LITA: Language Instructed Temporal-Localization Assistant

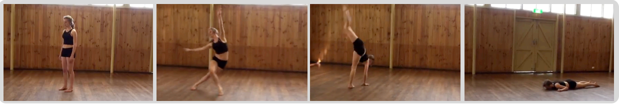
De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin,
Pavlo Molchanov, Zhiding Yu, Jan Kautz
NVIDIA
{deahuang,shijial,subhashreer,dannyy,pmolchanov,zhidingy,jkautz}@nvidia.com

Abstract

There has been tremendous progress in multimodal Large Language Models (LLMs). Recent works have extended these models to video input with promising instruction following capabilities. However, an important missing piece is temporal localization. These models cannot accurately answer the "When?" questions. We identify three key aspects that limit their temporal localization capabilities: (i) time representation, (ii) architecture, and (iii) data. We address these shortcomings by proposing Language Instructed Temporal-Localization Assistant (LITA) with the following features: (1) We introduce time tokens that encode timestamps relative to the video length to better represent time in videos. (2) We introduce SlowFast tokens in the architecture to capture temporal information at fine temporal resolution. (3) We emphasize temporal localization data for LITA. In addition to leveraging existing video datasets with timestamps, we propose a new task, Reasoning Temporal Localization (RTL), along with the dataset, ActivityNet-RTL, for learning and evaluating this task. Reasoning temporal localization requires both the reasoning and temporal localization of Video LLMs. LITA demonstrates strong performance on this challenging task, nearly doubling the temporal mean intersection-over-union (mIoU) of baselines. In addition, we show that our emphasis on temporal localization also substantially improves video-based text generation compared to existing Video LLMs, including a 36% relative improvement of Temporal Understanding. Code is available at: https://github.com/NVlabs/LITA

1. Introduction

Large language models (LLMs) [3, 7, 11, 28, 35, 36] have demonstrated impressive instruction following capabilities, and shown that language can be a universal interface for various tasks [7, 28]. These models can be further extended to multimodal LLMs to process language and other modalities, such as image, video, and audio [1, 24, 45].



Q: when does the woman's dance become the most **energetic** in the video?

A: Her dance is the most energetic between 7.37s and 12.81s when she does a **handspring**, which is more energetic than **standing** and **lying** on the floor.

Figure 1. Example to illustrate our proposed Reasoning Temporal Localization (RTL). Instead of directly querying about an event, questions in RTL require further reasoning to answer. Here, the model needs to compare all activities in the video to find the timestamps of the most energetic activity (*i.e*., handspring).

While most multimodal LLMs focus on images for visual content, several recent works introduce models that specialize in processing videos [20, 25, 27, 41]. These Video LLMs preserve the instruction following capabilities of LLMs and allow users to ask various questions about a given video. However, one important missing piece in these Video LLMs is *temporal localization*. When prompted with the "When?" questions, these models cannot accurately localize time periods, and often hallucinate irrelevant information [43]. Temporal localization is an important component that differentiates videos from images, and has been widely studied outside the context of instruction following LLMs [4, 10, 15, 33]. It is thus crucial for Video LLMs to have temporal localization capabilities.

We identify three key aspects that limit the temporal localization capabilities of existing Video LLMs: time representation, architecture, and data. First, existing models often represent timestamps as plain text (*e.g.* 01:22 or 142sec). However, given a set of frames, the correct timestamp still depends on the frame rate, which the model does not have access to. This makes learning temporal localization harder. Second, the architecture of existing Video LLMs might not have sufficient temporal resolution to interpolate time information accurately. For example, Video-LLaMA [41] only uniformly samples 8 frames from the entire video, which is insufficient for accurate temporal localization. Finally, temporal localization is largely ignored in the data used by existing Video LLMs. Data with timestamps are only a small subset of video instruction tuning data, and the accuracy of these timestamps is also not verified.

Our Approach. We address the aforementioned shortcomings of existing Video LLMs, and propose Language Instructed Temporal-Localization Assistant (LITA): (1) *Time Representation*: We introduce *time tokens* to represent relative timestamps and allow Video LLMs to better communicate about time than using plain text. (2) *Architecture*: We introduce *SlowFast tokens* to capture temporal information at fine temporal resolution to enable accurate temporal localization. (3) *Data*: We emphasize temporal localization data for LITA. We propose a new task, Reasoning Temporal Localization (RTL), along with the dataset, ActivityNet-RTL, for learning this task.

The first important design of LITA is to use relative representation for time (*e.g.* first 10% of the video) instead of the absolute time representation with plain text (*e.g.* 01:22). We divide a given video into T equal length chunks, and introduce T time tokens `<1>` to `<T>` to represent the relative time location in the video. During training and inference, these time tokens can be easily encoded and decoded from plain text timestamps given the length of the video. The start and end timestamps are well-defined by the time tokens given only the input video. This is in contrast to plain text timestamps. Without the frame rate, the correct absolute timestamp is ill-defined given just the video frames.

The second important design of LITA is to use densely sampled input frames from videos. It is unlikely to achieve accurate temporal localization with only sparsely sampled frames. The challenge is that the LLM module inside Video LLMs cannot naively process large numbers of frames simultaneously due to context length limitation. Take LLaVA [24] as an example. Each image is converted to 256 tokens, which are then fed into its LLM module as input. If we directly feed 100 frames to the LLM module, then that is $256 \times 100 = 25600$ tokens, which is already over the max context length for some LLMs [6, 36]. Inspired by the SlowFast architecture for videos [12], we instead consider two types of tokens, *fast tokens* and *slow tokens*, to address this efficiency issue. We generate *fast tokens* at a high temporal resolution to provide the temporal information, while keeping the tokens per frame at a low number. On the other hand, we generate *slow tokens* at a low temporal resolution, which allows us to use a higher number of tokens per frame to provide the spatial information.

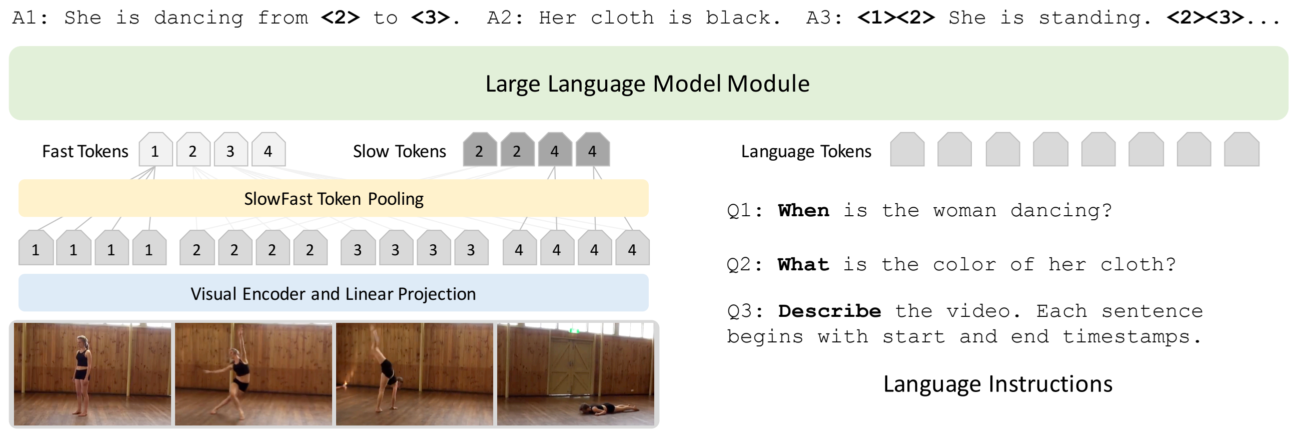
Finally, We emphasize temporal localization data for LITA. We include dense video captioning [16] and event localization [39] in instruction tuning of LITA. These tasks include human annotated timestamps to promote accurate temporal localization. In addition to leveraging existing data and tasks, we further propose a new task, Reasoning Temporal Localization (RTL), along with the dataset, ActivityNet-RTL, for training and evaluating this task. Answers to RTL questions can only be derived by utilizing world knowledge and temporal reasoning. Figure 1 shows an example. To answer the question: "When does the woman's dance become the most energetic?" the model needs to first recognize the woman's dance moves in the video, then reason about the most active part, and finally temporally localize the event (*i.e.* handspring). In addition to the predicted timestamps, we further consider the *explanation* provided by the model. Thus our new task not only assesses temporal understanding, but also requires strong reasoning capabilities that are unique to LLMs.

For the challenging RTL task, LITA doubles baseline's performance for temporal metrics (mIOU, Precision@0.5), while providing much better explanations. In addition to enabling accurate temporal localization, we show that our emphasis on temporal understanding also improves LITA's core Video LLM capabilities. LITA substantially improves all scores on a benchmark for video-based question answering [27]. This includes a 22% relative improvement for Correctness of Information, and a 36% relative improvement for Temporal Understanding compared to existing Video LLMs.

2. Related Work

Multimodal Large Language Models. Large language models (LLMs) [7, 28] inspire recent works to address multimodal tasks by leveraging LLMs [37]. Some approaches add additional parameters inside LLMs, such as gated cross-attention layers [1, 2, 19] or adapter layers [42], to adapt it to process multimodal inputs. Several works, on the other hand, only use modules, such as projection layers or Q-Formers, to project outputs of visual encoders to the input space of LLMs [8, 24, 45]. Recent works further expand multimodal LLM to visual grounding tasks, such as detection [5, 23, 29] and segmentation [18]. The most related to ours is LISA [18], which extends referring segmentation to reasoning segmentation. We share the same spirit and propose Reasoning Temporal Localization to jointly evaluate reasoning and temporal understanding.

Video Large Language Models. Building on the success of multimodal LLMs, several works extend image LLMs to Video LLMs [20, 25, 27, 41]. These works mainly use the approach of projecting visual tokens to LLMs' input space using projection layers [25, 27] or Q-Formers [20, 41]. While these models show promise in descriptive questions and instructions, they still lack temporal localization capabilities. LITA is designed to address this shortcoming,



A1: She is dancing from **<2>** to **<3>**. A2: Her cloth is black. A3: **<1><2>** She is standing. **<2><3>**...

Q1: **When** is the woman dancing?

Q2: **What** is the color of her cloth?

Q3: **Describe** the video. Each sentence begins with start and end timestamps.

Figure 2. Overview of LITA. The input video frames are first encoded into visual tokens (numbered by frame), which are further processed by two pathways. The Fast Token pathway averages all the tokens in a frame to maintain a high temporal resolution. The Slow Token pathway sparsely samples frames to maintain a larger number of tokens per frame to provide spatial information. Timestamps are converted to time tokens <1> to <T>. This is important for better temporal localization learning. Various video tasks on the right can be converted to natural language visual question answering (Q1-3 and A1-3) to jointly optimize LITA.

while also improving downstream video tasks. Concurrent works [13, 21, 22, 30, 32] further improve existing VideoLLMs. The most related concurrent works to ours are VTimeLLM [13], TimeChat [32], and Momentor [30]. These works also aim to address temporal localization of Video LLMs. We further introduce the reasoning aspect to temporal localization.

Temporal Localization in Videos. The goal of temporal localization is to pinpoint activities within untrimmed video sequences on a temporal scale [39]. The target activities can be predefined action classes [9, 15] or events described by natural language [4, 44]. Our goal of video temporal understanding is also related to various video tasks, such as dense video captioning [14, 16, 40] and action segmentation [17, 26, 34]. Models for these temporal tasks can have quite different design. Instruction following Video LLMs like LITA provide a way to unify these frameworks.

3. Language Instructed Temporal-Localization

LITA enables temporal localization for Video LLMs by: (1) relative time representation with the *time tokens*, (2) *SlowFast tokens* to capture temporal information at fine temporal resolution, (3) multi-task training that includes accurate timestamps. We will first introduce the overall architecture and discuss further details of individual components.

3.1. Architecture

An overview of LITA is shown in Figure 2. We build on Image LLMs. In particular, we select LLaVA due to its simplicity and effectiveness [24]. Note that LITA does not depend on the specific underlying Image LLM architecture and can be easily adapted to other base architectures.

Given a video, we first uniformly select T frames and encode each frame into M tokens. T should be large enough to support the desired granularity of temporal localization. $T \times M$ is a large number that usually cannot be directly processed by the LLM module. Thus, we then use SlowFast pooling to reduce the $T \times M$ tokens to $T + M$ tokens.

The slow and fast tokens are projected by a linear layer and concatenated with the text tokens to use as input to the LLM module. The text tokens (prompt) are processed to convert any referenced timestamps to specialized time tokens (<1> to <T>). All the input tokens are then jointly processed by the LLM module sequentially. We fine-tune the entire model with our reasoning temporal localization data (Section 4) along with other video tasks, such as dense video captioning and event localization. LITA learns to use time tokens instead of absolute timestamps. For temporal localization, we can then directly ask LITA the "When" questions (*e.g.* "When is she dancing?"). LITA would respond with time tokens (*e.g.* "She is dancing from <2> to <3>."), which can then be converted to timestamps given the video length.

3.2. Time Tokens

We use a relative time representation instead of absolute timestamps in LITA. As shown in Figure 2, the LLM module can only see the visual tokens (slow and fast) and the language tokens (text prompt). There is not enough information in this input space for the LLM module to infer the absolute timestamp because the frame rate is not known to the model in advance. A better way is to represent timestamps relative to the video length, thus removing the dependency on the frame rate. We divide the video into T chunks and use T specialized time tokens `<1>` to `<T>` for timestamps. Given a continuous timestamp τ and video length L, τ can be easily converted to time token `<t>`, where $t = \texttt{round}(\tau(T-1)/L) + 1$, and conversely `<t>` can be converted back to $\tau = L(t-1)/(T-1)$. While this does introduce discretization error, it greatly simplifies the time representation with LLMs. Relative timestamp is also used in other temporally heavy video tasks, such as dense video captioning [40].

Given this time representation, many video tasks related to temporal localization can be transformed into language instructions and answers. For example, dense video captioning can be achieved by prompting the model with "Describe the video. Each sentence begins with start and end timestamps." (`Q3` and `A3` in Fig. 2). Standard event localization is also transformed to "When does X happen?" (`Q1` and `A1` in Fig. 2). We also incorporate standard video question answering (`Q2` and `A2` in Fig. 2). More details are discussed in Section 3.4.

3.3. SlowFast Visual Tokens

We have discussed how we discretize time in videos into T steps in order to make Video LLMs better at reasoning about time. Still, the visual input should match the temporal resolution T in order to achieve effective temporal processing. Ideally, one would need at least T frames to temporally localize events with the resolution T. However, naively feeding all T frames into the LLM module could be computationally prohibitive. In our experiment, we use $T = 100$ and $M = 256$ (CLIP ViT-L-14 [31]). This leads to 25600 tokens per video.

Inspired by SlowFast models [12], we consider two pathways to pool the $T \times M$ tokens for T frames. The first is densely sampled *fast tokens* to provide temporal information. We obtain T fast tokens from T frames by averaging all the tokens belonging to the same frame. The second is the sparsely sampled *slow tokens* to maintain better spatial information. We select a spatial downsampling ratio of s, and uniformly select s^2 frames from the video. For each selected frame, we perform a $s \times s$ spatial average pooling to the M tokens, which lead to $\frac{M}{s^2}$ slow tokens per frame. This leads to a total $M = \frac{M}{s^2} \times s^2$ slow tokens. We use $s = 2$ in our experiments. This leads to a total of $T + M$ tokens to represent a video instead of $T \times M$ tokens.

3.4. Training Tasks

In addition to architecture, training tasks and data also play an important role for LLMs. We emphasize temporal localization data and train LITA with the following five tasks: (1) dense video captioning [16], (2) event localization [39], (3) video question answering [38], (4) natural language visual question answering [24], and (5) our proposed reasoning temporal localization. Temporal localization is a crucial component for three out of the five tasks (1, 2, and 5).

We now introduce each task in order. The first three tasks are standard video tasks and equip LITA with basic video understanding:

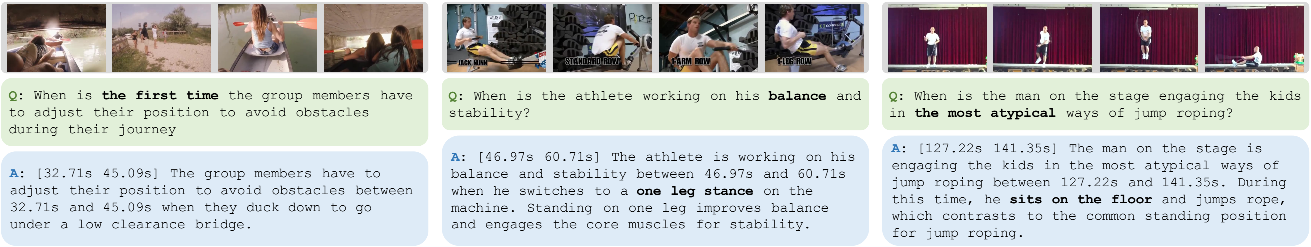
Dense Video Captioning. In dense video captioning [16], each video is described by a set of sentences, and each sentence comes with the start and end timestamps of the event. Each sentence in dense video captioning can thus be represented as: `<start time> <end time> SENTENCE`. We then sort all sentences by its start time, and directly concatenate all sentences and timestamps. One example prompt to the model for this task is: "Provide a detailed description of the given video. Each sentence should begin with the start and end timestamps." Other prompts are included in the supplementary materials.

Event Localization. In event localization, the goal is to temporally localize the event described by a sentence. We use a simple answer format: `<start time> <end time>`. One example prompt for this task is: "When does "`SENTENCE`" happen in the video? Answer the question only using start and end timestamps."

Video Question Answering. The question answering task is already represented as language instructions. However, answers in existing question answering datasets often consist of a single word or phrase because models for this task might not be able to generate longer text. We follow Liu *et al.* [23] and append the following prompt to the question: "Answer the question using a single word or phrase." The goal is to provide the context for short answers so that it affects the model's text generation less.

Natural Language Visual Question Answering. Training with the above three tasks gives LITA video understanding capabilities. However, we observe that models trained with only these tasks often provide short answers and lack natural language conversation capabilities. We thus also train LITA with natural language visual question answering or visual instruction tuning datasets [24]. The goal is to improve the natural language conversation of LITA. We find that mixing instruction tuning datasets [24] with standard video tasks improves LITA's conversation quality while maintaining good video understanding.

Reasoning Temporal Localization. Finally, we also train LITA with our reasoning temporal localization task (details in Section 4). The answer to a reasoning temporal localization question consists of two parts: timestamps and explanation. We find it challenging for models to simultaneously output both of them without any example. Nevertheless, with some training data, LITA quickly pick up reasoning and temporal localization, and provide both the timestamps and explanation of its reasoning in answers.



Q: When is **the first time** the group members have to adjust their position to avoid obstacles during their journey

A: [32.71s 45.09s] The group members have to adjust their position to avoid obstacles between 32.71s and 45.09s when they duck down to go under a low clearance bridge.

(a) Temporal Reasoning

Q: When is the athlete working on his **balance** and stability?

A: [46.97s 60.71s] The athlete is working on his balance and stability between 46.97s and 60.71s when he switches to a **one leg stance** on the machine. Standing on one leg improves balance and engages the core muscles for stability.

(b) World Knowledge

Q: When is the man on the stage engaging the kids in **the most atypical** ways of jump roping?

A: [127.22s 141.35s] The man on the stage is engaging the kids in the most atypical ways of jump roping between 127.22s and 141.35s. During this time, he **sits on the floor** and jumps rope, which contrasts to the common standing position for jump roping.

(c) Complex Reasoning

Figure 3. Examples from our ActivityNet-RTL dataset. RTL questions ask about events that are not explicitly described. The model needs to utilize reasoning or its world knowledge to answer. This is in contrast to standard temporal localization, which directly asks about the event of interest. For example in (c), standard temporal localization might directly ask: “when does the man sit on the floor?”

4. Reasoning Temporal Localization

We now discuss further details of the proposed Reasoning Temporal Localization (RTL) task. Standard temporal localization does not fully leverage the potential of Video LLMs. One impressive aspect of LLMs is its reasoning capabilities. LLMs can even answer complex questions that involve multi-step reasoning. Therefore, we propose the RTL task to utilize both of Video LLMs’ temporal understanding and reasoning capabilities.

4.1. Problem Definition

In *reasoning* temporal localization, the query is still a “when” question that asks about the start and end timestamps of an event. The key difference compared to the standard temporal localization task is that the target event is not directly described in the question, and can only be inferred by reasoning and using world knowledge of the model. The answer to such a question thus consists of two parts: (1) the start and end timestamps of the target event, and (2) an explanation of the reasoning process the model goes through to derive the timestamps.

Some examples are shown in Figure 3 to better illustrate the idea. The answer format is: `[start end] Explanation`. In Figure 3(a), the model not only has to localize “adjust their position to avoid obstacles,” but also needs to temporally reason about which instance happened earlier in the video. In Figure 3(b), instead of directly asking about “one-leg row,” it asks about the workout targeting balance and stability. The model thus needs to utilize its knowledge of what kind of exercises are good for balance and stability. Finally, there are questions that require multi-step reasoning. In Figure 3(c), the question asks about “the most atypical ways of jump roping,” which requires the model to understand what is typical and atypical for jump roping, and then temporally find the most atypical time period. A standard temporal localization task, in contrast, would just ask when the man is sitting on the floor.

4.2. ActivityNet-RTL Dataset

The above examples are from ActivityNet-RTL, a dataset curated by us for the Reasoning Temporal Localization (RTL) task. We build our dataset from the ActivityNet Captions dataset [16], which annotates multiple events described by sentences in a video, and all the events are temporally localized with start and end timestamps. Consider the following toy example:

```
[00:00-00:10] A woman is standing.
[00:12-00:30] The woman is dancing.
[00:32-00:36] The woman is sleeping.
```

We then use this as context and ask GPT-4 to generate temporal localization questions that require further reasoning to answer. We also ask GPT-4 to simultaneously generate the answer that includes the queried start and end timestamps, along with the explanation about the reasoning process.

Using the above toy example, by seeing that the woman has done three activities, one possible reasoning temporal localization question is to ask “When is she the least active?” Since sleeping is the least active out of the three activities, the target time period is 00:32-00:36, the period when she is sleeping. This illustrates how GPT-4 can still generate interesting questions without seeing the actual video. We annotate few-shot examples for GPT-4 as in previous works to improve the generation quality [24]. All of our prompts are included in the supplementary materials.

Training Set Generation. For our training set, we directly use the results generated by GPT-4 with 10,009 videos from the training split of ActivityNet-Captions. This leads to 33,557 Reasoning Temporal Localizaiton question-answer pairs. By inspecting the GPT generated results, we find that most of the questions are valid temporal localization questions given the context. The main shortcoming is that not all question-answer pairs require reasoning. Sometimes GPT-4 generates questions that directly ask about events that are already described by the dense video captions. However, we do hope that LITA can also answer these standard temporal localization questions correctly using natural language. We thus leave these questions in the training set.

Evaluation Set Curation. On the other hand, the evaluation set requires manual efforts otherwise we would end up with many non-reasoning questions. We start from the GPT-4 generated questions using a subset of the ActivityNet-Captions validation set, and manually remove non-reasoning questions. We also double check the timestamps and explanations in the answers. This leads to a total of 229 question-answer pairs for 160 videos.

4.3. Metrics

We consider three metrics for ActivityNet-RTL: mIOU, Precision@0.5, and GPT-4 Relative Scores. The first two metrics are for temporal localization, and the third metric is to evaluate the explanation capability. mIOU averages the intersection-over-union (IOU) between predicted and groundtruth start and end timestamps. Precision@0.5 measures the percentage of predictions that have over 0.5 IOU. We first average these two metrics per video, and then average over all videos in the evaluation set. This avoids over-weighting videos and time periods with more questions, as some time periods in a video have multiple questions.

To evaluate the quality of the explanation, we follow the evaluation pipeline of LLaVA [24] and leverage GPT-4 for evaluation. GPT-4 is asked to evaluate the helpfulness, relevance, accuracy, and level of details of the explanations, and give a score from 1 to 10. We ask GPT-4 to evaluate both the predicted and groundtruth explanations, and normalize the score for the prediction by the score of the groundtruth. For this metric, we directly average over all question-answer pairs as the explanations could be quite different even for questions about the same time period in the same video.

5. Experiments

We evaluate LITA with both temporal localization and video tasks that do not involve temporal localization because most existing Video LLMs cannot handle temporal localization. In addition to our proposed Reasoning Temporal Localization, we further evaluate LITA on Video-based Text Generation Performance Benchmarking proposed by Maaz *et al*. [27]. This provides a holistic evaluation of LITA as a Video LLM and not just for temporal localization.

5.1. Implementation Details

Architecture. We uniformly sample 100 frames from a video, and use 100 time tokens `<1>` to `<100>` to represent timestamps. We use CLIP-L-14 [31] as the visual encoder, and Vicuna [6] as the LLM module. We follow LLaVA's architecture and train a single linear layer for projection [24]. We use 4 frames for slow tokens and use average pool window $s = 2$. With 1 fast token per frame, this leads to a total of $100 + \frac{256}{4} \times 4 = 356$ tokens per video.

Training Datasets. We discussed training tasks in Section 3.4. We now discuss the training datasets for each task. For dense video captioning and event localization, we use the training splits of ActivityNet-Captions [16] and YouCook2 [44], which combine to around 11k videos. The event localization dataset can be generated from the dense video captioning dataset by using the caption as query and the timestamps as target. For video question answering, we use NExT-QA [38] as it contains more complex questions. For image instruction tuning, we use LLaVA-150K [24]. For reasoning temporal localization, we use the training split of our ActivityNet-RTL, which builts on the training split of ActivityNet-Captions.

Training Setup. For each of the five tasks, we randomly select 100K samples with replacement (total 500K). We then use batch size 128 and learning rate 2e-5 to train for 4k iterations. The training takes around 13 hours for 13B and 9 hours for 7B models using 8 A100 GPUs. The linear projection is initialized with the LLaVA pre-trained weights.

Table 1. Results on ActivityNet-RTL. LITA substantially outperforms all baselines for all metrics. This shows the importance of our design choices. Interestingly the temporal localization accuracy also improves as we scale the model from 7B to 13B.

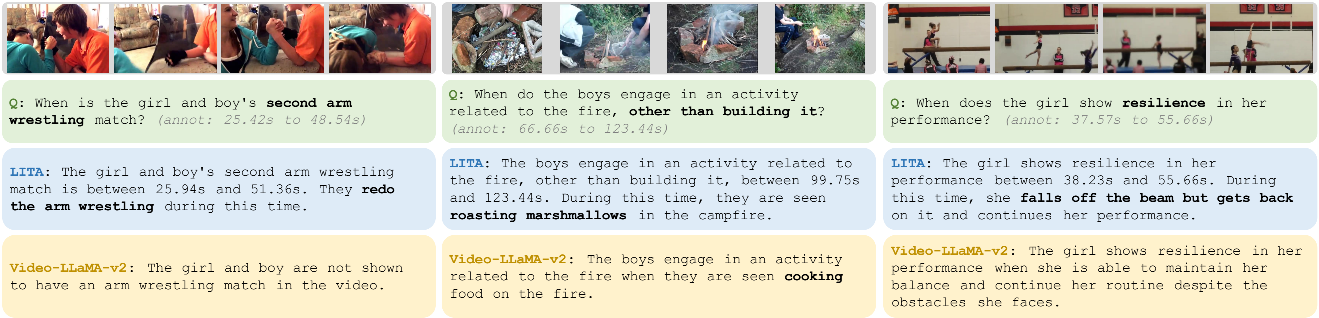
Model	Size	mIOU	P@0.5	Score
LITA-7B	7B	24.1	21.2	44.0
Video-LLaMA-v2 [41]	13B	–	–	32.1
Video-ChatGPT [27]	13B	–	–	38.8
Slow Tokens Only	13B	14.6	11.8	32.2
SlowFast Tokens	13B	17.5	14.5	34.1
LITA-13B	13B	**28.6**	**25.9**	**46.3**

5.2. Reasoning Temporal Localization Evaluation

We first evaluate on the newly proposed ActivityNet-RTL for reasoning temporal localization. Please refer to Section 4 for dataset details and metrics. We use "P@0.5" for Precision@0.5 and "Score" for the GPT evaluation score for explanations. Other than variations of our model, we include Video-LLaMA-v2 [41] and Video-ChatGPT [27] for comparison. We observe that most of their outputs on ActivityNet-RTL omit any timestamps and thus mIOU and Precision@0.5 become absolute. Therefore, for these methods we only evaluate the GPT-Score. In addition, we ablate the following variations for LITA, all of which are trained with the same five training tasks as LITA:

- *"Slow Tokens Only"* samples 4 frames from the video, and computes 64 tokens per frame. It does not use the fast tokens, and it also does not use time tokens for relative timestamps. This can be seen as naively implementing a Video LLM via the LLaVA architecture.

- *"SlowFast Tokens"* additionally includes fast tokens (*i.e.*



Q: When is the girl and boy's **second arm wrestling** match? *(annot: 25.42s to 48.54s)*

LITA: The girl and boy's second arm wrestling match is between 25.94s and 51.36s. They **redo the arm wrestling** during this time.

Video-LLaMA-v2: The girl and boy are not shown to have an arm wrestling match in the video.

Q: When do the boys engage in an activity related to the fire, **other than building it**? *(annot: 66.66s to 123.44s)*

LITA: The boys engage in an activity related to the fire, other than building it, between 99.75s and 123.44s. During this time, they are seen **roasting marshmallows** in the campfire.

Video-LLaMA-v2: The boys engage in an activity related to the fire when they are seen **cooking** food on the fire.

Q: When does the girl show **resilience** in her performance? *(annot: 37.57s to 55.66s)*

LITA: The girl shows resilience in her performance between 38.23s and 55.66s. During this time, she **falls off the beam but gets back** on it and continues her performance.

Video-LLaMA-v2: The girl shows resilience in her performance when she is able to maintain her balance and continue her routine despite the obstacles she faces.

Figure 4. Qualitative results on ActivityNet-RTL. Overall, LITA not only more accurately localizes events in the video, but also provides sensible explanations with more details. In the first example, LITA correctly identifies the second arm wrestling. In the second example, LITA provides further details that they are roasting marshmallows. In the third example, LITA impressively recognizes that the girl "falls off the beam but gets back" and explains that this shows resilience. These are in contrast to the more generic answers by Video-LLaMA-v2.

Table 2. Video-based Text Generation Benchmarking results. LITA significantly outperforms existing Video LLMs including Video-LLaMA-v2 [41] and Video-ChatGPT [27] on all evaluated aspects. This shows that LITA not only enables accurate temporal localization, but also generally improves video understanding for Video LLMs.

Model	Correctness	Detail	Context	Temporal	Consistency	Average
LLaMA-Adapter	2.03	2.32	2.30	1.98	2.15	2.16
Video-LLaMA	1.96	2.18	2.16	1.82	1.79	1.98
Video-LLaMA-v2	2.36	2.42	2.74	1.83	2.12	2.29
VideoChat	2.23	2.50	2.53	1.94	2.24	2.29
Video-ChatGPT	2.40	2.52	2.62	1.98	2.37	2.38
LITA	**2.94**	**2.98**	**3.43**	**2.68**	**3.19**	**3.04**

1 token per frame) compared to "Slow Tokens Only". This improves the architectural design for representing video, and should allow better temporal processing.

- *"LITA"* is our full model that further includes time tokens to better represent timestamps compared to "SlowFast Tokens." We consider two model sizes, 7B and 13B, for LITA to understand the effect of different model sizes.

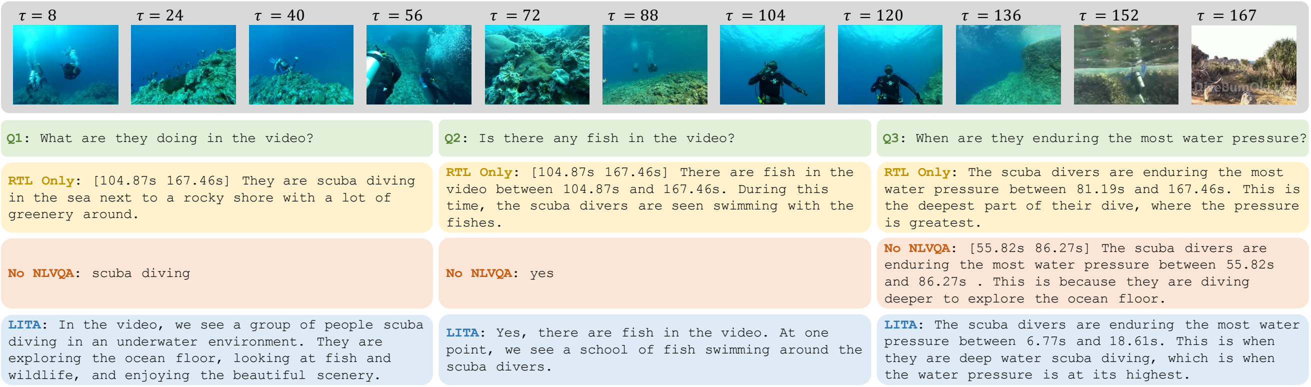
Importance of Our Model Components. Results on ActivityNet-RTL are shown in Table 1. All metrics are averaged over three trials. LITA substantially outperforms all baselines for all metrics, and almost double mIOU and P@0.5 when compared to "Slow Tokens Only", which is considered as a naive extension of Image LLMs to Video LLMs. "Score" is assisted by GPT-4 to evaluate the quality of the explanation provided by the models. While we prompt GPT-4 to ignore the timestamps mentioned in the explanations, we observe that the scores are still slightly affected by the timestamps. Nevertheless, LITA provides better explanations for its reasoning due to an overall better video understanding.

LITA Gives Detailed and Accurate Explanations. Qualitative results are shown in Figure 4. In the first example, LITA correctly localizes when the second arm wrestling happens. On the other hand, Video-LLaMA-v2 does not recognize that there is arm wrestling in the video. In the second example, while both models identify that the correct activity is cooking with fire, LITA provides much more accurate details, including the fact that they are roasting marshmallows, and the start and end time for that. Finally, in the third example, LITA impressively recognizes the event where the girl "falls off the beam but gets back" and correctly responds that this shows resilience in her performance. In contrast, Video-LLaMA-v2 gives a generic answer given the "resilience" prompt.

Temporal Localization Scales with LITA Size. One interesting observation is that the temporal localization quality of LITA also improves as we scale up the model from 7B to 13B (Table 1). One might think that scaling the model only improves language understanding, but our result shows that this could also improve temporal reasoning and understanding for Video LLMs.

5.3. Video-Based Generation Evaluation

In addition to our proposed reasoning temporal localization, we further evaluate LITA on standard evaluation for Video LLMs to better compare with existing Video LLMs. We use the "Video-based Text Generation Performance Benchmarking" proposed by Maaz *et al*. [27]. This benchmark



Q1: What are they doing in the video?	Q2: Is there any fish in the video?	Q3: When are they enduring the most water pressure?
RTL Only: [104.87s 167.46s] They are scuba diving in the sea next to a rocky shore with a lot of greenery around.	RTL Only: [104.87s 167.46s] There are fish in the video between 104.87s and 167.46s. During this time, the scuba divers are seen swimming with the fishes.	RTL Only: The scuba divers are enduring the most water pressure between 81.19s and 167.46s. This is the deepest part of their dive, where the pressure is greatest.
No NLVQA: scuba diving	No NLVQA: yes	No NLVQA: [55.82s 86.27s] The scuba divers are enduring the most water pressure between 55.82s and 86.27s . This is because they are diving deeper to explore the ocean floor.
LITA: In the video, we see a group of people scuba diving in an underwater environment. They are exploring the ocean floor, looking at fish and wildlife, and enjoying the beautiful scenery.	LITA: Yes, there are fish in the video. At one point, we see a school of fish swimming around the scuba divers.	LITA: The scuba divers are enduring the most water pressure between 6.77s and 18.61s. This is when they are deep water scuba diving, which is when the water pressure is at its highest.

Figure 5. Qualitative results to analyze the effect of training tasks. τ is the continuous time in seconds. The questions are not from ActivityNet-RTL as we are also interested in questions other than reasoning temporal localization. "RTL Only" answers in the format for reasoning temporal localization, as this is the only training data it has. "No NLVQA" further includes standard video tasks and can correctly answer non-temporal questions (Q1 and Q2). However, the answers are short due to the short answers in existing video question answering datasets. LITA improves both reasoning and natural language conversation by utilizing NLVQA datasets.

selects videos from ActivityNet [4] and annotates question-answer pairs with natural language. This is in contrast to existing video question answering benchmarks, where the answers are often limited to a single word or phrase. In this benchmark, there are specific questions like: "What kind of tricks is the man performing while skating?" or generic questions like "Can you describe the video in detail?" For evaluation, the benchmark uses GPT to measure the following aspects of Video LLMs' answers: Correctness of Information, Detail Orientation, Contextual Understanding, Temporal Understanding, and Consistency.

The results are shown in Table 2. We compare LITA with LLaMA-Adapter [42], Video-LLaMA [41], VideoChat [20], and Video-ChatGPT [27]. Video-ChatGPT slightly outperforms other baselines including Video-LLaMA-v2. LITA significantly outperforms these two existing Video LLMs from all aspects. In particular, LITA achieves a 22% improvement for Correctness of Information (2.94 vs. 2.40) and a 36% relative improvement for Temporal Understanding (2.68 vs. 1.98). This shows that our emphasis of temporal understanding in training not only enables accurate temporal localization, but also improves the video understanding of LITA. We hypothesize that temporal localization enables the model to learn more details about videos, leading to improved video understanding. A similar observation was made for Image LLMs [5], where joint training with grounding tasks also improved non-grounding text generation.

5.4. Evaluating the Effects of Training Tasks

We have analyze the effect of our model components in Section 5.2, where we use all of our training tasks. Now we further analyze the effect of these training tasks for our model.

Table 3. Analysis of LITA's training tasks on ActivityNet-RTL. "RTL" is needed to predict both timestamps and explanations. "Video" includes standard video tasks to improve video understanding. "NLVQA" further improves reasoning and natural language generation capabilities of LITA.

Model	RTL	Video	NLVQA	mIOU	P@0.5	Score
RTL Only	✓	✗	✗	26.6	20.9	43.5
No NLVQA	✓	✓	✗	26.9	23.5	44.9
LITA	✓	✓	✓	**28.6**	**25.9**	**46.3**

We split the five tasks into three groups: RTL, Video, and NLVQA. "RTL" only includes the proposed reasoning temporal localization. Without training on our ActivityNet-RTL, the model would not output timestamps for us to evaluate temporal localization in many cases. The second group "Video" includes all the standard video tasks: dense video captioning, event localization, and video question answering. Using these video tasks, the model should learn better video understanding. Finally, "NLVQA" refers to the natural language visual question answering task to improve LITA's natural language conversation. We refer to training with just RTL as "RTL Only," and training with both RTL and Video but without NLVQA as "No NLVQA." Training with all three and thus all tasks is our proposed LITA.

Results. The results on ActivityNet-RTL are shown in Table 3 and qualitative comparisons are shown in Figure 5. By only training on RTL, "RTL Only" does not have enough supervision to learn the task. This is reflected in both timestamp accuracy (P@0.5) and explanation quality (Score). In addition, for non-temporal questions (Q1 and Q2) in Figure 5, the model cannot properly answer and always use the answer format for reasoning temporal localization.

By adding standard video tasks in Video to training, “No NLVQA” improves all metrics compared to “RTL Only”. Qualitatively in Figure 5, it can also answer `Q1` and `Q2` correctly. However, this capability mainly comes from the inclusion of video question answering datasets, which leads to short answers. LITA further improves by including NLVQA, which contains complex reasoning questions to improve its reasoning capabilities. More importantly, as shown in Figure 5, LITA now answers questions with natural language instead of short phrases.

6. Conclusion

We propose Language Instructed Temporal-Localization Assistant (LITA), which enables accurate temporal localization using Video LLMs. LITA demonstrates promising capabilities to answer complex temporal localization questions. At the same time, LITA substantially improves video-based text generation compared to existing Video LLMs even for non-temporal questions. This is the result of both our model design and data strategy. For model design, we propose time tokens to better represent the time and SlowFast tokens to efficiently process video inputs. Our experiments show the importance of these model components. For data strategy, we emphasize temporal localization data in training LITA. To achieve this, we propose the Reasoning Temporal Localization task and curate the ActivityNet-RTL dataset. Our results show that the inclusion of temporal localization data not only enables temporal localization for Video LLMs, but also improves the general video understanding capabilities. We further analyze our training data and show the benefits of incorporating standard video tasks and image instruction tuning data to train Video LLMs.

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. In *NeurIPS*, 2022.

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. OpenFlamingo: An open-source framework for training large autoregressive vision-language models. *arXiv:2308.01390*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

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

[5] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv:2310.09478*, 2023.

[6] 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.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *arXiv:2204.02311*, 2022.

[8] 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.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS Dataset: Collection, challenges and baselines. *IEEE Trans. PAMI*, 43(11):4125–4141, 2021.

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 130: 33–55, 2022.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019.

[13] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. *arXiv preprint arXiv:2311.18445*, 2023.

[14] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL-IJCNLP*, 2020.

[15] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

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

[17] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.

[18] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. *arXiv:2308.00692*, 2023.

[19] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. MIMIC-IT: Multi-modal in-context instruction tuning. *arXiv:2306.05425*, 2023.

[20] 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:2305.06355*, 2023.

[21] 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.

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

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

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

[25] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability, 2023.

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[27] Salman Khan Muhammad Maaz, Hanoona Rasheed and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. *ArXiv 2306.05424*, 2023.

[28] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.

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

[30] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*, 2024.

[31] 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 *ICML*, 2021.

[32] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. *arXiv preprint arXiv:2312.02051*, 2023.

[33] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

[34] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019.

[35] 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:2302.13971*, 2023.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[37] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal LLM. *arXiv:2309.05519*, 2023.

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

[39] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. UnLoc: A unified framework for video localization tasks. In *ICCV*, 2023.

[40] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2Seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, 2023.

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

[42] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv:2303.16199*, 2023.

[43] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv:2309.01219*, 2023.

[44] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

[45] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023.