VideoLLM Knows When to Speak: Enhancing Time-Sensitive Video Comprehension with Video-Text Duet Interaction Format

Yueqian Wang[1], Xiaojun Meng[2], Yuxuan Wang[3], Jianxin Liang[1], Jiansheng Wei[2], Huishuai Zhang[1,4], Dongyan Zhao[1,4],
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Huawei Noah's Ark Lab [3]Beijing Institute for General Artificial Intelligence
[4]State Key Laboratory of General Artificial Intelligence

Correspondence: zhanghuishuai@pku.edu.cn, zhaodongyan@pku.edu.cn

Abstract

Recent researches on video large language models (VideoLLM) predominantly focus on model architectures and training datasets, leaving the interaction format between the user and the model under-explored. In existing works, users often interact with VideoLLMs by using the entire video and a query as input, after which the model generates a response. This interaction format constrains the application of VideoLLMs in scenarios such as live-streaming comprehension where videos do not end and responses are required in a real-time manner, and also results in unsatisfactory performance on time-sensitive tasks that requires localizing video segments. In this paper, we focus on a video-text duet interaction format. This interaction format is characterized by the continuous playback of the video, and both the user and the model can insert their text messages at any position during the video playback. When a text message ends, the video continues to play, akin to the alternative of two performers in a duet. We construct MMDuetIT, a video-text training dataset designed to adapt VideoLLMs to video-text duet interaction format. We also introduce the Multi-Answer Grounded Video Question Answering (MAGQA) task to benchmark the real-time response ability of VideoLLMs. Trained on MMDuetIT, MMDuet demonstrates that adopting the video-text duet interaction format enables the model to achieve significant improvements in various time-sensitive tasks (76% CIDEr on YouCook2 dense video captioning, 90% mAP on QVHighlights highlight detection and 25% R@0.5 on Charades-STA temporal video grounding) with minimal training efforts, and also enable VideoLLMs to reply in a real-time manner as the video plays.

1 Introduction

Videos are becoming an increasingly important medium to acquire information on a daily basis. Powered by recent advancements in large language models (LLMs) (Touvron et al., 2023; Jiang et al., 2023; Shao et al., 2024; Dubey et al., 2024; Yang et al., 2024) and vision encoders (Radford et al., 2021; Zhai et al., 2023; Sun et al., 2023; Oquab et al., 2023; Wang et al., 2024b), several video large language models (VideoLLM) (Li et al., 2023; Liu et al., 2024; Li et al., 2024b,a; Zhang et al., 2024b; Wang et al., 2024d) have already demonstrated strong abilities for holding conversations and answering questions about videos. A common feature of these models is using visual encoders to encode all frames sampled from the entire video at first, and integrate them into text input by concatenating them to input embeddings or using cross attention.

Recent research on VideoLLMs has primarily concentrated on model architectures and training datasets, with limited exploration of the interaction format between the user and the model. In this paper, the "interaction format" of VideoLLMs comprises the following two aspects: (1) a **chat template** used to convert input sources, e.g., video, user text query, and model response, into a sequence of tokens; (2) a **turn-taking rule** organizing inputs of different sources to finalize an interaction format. For example, for most existing VideoLLMs, the interaction format is: (1) for the chat template, the model uses (frames sampled from) the full video and a text query as input, and then outputs a response; (2) for the turn-taking rule, usually the model is permitted to take its turn to generate a response when both the whole video content and user query have ended, e.g., when an <eos> token is explicitly provided. We refer to this traditional interaction method as "*whole video*" in the rest of this paper.

However, this all-along used *whole video* interaction has the following two defects, which hinder the performance and real-world usage scenarios of VideoLLMs: Firstly, it does not admit timely interactions. As the video is often input as a whole, this

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 6338–6359
November 4-9, 2025

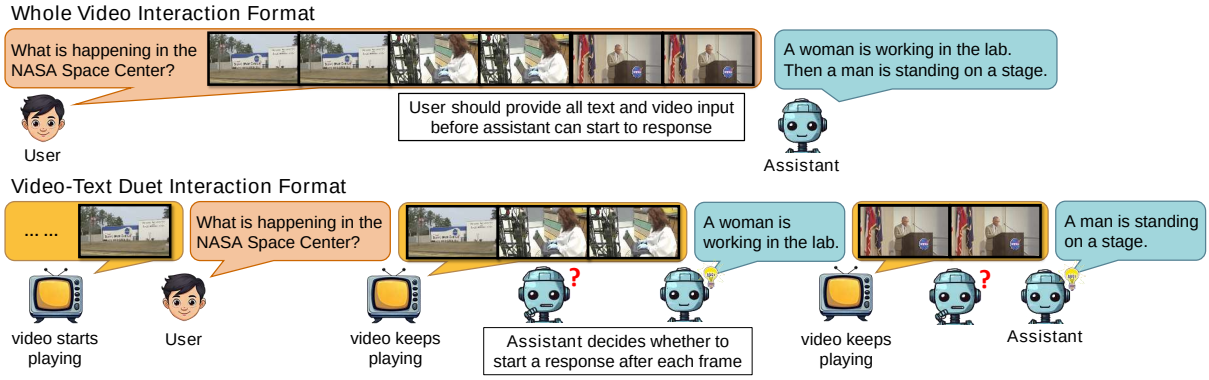


Figure 1: An example of the common *Whole Video* Interaction Format and our Video-Text Duet Interaction Format.

limits its usage in more scenarios like live broadcasts or surveillance videos, in which the video does not end at a specific time. Even if we can segment the video into multiple fixed-length clips for input, the model still cannot generate responses in a real-time manner when necessary, as it does not know whether it is feasible and appropriate to reply at the end of this clip. Secondly, it performs unfavorably on time-sensitive video comprehension tasks. In this paper we use **"time-sensitive tasks"** to refer to tasks in which the model is required to provide responses that include specific times in the video, such as temporal video grounding (Krishna et al., 2017; Gao et al., 2017; Hendricks et al., 2017), video highlight detection (Lei et al., 2021), dense video captioning (Zhou et al., 2017; Krishna et al., 2017), grounded video question answering (Xiao et al., 2023), etc.

In this work, we formalize the Video-Text Duet Interaction Format, an interaction method that aims to enhance VideoLLMs by addressing the aforementioned issues. An illustration of the *whole video* interaction format and the video-text duet interaction format is shown in Fig. 1. With our video-text duet interaction format, the video is continuously played and input to the model frame-by-frame. Both the user and model can insert their text messages right after any frame during the video play. When a dialogue turn from either the user or the model ends, the video stream can have the floor and input video frames to the model until another turn is started by either the user or the model, akin to the show of two performers in a duet. This improves the timeliness of interaction and better suits real-world applications such as live-streaming or surveillance video comprehension. Moreover, by inserting responses to the video where is most relevant, the model can learn to generate responses by referencing a smaller but fine-grained fraction of the video before this position. In this manner, it facilitates information retrieval to describe lengthy videos, as well as enables a response to be "grounded" at the targeted position of the video. We believe this design contributes to addressing the above discussed issues of existing *whole video* VideoLLMs.

To prove the effectiveness of the video-text duet interaction format, we construct MMDuetIT, a dataset to facilitate the training of a versatile VideoLLM following the video-text duet interaction format. We propose Multi-Answer Video Grounded QA (MAGQA), a novel task that requires the model to generate answers at appropriate timespans in a real-time manner to align with potential applications of live-streaming video comprehension. We also train MMDuet, a VideoLLM that implements our proposed video-text duet interaction format. Initialized with LLaVA-OneVision (Li et al., 2024a) and trained with MMDuetIT at a low cost, MMDuet achieves significant performance improvement in various time-sensitive tasks, and is able to generate responses in real-time as the video plays.

2 Related Works

The advancement of large language models (LLMs) and visual encoders has led to numerous efforts on their integration, aiming to utilize the powerful understanding and generation abilities of existing LLMs for video-related tasks (Li et al., 2023; Liu et al., 2024; Li et al., 2024b,a; Wang et al., 2024d; Xu et al., 2023). These models exhibit a decent ability of video understanding such as captioning or summarizing (Xu et al., 2023). However, their performance on time-sensitive tasks is still unsatisfactory.

Recent works attempt to empower VideoLLMs with the ability to localize and represent segments in videos, and thus achieve better performance on tasks like temporal video grounding or dense video captioning. These works explore new ways on how to easily represent video clips with texts, such as second numbers of timestamp (TimeChat (Ren et al., 2023)), timeline percentage (VTimeLLM (Huang et al., 2023)) or using special textual tokens (VTG-LLM (Guo et al., 2024), Grounded-VideoLLM (Wang et al., 2024a)). However, their performance has not been satisfactory yet, possibly due to LLMs' limited ability to accurately count and generate numbers (Schwartz et al., 2024) to localize each video frame. To alleviate this issue, HawkEye (Wang et al., 2024c) uses a coarse-grained method by referring to a larger fraction of the video, but it requires multiple rounds of recursive grounding to precisely locate a segment and may not express multiple segments at a time.

The work most similar to our motivation is VideoLLM-Online (Chen et al., 2024), which proposes a framework named LIVE for training VideoLLMs to interrupt video streams and insert responses. However, they only finetune a model on Ego4D (Grauman et al., 2021) and COIN (Tang et al., 2019) to demonstrate the LIVE training and inference, and do not explore on how the model capabilities vary with this new type of interaction, especially the zero-shot performance on time-sensitive tasks.

Our work differs from VideoLLM-Online at: Firstly, providing a more general description of the video-text dual interaction format, including a wider variety of criteria for determining whether a response should be generated, and its application on new tasks such as temporal video grounding and grounded question answering; Secondly, introducing a new dataset MMDuetIT and the method on building such datasets; Thirdly, proposing a new task MAGQA; Lastly, proposing a more powerful model MMDuet that has state-of-the-art performance on various time-sensitive tasks and zero-shot generalization ability.

3 The Video-Text Duet Interaction Format

In Section 1, we have defined the concept of "interaction format" with two aspects (*i.e.,* chat template & turn-taking rule), as well as the drawbacks of the commonly-used *whole video* interaction format. Now we re-emphasize and formalize our video-text duet interaction format, which is completely different from previous to implement VideoLLMs.

(1) For the chat template, inspired by but different from the LIVE framework which is used to implement VideoLLM-Online (Chen et al., 2024), we consider the video stream as a conversation participant just like the role of user/assistant, and the input sequence consists of alternating turns among these three roles. (2) For the turn-taking rule, when the turn of the user or assistant ends, the video stream can take the floor and start its turn to input video frames. When each single frame is consumed, both the user and the assistant role can interrupt the video stream at any time, and start its own turn to query or generate a response, as totally decided by the user or the assistant, respectively.

4 MMDuet: Our Proposed VideoLLM

4.1 Model Structure

We propose MMDuet, a model trained following the video-text duet interaction format, which can thus autonomously decide at what position in the video to generate what response. Like almost all existing VideoLLMs, MMDuet consists of three components: 1) a visual encoder that encodes sampled frames from the video to visual feature, 2) a linear projector that transforms the encoded visual feature to a list of visual tokens that is aligned into the LLM textual embedding space, and 3) a transformer-decoder-based LLM that takes both textual and visual tokens as input and uses its language modeling head to predict the next token.

The only difference in model structure between our MMDuet and existing VideoLLMs is that we add two more heads in addition to the language modeling head (LM Head) of the LLM, namely the **informative head** and the **relevance head**, for determining whether to start a response after each frame. Each head is a linear layer and has a weight with shape $h \times 2$, where h is the hidden size of the used LLM. Each head takes the final layer hidden state of the last visual token of each frame as input, and performs a binary classification. To be specific, 1) the informative head is designed to predict how much new information is acquired upon viewing the current frame. If the model can obtain a "significant amount" of new information upon viewing a new frame (which we will further discuss in Section 5.1), it should classify this frame as *TRUE* category; otherwise, it should classify it as *FALSE*. 2)

The relevance head is designed to predict whether the current frame is related to the user query. Similarly, *TRUE* category means to be related, while *FALSE* means not. We denote the probability of *TRUE* category of informative head and relevance head as **informative score** and **relevance score** for each sampled video frame. These two scores will be used to decide whether the model (*i.e.,* assistant role) should interrupt the video and start its own turn. Compared with VideoLLM-Online (Chen et al., 2024) that makes this decision by predicting one special token using the LM Head, our design has the following merits: (1) The ground truth labels of informative scores and related scores are acquired based on the characteristic of the video itself, rather than on ad-hoc response decisions. Therefore, there are better labels for models to converge during training. (2) By combining two scores we can flexibly set different criteria for response generation, rather than only relying on the logits of one special token; (3) The relevance head can be used to precisely perform temporal video grounding and highlight detection tasks, expanding the application scenarios of MMDuet.

4.2 Inference Procedure

When consuming every single sampled frame of the video, we first check if there is a user query happening at this time. If yes, we first input this user turn to the model. Then the sampled frame is input to the model, after which the informative score and relevance score are calculated. We use a function `need_response` to estimate whether the model should generate an assistant response according to the informative scores and relevance scores for this frame along with previous frames. If yes, the `generate` function of the LLM outputs a response. Different `need_response` functions can be designed depending on the specific task, which is introduced in the experiment section (Section 6). This process can be efficiently implemented by updating the KV Cache each time when a frame or text is input or generated, and a python-style sudo code is provided in Appendix B.3.

5 MMDuetIT: Dataset for Training MMDuet

We build MMDuetIT, a dataset for training the MMDuet model to learn to calculate the informative and relevance scores, and autonomously output replies at any necessary time in the play of the video. MMDuetIT is composed of three different types of tasks that benefit our model training: dense captioning, multi-answer grounded video question answering, and temporal video grounding. An example of the input format for each task is listed in Appendix D.

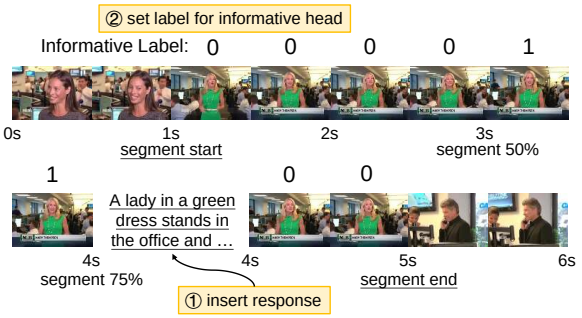


Figure 2: Example of reformatting the annotation of a video segment to video-text duet interaction format in MMDuetIT. Information from the original annotation is emphasized with underlines.

5.1 Dense Captioning

We use Shot2Story (Han et al., 2023), a video-text dataset with segment-level captions, as our dense captioning training data. Specifically, we use the 43k human-annotated subset due to its high-quality and detailed annotations. We preprocess the data to serve our purposes, and an illustration of reformatting the video segment and caption annotations to video-text duet interaction format is in Fig. 2: we randomly sample a position from 50% to 75% time duration for the corresponding video segment, and insert the caption at that position as a model response. We also create labels for the informative head in dense captioning tasks by setting the informative head's label to *TRUE* for frames between 50% of this segment and the insertion point of the response, and set labels to *FALSE* for the other frames. To adapt to long video input, we also select videos with 2 to 4 minutes in length from COIN (Tang et al., 2019) as a dense captioning task to MMDuetIT. The annotations in COIN are reformatted using the same method as Shot2Story. For more details about this data reformat process please refer to Appendix B.1.

5.2 Multi-Answer Grounded Video QA

An important application scenario for the video-text duet interaction format is multi-answer grounded video question-answering (MAGQA). Consider when we are watching a live broadcast of a basketball game and want to track the actions

of a particular player in the game. This exemplifies a MAGQA task: the question is "What does this particular player do in the video?". Each time this player performs an action, the model should respond with a description of this action (*i.e.,* multiple answers) in a real time manner. We believe this newly proposed MAGQA task can be widely used in real-world scenarios when users interact with a live-streaming video.

We construct training data for this task using GPT4o-2024-08-06 (OpenAI, 2024). Given the captions of all segments from the video as input, GPT4o is prompted to generate a question related to one or more captions. For each of the segment captions, if it is related to the question, then GPT4o should also generate an answer that can be inferred from this caption. Otherwise, GPT4o should reply with "`Not Mentioned.`", and this answer is not added to the training data. We use the same insertion method of dense captioning task as described in Section 5.1, to insert the answers into the video stream and construct informative head labels, and the question is inserted at a random place before the first answer. We also use the same insertion method to convert the human-annotated Shot2Story test set and randomly sampled 2000 examples as the test set of our MAGQA benchmark in Section 6.3. Therefore, this dataset contains 36834 examples in the train set and 2000 examples in the test set, and we name it as "Shot2Story-MAGQA-39k".

We have manually checked its data quality, and details of this process are stated in Appendix A.

5.3 Temporal Video Grounding

We also add DiDeMo (Hendricks et al., 2017), HiREST$_{grounding}$ (Zala et al., 2023) and QuerYD (Oncescu et al., 2021), three temporal video grounding tasks in MMDuetIT. Note that these data are used only for training the relevance head, which is designed for performing temporal video grounding tasks and judging the relevance between the question and the video for QA tasks. The query is first added at the beginning of the input sequence. For frames that are annotated as relevant to the query, we set the relevance head's label to *TRUE*; otherwise, we set it to *FALSE*.

5.4 Dataset Statistics

The data distribution of MMDuetIT is shown in Fig. 3. Note that this dataset only contains 109k examples, which is relatively small compared to modern post-training datasets like (Li et al., 2023, 2024a; Wang et al., 2024c). The reason is that due to computational resource constraints, we plan to demonstrate the feasibility of our proposed video-text duet interaction format by fine-tuning a state-of-the-art VideoLLM. We assume that the used backbone model already possesses enough video comprehension capabilities. By using a small dataset, we aim to train this model to efficiently adopt this new interaction with minimum catastrophic forgetting of its existing abilities.

6 Experiments

Implementations MMDuet is initialized with LLaVA-OneVision (Li et al., 2024a). We train the model on MMDuetIT for one epoch. The training takes about one day on a node with 8 Tesla V100 GPUs, and the inference runs on 1 Tesla V100 GPU. More implementation details are listed in Appendix B.2.

Baselines As MMDuet mainly focuses on time-sensitive video tasks, we use the following baselines that are able to represent time spans in videos by different representation formats: TimeChat (Ren et al., 2023), VTimeLLM (7B) (Huang et al., 2023), HawkEye (Wang et al., 2024c), VTG-LLM (Guo et al., 2024), and VideoLLM-Online (Chen et al., 2024). For VideoLLM-Online, we experimented with $\theta \in \{0.5, 0.6, 0.7, 0.8\}$ as suggested in their paper and report the best results (0.8 for both dense video captioning and MAGQA).

Since the initialization of MMDuet is stronger than that of the baselines, for a fair comparison we also conduct a controlled experiment in which the only difference is the interaction format. Specifically, we use the same initialization model (LLaVA-OneVision), training data (MMDuetIT) and training schedule, but reformat the data to the respective interaction formats and video segment representation formats used by TimeChat and VTimeLLM to train two baseline models. We name these models as LLaVA-OV-TC and LLaVA-OV-VT.

6.1 Highlight Detection and Temporal Video Grounding

We use highlight detection and temporal video grounding to evaluate the performance of the relevance head of MMDuet. Baseline models are required to generate a list of float numbers to represent the relevance score for each clip in QVHighlights (Lei et al., 2021), and a start and end time for the relevant video span in Charades-STA. However,

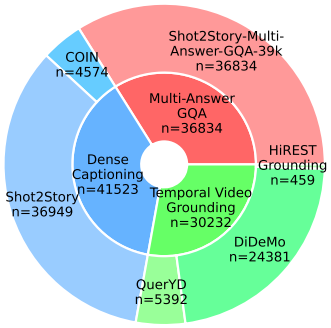


Figure 3: Data Distribution of MMDuetIT.

	QVHighlights mAP/HIT@1	Charades-STA R@IoU=0.5/0.7	YouCook2 SODAc/CIDEr/F1
Video-LLaMA	11.3/15.6	2.7/1.2	0.0/0.0/0.1
VideoChat-Embed	13.1/18.1	3.2/1.4	0.2/0.6/3.4
VideoChatGPT	-	7.7/1.7	-
TimeChat	14.5/23.9	32.2/13.4	1.2/3.4/12.6
VTimeLLM	-	31.2/11.4	-
HawkEye	-	31.4/14.5	-
VTG-LLM	16.5/33.5	33.8/15.7	1.5/5.0/17.5
VideoLLM-Online	-	-	0.4/0.9/5.8
LLaVA-OV-TC	17.6/32.9	33.1/12.4	1.9/3.3/**21.8**
LLaVA-OV-VT	19.0/40.0	36.5/12.3	2.5/6.7/**14.0**
MMDuet (Ours)	**31.3/49.6**	**42.4/18.0**	2.4/5.7/19.2
+ rm. prev. resp.	-	-	**2.9/8.8**/21.7

Table 1: Zero-shot performance on highlight detection, temporal video grounding, and dense video captioning. All models uses 7B LLMs.

for LLaVA-OV-TC and LLaVA-OV-VT, despite using different prompts as input, we were still unable to instruct the model to output a sequence of scores as in (Ren et al., 2023). Therefore, we follow the method of Charades-STA to instruct the model to output a related span, and assign the score to 1 for clips within this span and 0 otherwise. MMDuet uses the relevance score min-max normalized to $[0, 1]$ as the score in QVHighlights, and to classify whether this frame is relevant and calculate frame-level IoU in Charades-STA.

Since the relevance head provides a relevance score immediately after each frame, its prediction cannot leverage the context from subsequent video frames. To mitigate this limitation, we smooth the relevance score sequence. Specifically, we set each frame's smoothed relevance score as the mean value of its original score, the relevance scores of the preceding w frames and the following w frames, where w is the window size. We set $w = 2$ for QVHighlights and $w = 6$ for Charades-STA. Results are shown in Table 1. We observe that, compared to the baselines, MMDuet exhibits a significantly greater improvement in performance on QVHighlights. This indicates that traditional VideoLLMs struggle with generating a long sequence of relevance scores using a text-based form or identifying multiple related video segments in its text-based responses, whereas MMDuet's approach of directly assigning relevance scores to each frame circumvents this issue. For VideoLLM-Online, we instruct it to reply with "start" / "end" at the start / end time of the target clip following the examples given in its paper but it does not follow the instructions despite trying different wordings, so we are not able to report its performance.

w is robust to different values Though the w is empirically set for the results in Table 1, we also find that within a fairly large range of w, the performance of MMDuet is robust and consistently outperforms all baseline models. Detailed results are listed in Appendix C.1.

6.2 Dense Video Captioning

We test dense video captioning performance on YouCook2 (Zhou et al., 2017), a challenging task that requires models to output the caption, start point and end point for about 8 steps in a minutes-long cooking video. Baseline models output the start time, end time and caption for each step in the text-based form. For MMDuet, since this task requires the model to continuously identify important actions from the video and output periodically, we employ a heuristic method to determine whether a model response should be output after each frame (`need_response` function in Section 4.2). We sum up the informative score for each frame as the video plays. When the sum reaches a threshold s (we set $s = 2$), the model generates a response right after this frame as the caption for that step, and then we reset the sum to 0 to start a new round of sum.

However, MMDuet cannot directly predict when a step starts or ends just by this video-text duet interaction format, as the model is unable to determine whether a frame is the beginning of a step without observing enough subsequent content. To get the start and end time for each step as required by this task, we adopt a simple workaround: we use the time of the previous response and the current response as the start time and end time for a step. If two adjacent steps have the same caption, we merge them into one step. This workaround is also applied on VideoLLM-Online.

It has been a long-lasting problem that LLMs tend to repeat previously-generated content (Xu et al., 2022), and we find that this problem is especially severe in dense video captioning. It indicates that VideoLLMs are probably generating captions relying on text shortcuts rather than the video content. We have attempted common solutions such as repetition penalty (Keskar et al., 2019), which though is still sub-optimal. Since the responses from MMDuet are separated across multiple turns, we find that simply removing previously generated turns from the context ("rm. prev. resp." for short) by not appending their attention keys and values to the KV Cache alleviates this issue, leading to a significant improvement in performance. However, this simple trick is not applicable to "whole-video" format baselines, as if the latest words are removed from the KV Cache, it will remain the same as before generating the latest words and the model will generate the same words again, despite some minor changes due to random sampling. In contrast, for MMDuet new video contents continuously bring new KV Cache and drive the conversation forward.

As shown in Table 1, MMDuet does not show significant improvements on F1 metric, likely due to the simple solution we use to derive the start and end time based on responses. Even so, the CIDEr and CODA_c metric (inaccurate predicted time spans can have negative effects on these metrics) of MMDuet is still higher than all baselines, indicating that MMDuet outperforms baselines in terms of text quality, possibly due to its facilitation to information retrieval discussed in Section 1.

s is robust to different values We also find that the threshold s is quite robust across a wide range of from 1 to 3, and we can use different s to suit various downstream tasks especially in such zero-shot setting. Detailed results are listed in Appendix C.1.

6.3 Multi-Answer Grounded Video QA

To align closely with the widely-used streaming video comprehension scenario, we propose MAGQA that requires a model to generate answers at multiple necessary positions of a video. Different from conventional Video QA in which one question corresponds to only one answer, In MAGQA, a question corresponds to multiple turns of answers, and these turns are derived from different video segments. Therefore, this task requires the response to be accurate and in time. Though under the video-text duet interaction format users may raise arbitary number of questions at any time, to ensure the feasibility of evaluation, in this experiment we assume that the user raises only one question at the beginning of the video, and leave the extension to multiple questions as future work.

As this task is a newly-proposed one, we introduce an "in-span score" metric, which uses LLMs to calculate the average similarity of pred answers and gold answers that falls into the same time span of response, to evaluate both the correctness and timeliness of model responses. A detailed description of this metric is in Appendix B.5. To prevent reproducibility issues due to potential changes of OpenAI API, besides GPT-4o-2024-08-06 (OpenAI, 2024), we also report the in-span score obtained using LLaMA 3.1 70B Instruct (Dubey et al., 2024) to calculate pred-gold similarities.

As MAGQA requires the answers to be both informative and related to the question, we set `need_response` as: if the sum of informative score and relevance score of a frame is larger than a threshold t, then the model needs to generate a response right after this frame. We also use the "rm. prev. resp." method in dense video captioning task introduced in Section 6.2. As baseline models are not capable of generating responses at specific positions in the video, we employ an output format the same as dense video captioning, *i.e.,* output the start time, end time, and predicted text for each turn after watching the entire video in both training and testing, and use the average of the start and end time as the response time. We also observe that for some cases the baseline models directly give one answer instead of generating multiple replies and their corresponding time spans, and we do not count these examples into the metrics when reporting results. **Note that this is a significantly simplified requirement than that of MMDuet**, as the MAGQA task simulates streaming video comprehension application scenario, which requires the model to respond as soon as the video plays to segments relevant to the question, which ensures that users can see the responses timely, rather than waiting until the entire video concludes before generating replies.

MMDuet has better performance than baselines and provides real-time replies. Results on the test set of Shot2story-MAGQA-39k are shown in the lelf half of Table 2. We provide results for different t as it represents a trade-off between inference time and performance: as t decreases from

Model	Real-Time?	original test set			5-time prolonged video test set		
		In-Span Score LLaMA/GPT	# turns (w/o. / w/. dedup)	time per example	In-Span Score LLaMA/GPT	# turns (w/o. / w/. dedup)	time per example
Baselines							
LLaVA-OV-TC	2718	2.77/2.64	4.1/2.2	**1.00**	1.67/1.62	7.6/2.4	1.00
LLaVA-OV-VT	2718	2.54/2.42	4.1/3.1	1.06	1.64/1.60	10.2/3.4	**0.99**
VideoLLM-Online	2714	1.33/1.26	1.3/1.1	0.44*	-	-	-
MMDuet (Ours)							
$t = 0.6$	2714	2.46/2.33	13.7/4.0	1.90	1.83/1.73	22.3/7.0	1.04
$t = 0.5$	2714	2.77/2.61	18.4/5.3	2.36	2.16/2.02	31.2/9.8	1.45
$t = 0.4$	2714	3.00/2.81	23.0/6.6	2.75	2.44/2.28	41.7/13.0	2.17
$t = 0.3$	2714	**3.13/2.93**	27.0/7.6	2.90	**2.63/2.45**	52.8/16.5	2.62

Table 2: Results on the test set of Shot2Story-MAGQA-39k with the rm. ass. turns method used. For the "time per example" column, the time used by "LLaVA-OV-VT" is set to 1, and the times for other rows are set as multiples of the time used by "LLaVA-OV-TC". *: Inference time of VideoLLM-Online is changed to gray and de-emphasized as it only generates one reply immediately after the question and is hardly helpful for answering the question, and thus we no longer evaluate it on the 5-times prolonged video test set.

Model	Acc
Flash-VStream	1.96
VLLM-Online	3.92
Dispider	25.34
MMDuet	**29.44**

Table 3: Performance on the Proactive Output task of StreamingBench.

Model	YouCook2
MMDuet	**2.9/8.8/21.7**
w/o rand. resp. pos.	2.1/7.3/19.0
w/o multi informative	**2.9**/8.0/16.5

Table 4: Ablation study on training methods.

0.6 to 0.3, the performance of MMDuet's real-time replies continuously rises and outperforms baselines with a simplified setting of providing non-real-time replies after watching the entire video. However, this is achieved at a cost of generating lots of duplicate replies with more inference time.

MMDuet performs much better than baselines on longer videos. Since the average video length of the test set of Shot2story-MAGQA-39k is only 16.9 seconds, to demonstrate MMDuet 's real-time QA capabilities on longer videos we use a simple approach to make videos in the test set longer: we splice the video with 4 other videos randomly selected from the test set in random order to prolong the video to approximately 5 times longer by padding with videos irrelevant to the question. Results on the prolonged videos are shown in the right half of Table 2. When the videos are long, it becomes harder for baseline models to output correct time spans for the answers which results in low in-span scores, while MMDuet is more likely to generate correct answers at the right time.

6.4 Proactive Output on StreamingBench

To further demonstrate the timeliness of the replies of MMDuet, we also report results on the Proactive Output task of StreamingBench (Lin et al., 2024). StreamingBench evaluates VideoLLMs in real-time, streaming video understanding tasks. Specifically, for the "Proactive Output" task, a question is considered as correctly answered if a reply is raised by the model within two seconds when a certain scene that contains the answer appears. Results in Table 3 show that MMDuet outperforms all Streaming or Proactive MLLMs (Zhang et al., 2024a; Chen et al., 2024; Qian et al., 2025). Refer to Appendix C.2 for more details and baselines.

6.5 Ablation Studies

We conduct ablation studies on YouCook2 dense video captioning to assess two empirical yet important findings for effectively training the informative head in data construction: randomly inserting the response at a position from 50% to 75% of the corresponding video segment (rand. resp. pos.), and setting informative head's label to *TRUE* for all frames between 50% of the segment and the response time (multi informative). When "rand. resp. pos." is disabled, the response is always inserted at the end of the corresponding segment. When "multi informative" is disabled, only the informative label of the frame right before the response is set as *TRUE*. As illustrated in Table 4, disabling either method negatively impact MMDuet's performance, which shows the importance of carefully handling the response time and informative labels.

7 Conclusion

In this paper, we first formalize the video-text duet interaction format. We collect MMDuetIT for training models to follow the video-text duet interaction

format. Based on MMDuetIT we train MMDuet, a model with significant improvements on various time-sensitive tasks and is able to automatically decide when to response in a real-time manner. We believe such improvements can be a substantial step towards building powerful and useful video comprehension applications.

Limitations

We acknowledge that there is much room for improvement which should be addressed in future research: (1) Some hyperparameters (*e.g.,* the `need_response` criterion) are required during inference. However, we have shown that this criterion is quite robust across different thresholds. (2) Information from subsequent frames is not incorporated when generating in-time responses for the current frame, especially for the live-streaming video that indeed has unpredictable future frames. It can be crucial in some scenarios, such as determining the start of an action. (3) Slow inference speed. A better inference process is needed for avoid generating duplicate responses. (4) Real-time response datasets with longer live-streaming videos are required to be collected to better fit the real-world application scenarios.

Acknowledgement

This work is supported in part by the State Key Laboratory of General Artificial Intelligence.

References

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024. Videollm-online: Online video large language model for streaming video. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18407–18418.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Laurens Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline C. Muzzi, Mahesh Babu Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, An-

drew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

J. Gao, Chen Sun, Zhenheng Yang, and Ramakant Nevatia. 2017. Tall: Temporal activity localization via language query. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5277–5285.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Kahsay Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. 2021. Ego4d: Around the world in 3,000 hours of egocentric video. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990.

Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. 2024. Vtg-llm: Integrating

timestamp knowledge into video llms for enhanced video temporal grounding. *ArXiv*, abs/2405.13382.

Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. 2023. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *ArXiv*, abs/2312.10300.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2017. Localizing moments in video with natural language. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5804–5813.

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

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. 2023. Vtimellm: Empower llm to grasp video moments. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14271–14280.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 706–715.

Jie Lei, Tamara L. Berg, and Mohit Bansal. 2021. Qvhighlights: Detecting moments and highlights in videos via natural language queries. *ArXiv*, abs/2107.09609.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *ArXiv*, abs/2408.03326.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024b. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *ArXiv*, abs/2407.07895.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2023. Mvbench: A comprehensive multi-modal video understanding benchmark. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22195–22206.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *ArXiv*, abs/2411.03628.

Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. 2024. St-llm: Large language models are effective temporal learners. *ArXiv*, abs/2404.00308.

Andreea-Maria Oncescu, João F. Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. 2021. Queryd: A video dataset with high-quality text and audio narrations. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2265–2269.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-11-13.

Maxime Oquab, Timoth'ee Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2023. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193.

Rui Qian, Shuangrui Ding, Xiao wen Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2025. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. *ArXiv*, abs/2501.03218.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2023. Timechat: A time-sensitive multimodal large language model for long video understanding. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14313–14323.

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

Zhihong Shao, Damai Dai, Daya Guo, Bo Liu (Benjamin Liu), Zihan Wang, and Huajian Xin. 2024.

Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *ArXiv*, abs/2405.04434.

Quan Sun, Yuxin Fang, Ledell Yu Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *ArXiv*, abs/2303.15389.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. 2024a. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. 2024b. Internvideo2: Scaling video foundation models for multimodal video understanding. *ArXiv*, abs/2403.15377.

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

Yuxuan Wang, Yueqian Wang, Pengfei Wu, Jianxin Liang, Dongyan Zhao, and Zilong Zheng. 2024d. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. 2023. Can i trust your answer? visually grounded video question answering. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13204–13214.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *ArXiv*, abs/2206.02369.

Zenan Xu, Xiaojun Meng, Yasheng Wang, Qinliang Su, Zexuan Qiu, Xin Jiang, and Qun Liu. 2023. Learning summary-worthy visual representation for abstractive summarization in video. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5242–5250. International Joint Conferences on Artificial Intelligence Organization. Main Track.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. 2024. Qwen2 technical report. *ArXiv*, abs/2407.10671.

Abhaysinh Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Ouguz, Yasher Mehdad, and Mohit Bansal. 2023. Hierarchical video-moment retrieval and step-captioning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23056–23065.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024a. Flash-vstream: Memory-based real-time understanding for long video streams. *ArXiv*, abs/2406.08085.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024b. Video instruction tuning with synthetic data.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2017. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*.

A Data Quality Check of Shot2Story-MAGQA-39k

We sample 100 examples (with 290 answers) from our test set for manual quality assessment. Among

the sampled examples, we find 1 example with a question unanswerable from the video, 5 examples have 6 answers (2.1%) that contradict the video content, and 5 examples have 7 answers (2.4%) unrelated to the question. Overall, manual quality assessment shows that above 95% data of our test set belongs to the high quality, which confirms the potential value of using Shot2Story-MAGQA-39k to benchmark models. The reason for the high quality is when the video captions are provided, generating questions and answers based on these text captions is a very simple task for advanced LLMs like GPT4o. However, we also find that in 21 examples, the video contains additional information that is not covered in the answers. This is because some questions are very general, like "What scene is the video displaying?", and describing scenes in videos elaborately has been a long-lasting challenge for annotating video datasets.

B Details of Training and Inference

B.1 Data Reformat Process of MMDuetIT

In Section 5.1 we briefly introduced how the annotations for offline dense captioning / QA are converted into image-text interleave interactive format for training MMDuet. Here we elaborate more details and the reasons of this design:

Choices of insertion We randomly sample a position from 50% to 75% time duration for the corresponding video segment, and insert the caption at that position as a model response. Here we introduce some randomness in the insertion position to prevent the model from developing a bias or a shortcut such as responses can only be generated at some specific positions. The earliest and latest time for inserting responses, *i.e.,* at the 50% and 75% place of segment duration, are empirically chosen, as it works well in our preliminary study. We avoid inserting responses too early like in the first half of duration, because it is unfeasible to generate responses related to this video segment at a very starting point. It is reasonable that some further observations are required to gain a more comprehensive understanding of it. We also avoid inserting responses too late like in the last one-fourth duration, as we hope the model to output a response as soon as it has a sufficient understanding of the segment, rather than wait until the disappearance of the segment. It thereby improves the timeliness of the whole interaction between users and videos, especially when the user can still watch the segment as well as perceive the content of the model response talking about it.

Creating informative labels We also create labels for the informative head in dense captioning tasks. According to the previous paragraph, the model can not have a comprehensive understanding of this video segment until it has viewed a sufficient portion of the segment (50% in this case). Meanwhile, once the caption has been generated as model response, we assume that the remaining frames in this video segment no longer provide new information that is not covered in the caption. Therefore, we set the informative head's label to *TRUE* for frames between 50% of this segment and the insertion point of the response, and set labels to *FALSE* for the other frames.

B.2 Training Hyperparameters

LLaVA-OneVision uses SigLIP-Large (Zhai et al., 2023) as the vision encoder, and converts an image with 384×384 into $24 \times 24 = 576$ tokens. In the official settings of LLaVA-OneVision (Li et al., 2024a), when encoding videos, the visual tokens corresponding to each frame are spatially downsampled to $12 \times 12 = 144$ tokens using a pooling operation with a size of 2. However, this number of tokens is also too large when training and inference with long videos. To address this, we further modified the pooling size to 4, resulting in $7 \times 7 = 49$ tokens per frame.

We set the maximum number of frames sampled from each video to 120 in the training process, which is constrained by the memory of our GPUs. The sampling frame rates are set to different numbers for different video sources to ensure that for the vast majority (>90%) of videos, video length (in seconds) $\div$ sampled frame per second (fps) ≤ 120. For the videos that are too long, we only keep the first 120 frames (and the conversation turns that are inserted within the first 120 frames), and discard the subsequent contents. Specifically, the sampled frame per second (fps) is set as: 2 for videos from Shot2Story (Han et al., 2023) and DiDeMo (Hendricks et al., 2017), 0.5 for COIN (Tang et al., 2019) and QueryD (Oncescu et al., 2021), and 0.33 for HiREST$_{grounding}$ (Zala et al., 2023).

The projector, the relevance head, the informative head and LoRA (Hu et al., 2022) weights of the LLM (add to all attention proj. layers and FFN

```
# Input:
#  system_prompt
#  video: list of frames
#  fps: frames per second to sample from video
#  user_turns: list of (time, text) sorted by time
# Output:
#  model_turns: generated list of (time, text)

model_turns = []
v_inf_list, v_rel_list = [], []
kv_cache = model(system_prompt)
time = 0
for frame in video:
  if len(user_turns) and time>= user_turns[0].time:
    kv_cache = model(kv_cache, user_turns[0].text)
    user_turns = user_turns[1:]
  kv_cache, v_inf, v_rel = model(kv_cache, frame)
  v_inf_list.append(v_inf)  # informative score
  v_rel_list.append(v_rel)  # relevance score
  if need_response(v_inf_list, v_rel_list):
    kv_cache, response = model.generate(kv_cache)
    model_turns.append((time, response))
  time += 1 / fps
```

Listing 1: Inference Process of MMDuet

layers) are trained, while other parameters of the model are frozen. More training hyperparameters are listed in Table 5.

B.3 Pseudo Code of the Inference Process

We provide a python-style pseudo code of the inference process in Listing 1.

B.4 Inference Settings

Videos from different sources are also sampled with different fps during inference. Specifically, we set the maximum number of frames sampled from each video to 400, and fps to 2 for videos from Shot2Story (Han et al., 2023) and Charades-STA (Gao et al., 2017), 1 for videos from QVHighlights (Lei et al., 2021), and 0.5 for videos from YouCook2 (Zhou et al., 2017). For a few videos in YouCook2 that are even longer than 400(frames) $\div$ 0.5(fps) $= 800$ seconds, we uniformly sample 400 frames from this video to ensure that information from the latter part of the video is not truncated. This inference setting is consistent across MMDuet, LLaVA-OV-TC, and LLaVA-OV-VT.

Hyper-parameter	value
batch_size	1
gradient_acc_steps	8
learning_rate	2e-5
warmup_ratio	0.05
lora_r	16
lora_alpha	32
attn_implementation	sdpa

Table 5: Hyper-parameters used for training MMDuet.

B.5 Details of the In-Span Score

Suppose the model prediction has P answers, each answer has a prediction time $time_p$ and prediction text $pred_p$, $p = 1, 2, \ldots, P$. The ground truth has Q answers, each answer has a ground truth start time $start_q$, a ground truth end time end_q, and a ground truth text $gold_q$, $q = 1, 2, \ldots, Q$. First, we use an LLM to calculate a relevance score from 1 to 5 between each answer in prediction $pred_p$ and ground truth $gold_q$: $S = \{s_{p,q}\} \in \mathcal{R}^{P \times Q}$. For each ground truth answer q, we select the predicted answers with predicted time in ground truth time span: $\mathcal{P}_q = \{p \mid time_p \in [start_q, end_q]\}$, and use the average score between the ground truth answer and the selected predicted answers as the score for this ground truth answer: $score_q = \frac{1}{|\mathcal{P}_q|} \sum_{p \in \mathcal{P}_q} s_{p,q}$ if $|\mathcal{P}_q| > 0$. If $|\mathcal{P}_q| = 0$ (no predicted answer falls in this ground truth span), $score_q$ is set to 1. Finally, we calculate the average score of all ground truth answers as the final in-span score of this example: $in\_span\_score = \frac{1}{|Q|} \sum_{q=1}^{|Q|} score_q$.

C More Experimental Results

C.1 Hyperparameter Sensitivity

We list the experiments using different window size w for temporal grounding in Fig. 4 and threshold s for dense captioning in Fig. 5.

C.2 Details of the Proactive Output Experiment

More results and baselines are listed in Table 6. For results of models without streaming abilities (Proprietary MLLMs & Open-Sourced VideoLLMs), we follow the evaluation method of (Lin et al., 2024) and (Qian et al., 2025): We gradually extend the input video one second at a time and ask the model with the question "Is it the right time to output?". If the model responds with "Yes.",

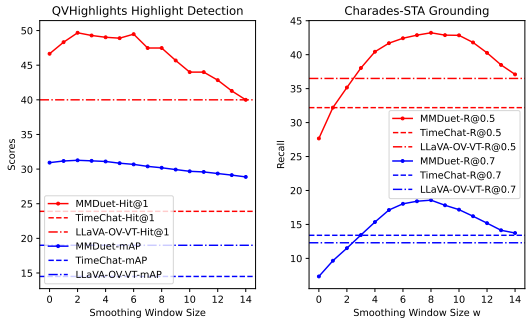


Figure 4: Performance on temporal video grounding and highlight detection with different w.

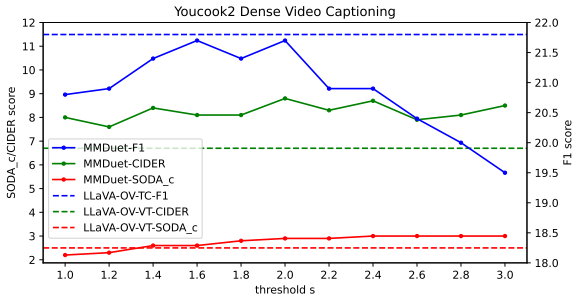


Figure 5: Performance on dense video captioning with different s.

this moment is recorded as the predicted output timestamp. For MMDuet, we use the time of the first reply after the user question is input as the predicted output timestamp. For examples that MMDuet does not provide any reply at all, we consider them as failing cases and the difference between ground truth output time and predicted output time is recorded as $+\infty$.

Model	Acc	Model	Acc
Proprietary MLLMs			
Gemini 1.5 pro	45.10	GPT-4o	56.86
Claude 3.5 Sonnet	64.71		
Open-Sourced VideoLLMs			
LLaVA-OV	29.55	Qwen2-VL	22.73
MiniCPM-V 2.6	22.22	LLaVA-NeXT-Video	18.18
InternVL2	40.91	LongVA	15.91
Streaming MLLMs			
Flash-VStream	1.96	VideoLLM-Online	3.92
Dispider	25.34		
MMDuet $t = 0.3$	29.44	MMDuet $t = 0.4$	31.85
MMDuet $t = 0.5$	26.61	MMDuet $t = 0.6$	18.95

Table 6: Performance of more baselines and MMDuet on the Proactive Output task of StreamingBench with different t.

D Example Inputs for Each Task in MMDuetIT

Example inputs for each task for training and inference are listed in Table 7. The dense video captioning user input is selected from one of the following sentences:

Please concisely narrate the video in real time.
Help me to illustrate my view in short.
Please simply describe what do you see.
Continuously answer what you observed with simple text.
Do concise real-time narration.
Hey assistant, do you know the current video content? Reply me concisely.
Simply interpret the scene for me.
What can you tell me about? Be concise.
Use simple text to explain what is shown in front of me.
What is the action now? Please response in short.

The temporal video grounding user input is selected from one of the following sentences (where "%s" denotes the caption to localize):

%s What segment of the video addresses the topic '%s'?
At what timestamp can I find information about '%s' in the video?
Can you highlight the section of the video that pertains to '%s'?
Which moments in the video discuss '%s' in detail?
Identify the parts that mention '%s'.
Where in the video is '%s' demonstrated or explained?
What parts are relevant to the concept of '%s'?
Which clips in the video relate to the query '%s'?
Can you point out the video segments that cover '%s'?
What are the key timestamps in the video for the topic '%s'?

E Qualitative Study

We list some examples of dense video captioning on videos with several minutes in length and contains many actions in Figs. 6 to 8, and examples of multi-answer grounding video question answering (MAGQA) in Figs. 9 to 11. For LLaVA-OV-TC and LLaVA-OV-VT, we directly list their generated outputs. For MMDuet, we list the numerical order (in round brackets), time (in square brackets) and content (in the second line) for each turn. If a line contains multiple numerical orders and times, this indicates that these turns have the same content, which is shown in the following line. To help readers to identify the position of these turns within the video, we also annotate the numerical order of the turns at the corresponding timestamps in the video stream.

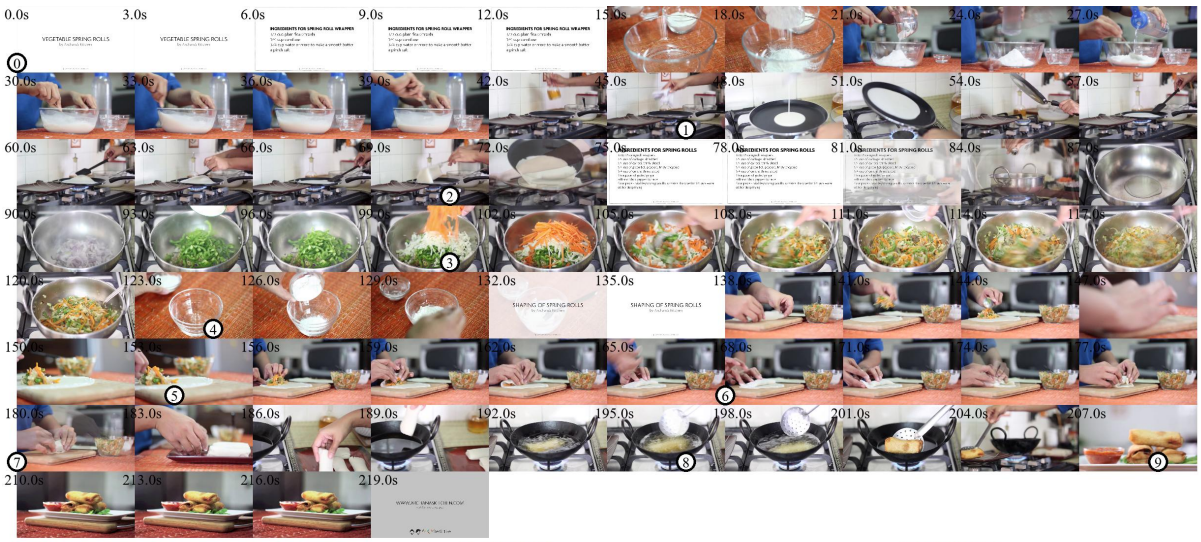
When handling long videos for dense video captioning, baseline models often recall only part of the video or generate repeated content, failing to

Dense Video Captioning	<im_start>system A multimodal AI assistant is helping users with some activities. Below is their conversation, interleaved with the list of video frames received by the assistant. <im_end> <im_start>user (A Dense Video Captioning Query)<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> <im_start>assistant A person pulls a knife from a black bag.<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> <im_start>assistant A man in a hat and red clothes speaks with a dagger, and a tree behind him.<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> . . . (More stream and assistant turns)
MAGQA	<im_start>system A multimodal AI assistant is helping users with some activities. Below is their conversation, interleaved with the list of video frames received by the assistant. <im_end> <im_start>stream <frame><frame><frame> . . . <im_end> <im_start>user What happens during the basketball game?<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> <im_start>assistant Several players in white jerseys are celebrating by high-fiving each other.<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> <im_start>assistant A player in a white jersey makes a successful shot.<im_end> <im_start>stream <frame><frame><frame> . . . <im_end> . . . (More stream and assistant turns)
Temporal Video Grounding	<im_start>system A multimodal AI assistant is helping users with some activities. Below is their conversation, interleaved with the list of video frames received by the assistant. <im_end> <im_start>user (A Temporal Video Grounding Query)<im_end> <im_start>stream <frame><frame><frame> . . . <im_end>

Table 7: Input examples of different tasks during the training and evaluation phase of MMDuet.

provide a complete description of all steps in the video. In contrast, MMDuet, due to its ability to focus only on a small portion of the video content preceding each generation step and using the "rm. prev. turns" trick to avoid interference from previous turns, can provide more accurate and detailed video descriptions.

For the MAGQA task, due to the relatively short video length, baseline models can also locate video segments and answer questions effectively. The advantage of MMDuet in this task is its ability to provide answers in a real-time manner.



MMDuet (Ours)
(0) [0s]
USER: Please concisely narrate the video in real time.
(1) [44.0s]
ASSISTANT: pour the flour into the bowl
(2) [68.0s]
ASSISTANT: pour the dough into the pan
(3) [98.0s]
ASSISTANT: pour the filling
(4) [122.0s]
ASSISTANT: add the filling
(5) [154.0s]
ASSISTANT: put the filling on the dough
(6) [168.0s]
ASSISTANT: fold the spring roll
(7) [180.0s]
ASSISTANT: fold the dough
(8) [194.0s] (9) [206.0s]
ASSISTANT: fry spring rolls

LLaVA-OV-TC
10.0 seconds, pour the flour and water into the bowl.
23.0 seconds, stir the mixture.
45.0 seconds, fry the dough.
69.0 seconds, fry the vegetables.
87.0 seconds, pour the flour and water into the bowl.
100.0 seconds, stir the mixture.
110.0 seconds, fry the vegetables.
120.0 seconds, put the filling on the dough.
130.0 seconds, fold the dough.
140.0 seconds, fry the dough.
150.0 seconds, fry the dough.
160.0 seconds, fry the dough.
170.0 seconds, fry the dough.
180.0 seconds, fry the dough.
200.0 seconds, fry the dough.
210.0 seconds, fry the dough.

LLaVA-OV-VT
From 00 to 31, pour the batter.
From 31 to 42, fry the vegetables.
From 42 to 56, pour the batter.
From 56 to 70, wrap the spring rolls.

Figure 6: An example of dense video captioning with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.

MMDuet (Ours)
(0) [0s]
USER: Please concisely narrate the video in real time.
(1) [58.0s]
ASSISTANT: prepare the ingredients
(2) [82.0s]
ASSISTANT: pour the seasoning
(3) [102.0s] (4) [122.0s]
ASSISTANT: pour the ingredients into the bowl
(5) [140.0s]
ASSISTANT: pour the salad into the bowl
(6) [156.0s]
ASSISTANT: pour the sauce
(7) [170.0s] (8) [188.0s]
ASSISTANT: pour the salad into the bowl

LLaVA-OV-TC
10.0 seconds, cut vegetables.
34.0 seconds, prepare seasoning.
52.0 seconds, prepare seasoning.
68.0 seconds, prepare seasoning.
79.0 seconds, add seasoning to vegetables.
100.0 seconds, add seasoning to vegetables.
115.0 seconds, add seasoning to vegetables.
125.0 seconds, add seasoning to vegetables.
135.0 seconds, add seasoning to vegetables.
145.0 seconds, mix vegetables and seasoning.
160.0 seconds, add seasoning to vegetables.
170.0 seconds, add seasoning to vegetables.
180.0 seconds, mix vegetables and seasoning.

LLaVA-OV-VT
From 00 to 31, prepare the ingredients.
From 31 to 42, prepare the dressing.
From 42 to 56, add the dressing to the salad.
From 56 to 78, mix the salad.

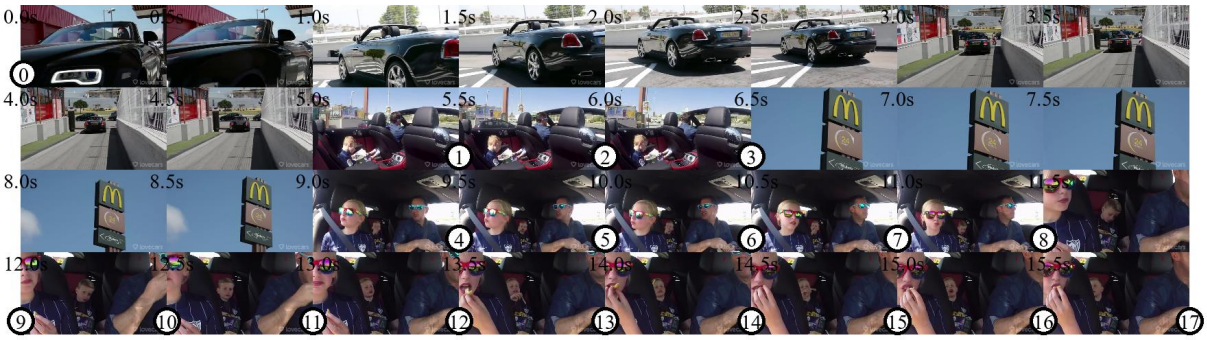
Figure 7: An example of dense video captioning with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.

MMDuet (Ours)
(0) [0s]
USER: What can you tell me about? Be concise.
(1) [44.0s] (2) [66.0s]
ASSISTANT: pour the noodles into the water
(3) [96.0s]
ASSISTANT: prepare the ingredients
(4) [126.0s]
ASSISTANT: pour the sauce into the pan
(5) [150.0s]
ASSISTANT: pour the oil into the pan
(6) [170.0s]
ASSISTANT: pour the raw materials into the pot
(7) [184.0s]
ASSISTANT: put the noodles into the pot
(8) [200.0s]
ASSISTANT: pour the noodles into the pot
(9) [214.0s]
ASSISTANT: add seasoning
(10) [228.0s]
ASSISTANT: pour sauce
(11) [240.0s]
ASSISTANT: pour the noodles into the pot
(12) [258.0s]
ASSISTANT: add seasoning
(13) [278.0s]
ASSISTANT: put the food on the plate
(14) [296.0s] (15) [308.0s]
ASSISTANT: add seasoning

LLaVA-OV-TC
10.0 seconds, Intro.
25.0 seconds, Showing noodles.
34.0 seconds, Boiling noodles.
67.0 seconds, Showing ingredients.
89.0 seconds, Showing sauce.
108.0 seconds, Cooking noodles.
130.0 seconds, Adding shrimp.
150.0 seconds, Adding tofu.
165.0 seconds, Adding eggs.
180.0 seconds, Stirring noodles.
200.0 seconds, Seasoning noodles.
220.0 seconds, Plating dish.
240.0 seconds, Outro.

LLaVA-OV-VT
From 00 to 31, pour the noodles into the water.
From 31 to 42, fry the ingredients in the pan.

Figure 8: An example of dense video captioning with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.



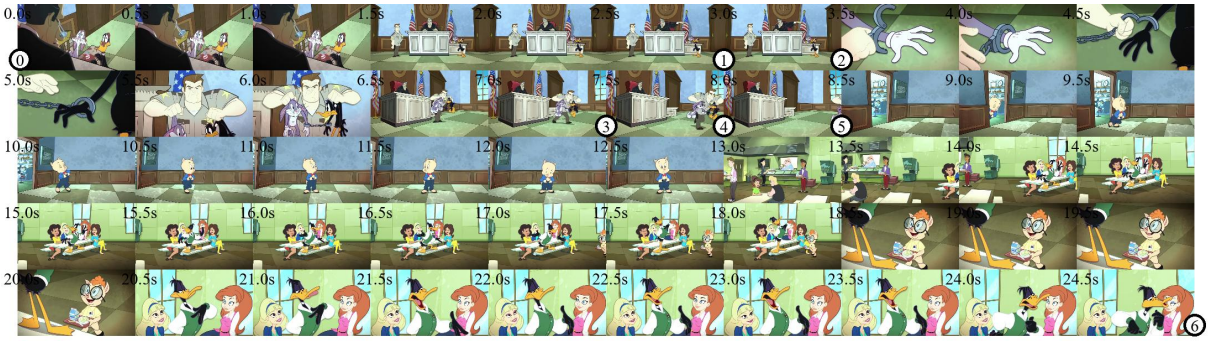
Question
What are the people doing in the car?

MMDuet (Ours)
(0) [0s]
USER: What are the people doing in the car?
(1) [5.5s] (2) [6.0s] (3) [6.5s]
ASSISTANT: A man is driving the car and a boy is sitting in the back seat.
(4) [9.5s]
ASSISTANT: A man and a child are sitting in the car, with the man driving and the child in the back seat.
(5) [10.0s]
ASSISTANT: A man and a child are sitting in the car, with the man driving and the child in the passenger seat.
(6) [10.5s]
ASSISTANT: A man and a child are sitting in the car, wearing sunglasses.
(7) [11.0s] (8) [11.5s]
ASSISTANT: A man and a child are sitting in the car, and the man is driving.
(9) [12.0s]
ASSISTANT: A man and a boy are sitting in the car, and the boy is wearing sunglasses.
(10) [12.5s] (11) [13.0s] (12) [13.5s] (13) [14.0s] (14) [14.5s] (15) [15.0s] (16) [15.5s]
ASSISTANT: A man and a boy are sitting in the car, and the boy is eating something.
(17) [16.0s]
ASSISTANT: A man and a child are sitting in the car, and the man is driving.

LLaVA-OV-TC
0.0 - 5.0 seconds, A man is driving a black convertible car with a child in the back seat.
5.0 - 6.5 seconds, A man is driving a black convertible car with a child in the back seat.
6.5 - 10.0 seconds, A man and a child are sitting in the back seat of a car, with the child eating something.
10.0 - 16.5 seconds, A man and a child are sitting in the back seat of a car, with the child eating something.

LLaVA-OV-VT
From 00 to 10, A man is driving a black convertible car with the top down.
From 10 to 20, A man is driving a black convertible car with the top down.
From 20 to 31, A man is driving a black convertible car with the top down.
From 31 to 41, A man is driving a black convertible car with the top down, and a child is sitting in the back seat.
From 41 to 51, A man is driving a black convertible car with the top down, and a child is sitting in the back seat.
From 51 to 94, A man is driving a black convertible car with the top down, and a child is sitting in the back seat.

Figure 9: An example of multi-answer grounded video question answering with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.



Question
What happens to the rabbit and the duck?

MMDuet (Ours)
(0) [0s]
USER: What happens to the rabbit and the duck?
(1) [3.0s] (2) [3.5s]
ASSISTANT: The rabbit and the duck are sitting at a table in a courtroom.
(3) [7.5s]
ASSISTANT: The rabbit and the duck are held by a policeman.
(4) [8.0s] (5) [8.5s]
ASSISTANT: The rabbit and the duck are held by a man in a police uniform and run away.
(6) [25.0s]
ASSISTANT: The duck and the rabbit are sitting at a table in a restaurant.

LLaVA-OV-TC
8.7 - 11.0 seconds, The rabbit and the duck are tied up by a man in a police uniform.

LLaVA-OV-VT
From 00 to 11, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 11 to 20, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 20 to 25, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 25 to 31, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 31 to 35, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 35 to 41, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 41 to 50, The rabbit and the duck are sitting at a table in a courtroom, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 50 to 60, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 60 to 65, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 65 to 70, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 70 to 75, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 75 to 81, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.
From 81 to 96, The rabbit and the duck are sitting at a table in a cafeteria, with the rabbit wearing a purple shirt and the duck wearing a black shirt.

Figure 10: An example of multi-answer grounded video question answering with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.

Question
What is the man doing with the crocodile?

MMDuet (Ours)
(0) [0s]
USER: What is the man doing with the crocodile?
(1) [7.5s]
ASSISTANT: The man is standing next to a crocodile on a table.
(2) [8.0s] (3) [8.5s]
ASSISTANT: The man is standing next to a crocodile on a table and talking to the camera.
(4) [9.0s] (5) [9.5s]
ASSISTANT: The man is standing next to a crocodile and talking to the camera.
(6) [10.0s]
ASSISTANT: The man is holding a crocodile and talking to the camera.
(7) [10.5s] (8) [11.0s] (9) [11.5s] (10) [12.0s]
ASSISTANT: The man is touching the crocodile's head.
(11) [12.5s] (12) [13.0s] (13) [13.5s] (14) [14.0s]
ASSISTANT: The man is cutting the crocodile's head.
(15) [14.5s] (16) [15.0s] (17) [15.5s] (18) [16.0s] (19) [16.5s] (20) [17.0s]
ASSISTANT: The man is cutting the crocodile's meat.

LLaVA-OV-TC
0.0 - 5.0 seconds, The man is talking to another man and gesturing towards the crocodile.
5.0 - 18.0 seconds, The man is standing in front of a table with a crocodile on it, touching it, and then cutting it with a knife.

LLaVA-OV-VT
From 16 to 27, The man is smiling and gesturing towards the crocodile.
From 27 to 42, The man is standing in front of a table with a crocodile on it.
From 42 to 94, The man is standing in front of a table with a crocodile on it, touching it, and then cutting it with a knife.

Figure 11: An example of multi-answer grounded video question answering with MMDuet, LLaVA-OV-TC and LLaVA-OV-VT.