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2

[58] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024. 2

[59] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer, 2025.

[60] Qi Wu, Yubo Zhao, Yifan Wang, Yu-Wing Tai, and Chi-Keung Tang. Motionllm: Multimodal motion-language learning with large language models. *arXiv preprint arXiv:2405.17013*, 2024. 2

[61] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024. 2

[62] Haiyang Xu, Qinghao Ye, Mingshi Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qiuchen Qian,

Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Feiran Huang, and Jingren Zhou. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv preprint arXiv:2302.00402*, 2023. 2

[63] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024. 8

[64] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13040–13051, 2024. 2

[65] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024. 2

[66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 3

[67] 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 arXiv:2312.17235*, 2023. 2

[68] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2

[69] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. 2024. 3, 6

[70] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. 2

[71] Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, et al. Internlm-xcomposer2. 5-omnilive: A comprehensive multimodal system for long-term streaming video and audio interactions. *arXiv preprint arXiv:2412.09596*, 2024. 2

[72] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024. 2

[73] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 8

[74] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 8

[75] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 2, 6