M-LLM Based Video Frame Selection for Efficient Video Understanding

Kai Hu[1,*] Feng Gao[3] Xiaohan Nie[3] Peng Zhou[3] Son Tran[3] Tal Neiman[3]
Lingyun Wang[3] Mubarak Shah[2,3] Raffay Hamid[3] Bing Yin[3] Trishul Chilimbi[3]
Carnegie Mellon University [1] University of Central Florida [2] Amazon [3]
kaihu@cs.cmu.edu[1,*] shah@crcv.ucf.edu[2]
fenggo,nxiaohan,zhoupz,sontran,taneiman,lingyunw,raffay,alexbyin,trishulc}@amazon.com[3]

Abstract

Recent advances in Multi-Modal Large Language Models (M-LLMs) show promising results in video reasoning. Popular Multi-Modal Large Language Model (M-LLM) frameworks usually apply naive uniform sampling to reduce the number of video frames that are fed into an M-LLM, particularly for long context videos. However, it could lose crucial context in certain periods of a video, so that the downstream M-LLM may not have sufficient visual information to answer a question. To attack this pain point, we propose a light-weight M-LLM-based frame selection method that adaptively select frames that are more relevant to users' queries. In order to train the proposed frame selector, we introduce two supervision signals (i) Spatial signal, where single frame importance score by prompting an M-LLM; (ii) Temporal signal, in which multiple frames selection by prompting Large Language Model (LLM) using the captions of all frame candidates. The selected frames are then digested by a frozen downstream video M-LLM for visual reasoning and question answering. Empirical results show that the proposed M-LLM video frame selector improves the performances various downstream video Large Language Model (video-LLM) across medium (ActivityNet, NExT-QA) and long (EgoSchema, LongVideoBench) context video question answering benchmarks.

1. Introduction

Large Language Model (LLM) has revolutionized numerous domains in Artificial Intelligence (AI) [3, 34, 41, 44]. In the past year, Multi-Modal Large Language Model (M-LLM) has significantly improved the performances of Vision-Language Models (VLM) to an unprecedented level in tasks such as image captioning and Visual Question Answering (VQA) [2, 4, 16, 46]. To extend VQA to the temporal domain, video QA task requires a model to understand consecutive images along the time to answer a question, raising new challenges to the current M-LLM. One of the most critical challenges is the length of the video context, where a model needs to comprehend all frames in a video.

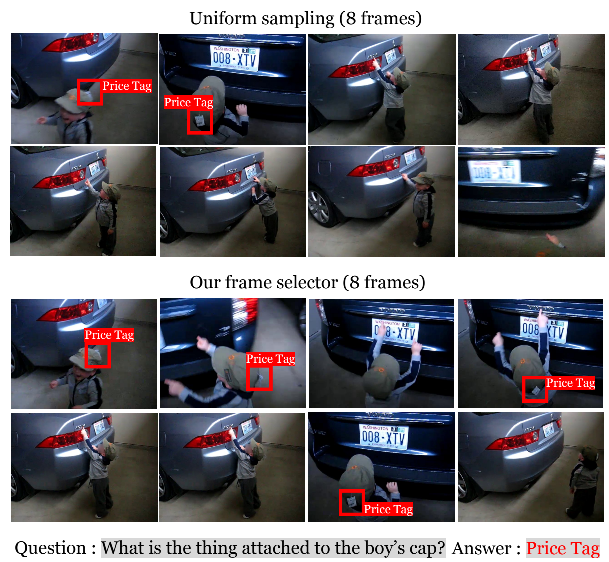

Figure 1. An example of our video frame selection for video QA. Compared to uniform sampling, ours has higher hit rate.

To balance the trade-off between the capability of understanding all video frames and the available context length in a specific M-LLM, conventional practices [55, 61] rely on uniform sampling of frames. Frames are extracted at predefined intervals regardless of their relevance to the specific questions. Although such methods maximize the coverage of a video in time-axis, it introduces insufficient visual information. The uniformly sampled frames may be irrelevant or redundant, meanwhile some important frames are ignored. This fact prevents the model from focusing on key events in the video and sometimes increases the computational costs. Generally in video QA [26, 59], some specific frames are more likely to contain information relevant to the question. The conventional one-size-fits-all approach limits both the performance and practicality of M-LLMs, especially when

*This work is done during this author's internship at Amazon.

working with long videos or resource-limited environments.

To address these limitations, a straightforward idea is to select frames instead of uniform sampling [35, 57]. Focusing on the frames that are most helpful for the question, we can use significantly less number of frames without sacrificing the quality of video understanding. Illustrated as an example in Figure 1, frames that contains the most informative visuals can help downstream model to answer the question. To implement the above idea, we propose a light weight frame selector that employs fine-tuned version of an LLM. It makes use of M-LLM's capability of multi-modality understanding to effectively capture the relevance between video frames and the question. We introduce two design choices to make the framework lightweight: (i) small LLMs are capable to understand complicated user questions; (ii) compress per video frame tokens to balance the long context trade-off.

Although video QA tasks often require a large number of tokens per frame to capture content details, we hypothesize that determining frame importance does not require excessive tokens. Instead of leveraging all visual tokens of the frames, we apply an aggressive pooling-based token reduction on each video frame, which significantly improves the computational efficiency and increases the number of frames that the lightweight LLM selector can ingest.

We also propose a training method for our LLM-based video frame selector. Since well-maintained video frame selection datasets for video QA are scarce, supervised training alone is not feasible. To address this issue, we adopt two pseudo-labeling strategies to estimate frame importance. The first strategy focuses on spatial understanding: we prompt a well-trained M-LLM using a video QA question, which then assigns importance scores to each frame. However, due to the limited context length of a M-LLM, it cannot evaluate all frames together from a temporal viewpoint. Therefore, our second strategy leverages an LLM to identify the top-k relevant frames using their captions. Instead of visual tokens, the LLM can analyze more frames simultaneously with caption. We integrate these two approaches to generate pseudo-labels that reflect frame importance relative to a specific question for effective training of the frame selector.

Our frame-selector employs a plug-and-play design that requires no additional fine-tuning of the downstream M-LLM. It reduces noise caused by irrelevant frames, allowing the video-LLM to focus more effectively on the relevant content. Additionally, the selector only needs to be trained once and can subsequently enhance the video QA performance of multiple M-LLMs. We demonstrate significant improvements with several popular M-LLMs on video question-answering tasks across various benchmarks, including short-to-medium context (ActivityNet, NExt-QA) and long context (EgoSchema, VideoMME) scenarios. In summary, our contributions are threefold:

- We propose a lightweight M-LLM-based adaptive video frame selector to for both efficient and stronger video QA performances of M-LLMs.
- We propose spatial and temporal pseudo-labeling to generate importance scores for video frame selector training.
- Our proposed method is plug-and-play friendly. We demonstrates comprehensive video QA improvements across popular M-LLMs with further fine-tuning.

2. Related Work

Multi-Modal Large Language Model (M-LLM) As Large Language Models (LLMs) continue to demonstrate impressive abilities in language comprehension and reasoning [1, 3, 41, 44], interest is growing within the computer vision community to explore their potential for handling multimodal inputs. Flamingo [2] demonstrates the capacity to process image and text inputs for a wide range of multi-modal tasks. BLIP-2 [17] introduces a Q-Former to map learned image features into the text embedding space of LLMs, while LLaVA [25] employs simple Multi Layer Perceptron (MLP) projector to align visual and textual features. Further research has focused on best practices for M-LLM, including areas such as dynamic high-resolution [6, 26], instruction-tuning data [19, 28], and different visual encoders [43, 51]. MM1 [33] and Idefics2 [15] provide comprehensive ablation studies on the design space of M-LLM.

Video M-LLMs As image-based M-LLM become more mature, research naturally extends to the video modality. Video-ChatGPT [31] and Valley [30] use pooling over video features to generate compact visual tokens for downstream M-LLM. Video-LLaVA [23] aligns images and videos before projection, allowing LLM to learn from a unified visual representation. Video-Teller [24] points out the importance of modality alignment in pre-training. PLLaVA [55] studies how different pooling of visual features affects downstream video question answering performance. Regarding long video inputs, LLaMA-VID [21] represents each frame with two tokens to reduce the overload of long videos while preserving critical information. MovieChat [39] propose an effective memory management mechanism to reduce the computation complexity and memory cost, enabling very long video with over 10K frames understanding. Weng et al. [52] proposes to extract video representations as sequences of short-term local features, and integrate global semantics into each short-term segment feature.

Video Frame Selection Before the rise of video M-LLM, language-aware video key-frame selection and localization had attracted great interest [7, 10, 27, 49]. Buch et al. [5] optimized an end2end pipeline that uses ground truth question-answering labels to select a single key frame for downstream tasks. Lu et al. [29] uses a CLIP-like paradigm to train a transformer for visual and text feature alignment. Qian et al. [36] trains a video clip proposal model and the downstream QA model in a iterative manner. Kim et al. [14] employs a

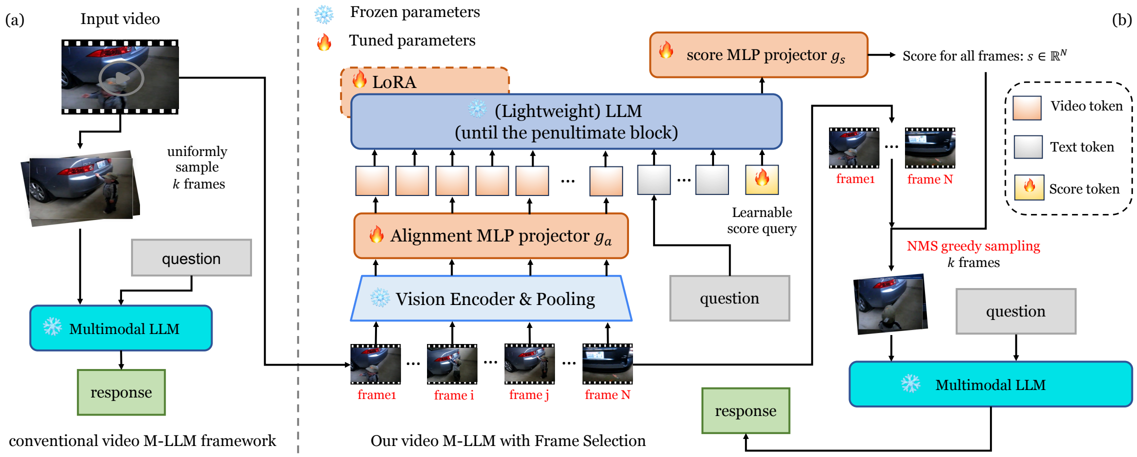

Figure 2. An illustration of the conventional *n*-frame video M-LLM framework and our video M-LLM framework with frame selection.

semi-parametric retriever to obtain the key frames by comparing similarities between frame and language features.

Video M-LLM Frame Selection Several works propose to select key frames to improve video QA performance. SeViLA [57] prompts an M-LLM to obtain a relevance score to each frame, and uses these scores to localize important frames. MVU [37] and adopts a similar approach as SeViLA. However, a critical limitation of this kind of methods [12, 22] is the absence of temporal reasoning. Each frame is assessed independently without contextual information from other frames. Furthermore, this method is expensive during inference. Importance score of every frame is estimated via a M-LLM. The longer the video the higher the cost. In contrast, our method outputs importance scores for all frames in a single pass with the compressed visual tokens, which reduces the computational costs and enables temporal reasoning. Koala [40] uses sparsely sampled key frames as conditions for processing subsequent visual tokens. ViLA [47] trains an end-to-end frame selection module to mask input frames. However, both of these methods lack text awareness: during inference, Koala's subsequent visual token processing and ViLA's frame masking do not account for the specific question posed about the video. Recent work such [48, 50] are not end-to-end methods. Han et al. [11] finds frame selection is helpful for video chain-of-thought reasoning.

3. Method

This section introduces our video frame selector designed for efficient video-LLM QA. Section 3.1 disccuss our motivation. Section 3.2 outlines the design details of the frame selector. Section 3.3 explains the generation of pseudo labels for training the frame selector. Section 3.4 describes the training process of the frame selector.

3.1. Rethinking Uniform Sampling in Video LLMs

A typical framework for video LLM An *n-frame* framework is widely adopted in existing research in video M-LLM [18, 55, 61]. The number of frames in the input video, denoted by T, is variable. For example, a 3-minute video at 30 frames per second (FPS) contains $T = 5400$ frames. The *n-frame* framework uniformly samples a fixed number of frames, $[x_1, x_2, \cdots, x_n]$, from the total T frames, where each frame $x_i \in \mathbb{R}^{H \times W \times 3}$, with $H \times W$ representing the frame resolution, and typically $n \ll T$. A pre-trained visual encoder f_v extracts visual features from n frames. These features are subsequently projected into the LLM's space using an alignment projector g_a and then flattened. Spatial pooling may also be applied to reduce the number of output tokens:

$$h_i = AvgPooling(g_a(f_v(x_i))), h_i \in \mathbb{R}^{m \times d} \tag{1}$$

where m is the number of tokens to represent a frame and d is the hidden dimension of the LLM. Let $Q \in \mathbb{R}^{l \times d}$ denote the input embedding of the input text question. The *n-frame* framework generates a response r as following:

$$r = LLM(h_1, \cdots, h_n, Q). \tag{2}$$

Uniform sampling is not optimal In the *n-frame* framework, as shown in Figure 2 (a), the input video is represented by $n \times m$ tokens where m is the number of visual tokens per frame. For example, in LLaVA-NeXT-Video [61], where $n = 32$ and $m = 12 \times 12$, this results in 4608 tokens. To reduce the computational cost of LLM inference, previous work has either chosen to reduce n with sliding Q-Former [18] or m [21, 55] by spatial pooling, in other words, reducing m. However, both of them ignore the importance of reducing n, the number of intake frames, before encoding and n could be large especially in long videos. Denser uniform sampling makes a larger n, thereby reducing the efficiency of the video M-LLM. On the other hand, sampling fewer frames risks omitting crucial information. For example, sampling 32 frames from a 3-minute video means taking one frame every 6 seconds, potentially missing actions that occur within shorter time windows. In fact, most questions about a video can be answered using only a limited number of key frames. Inspired by this factor, we argue that adaptive frame selection according to question is more efficient and effective than uniform frame sampling.

3.2. Design of the Frame Selector

An ideal frame selector should be able to understand complex questions and analyze temporal relevance in the video input. To achieve this, we fine-tune an LLM to function as the frame selector. The frame selector utilizes the base LLM's strong language comprehension and reasoning abilities to identify key frames in a video.

Similar to existing video M-LLM, our M-LLM-based frame selection method takes as inputs n sampled video frames along with the corresponding text-based question. Instead of generating an answer to the question, the frame selector identifies the most relevant frames for answering the question. Specifically, we make use of well-trained decoder only LLM as the frame selector to output an n-dimensional vector s as follows:

$$s = \textit{FrameSelector}(x_1, \cdots, x_n, Q) \in \mathbb{R}^n \tag{3}$$

where the i^{th} element in the vector s indicates the importance score of the i^{th} input frame. To achieve this, we append a learnable score query $q \in \mathbb{R}^{1\times d}$ to the end of all input tokens and use the concatenation of the visual tokens, text tokens and the score token as the LLM as input:

$$e_1, \cdots, e_n, e^Q, e^q = \textit{LLM}(x_1, \cdots, x_n, Q, q_{score}) \tag{4}$$

where e_i and e^q denote the intermediate output of x_i and q_{score} from the penultimate transformer block. Because the nature of causal attention, e^q aggregates information from all visual and text tokens. We then employ an MLP to exact frame importance information from the intermediate output of the score query from the penultimate transformer block:

$$s = \textit{MLP}(e^q), s \in \mathbb{R}^n \tag{5}$$

Figure 2 (b) presents an overview of the proposed architecture for the M-LLM-based frame selector. Instead of generating tokens that represent the frame selection, we append a learnable query vector at the end of the input sequence and supervisedly learn this query token. The hidden vector of this query token, e^q, then serves as the input to generate the n-dimensional importance vector s.

Select frames from the importance score After obtaining per-frame importance scores, we need to sample k frames for downstream video question-answering via a video M-LLM. Naively selecting the top k frames with the highest importance scores is suboptimal because neighboring frames within short time intervals often have similar scores. For example, if frame i has the highest importance score, the $i+1$ or $i-1$ frames usually have a closely high scores. The adjacent frame adds little additional information if frame i is

Algorithm 1 Greedy NMS sampling

Input: Importance score $s \in \mathbb{R}^n$, Number of frames selected k
Initialize the neighbor gap $\delta \leftarrow$ integer$({}^{n}/_{4k})$.
Initialize the selected index list $I_s = []$.
for step in $1 \cdots, k$ **do**
    (Greedy) Selected frame index $i \leftarrow \arg\max s$. Append index i to I_s.
    (NMS) Update the importance score: $s[j] = -1$ if $|i - j| \leq \delta$
$I_s \leftarrow$ sort(I_s)
Return: I_s

already selected. To address this issue, we use a greedy algorithm combined with non-maximum suppression (NMS) to select the most informative frames. The "Greedy" approach involves sequentially selecting the top k frames, without replacement, by choosing the frame with the highest importance score from the remaining set of frames. With "NMS", once a frame is selected, its neighboring frames are excluded as they contain similar information. The "NMS-Greedy" procedure is detailed in Algorithm 1. In practice, when selecting k frames from n frames, frame j is considered a neighboring frame of frame i if $|i - j| \leq ({}^{n}/_{4k})$.

Efficiency of the frame selector As discussed in Section 3.1, dense uniform sampling is inefficient for video M-LLM. However, we still employ dense uniform sampling for the frame selector.

Although dense uniform sampling is inefficient for video M-LLM, we keep using it at the beginning to maximally preserve the video information. To overcome the drawback of dense uniform sampling, we apply spatial pooling to reduces the token count per video frame before feeding the frames into the selector. In particular, the spatial pooling reduce the number of visual token to a smaller value, e.g., $m = 3 \times 3$ (9 tokens), which is substantially fewer than the tokens per frame used in existing video LLMs, e.g., $m = 12 \times 12$ (144 tokens). This design follows such an assumption: for video question answering, the model requires a substantial number of tokens per frame to capture visual details, but far fewer tokens are needed to determine whether a frame is important. A rough outline of the frame is sufficient. Our empirical results Table 7 also justify this assumption.

3.3. Pseudo Labels for the Frame Selector

To train the frame selector, we need supervision signals for the output importance scores $s \in \mathbb{R}^n$. Unfortunately, there is no existing dataset to label the frame level importance score for video QA, we propose two methods to generate pseudo labels for training the frame selector.

Spatial Pseudo Labels Previous works use M-LLMs to evaluate whether a video frame is relevant to a question [37, 57].

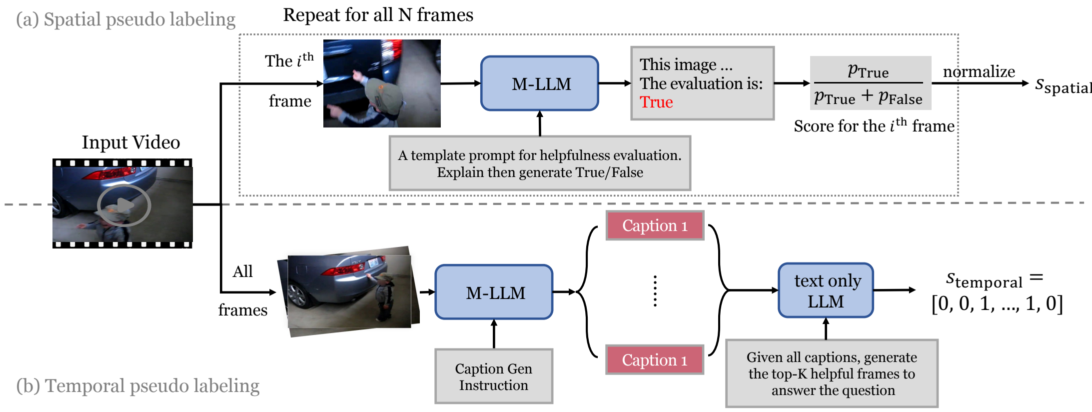

Figure 3. An illustration of the spatial and temporal pseudo labeling for the importance scores

A common practice is to prompt an M-LLM and ask if the video frame provides useful information to answer the question. The relevant score of the frame is the probability that the M-LLM generates the "Yes" token. In our experiments, we observe that this method does not always provide a reasonable estimation. Even if the model agrees the frame is relevant, the M-LLM may not generate the "Yes" token but other expressions depending on its text generation style. To address this issue, we apply chain-of-thoughts (CoT), asking the M-LLM to explain first then generate a Boolean evaluation (check appendix for the detailed prompt). This prompts allows the M-LLM to improve the evaluation quality by extra inference time reasoning. An ideal M-LLM should adhere to the instruction by generating either "True" or "False". In a few cases the model may fail to follow the instruction, and we manually append the text "Evaluation: True" to the end of the generated response. Thus we can always compute the probabilities of generating "True" and "False", denoted as p_{True} and p_{False} respectively. The importance score of the input frame is determined by

$$s = p_{\text{True}}/(p_{\text{True}} + p_{\text{False}}) \tag{6}$$

In our experimental setup, we uniformly sample $n = 128$ frames from the video and obtain the spatial pseudo labels for each frame independently. Let s_i denote the score for the i^{th} frame, we normalize the score vector as $s_i/\max_j s_j$. Figure 3 (a) shows the pipeline.

Temporal Pseudo Labels A significant limitation of single-frame evaluation is the lack of temporal reasoning. For example, considering the question: "**What did the man do after picking up his hat?**", the video content following the action of picking up the hat is crucial for answering this question. However, when generating the spatial pseudo labels, it only considers one frame without taking care of the temporal context. Therefore, the spatial labels don't know what occurs after the action.

To address this issue, we propose the temporal pseudo labeling. Since most of the publicly available M-LLMs are not able to consume a large number of image tokens, we alter to take advantages of the frame captions and use a LLM to reason over all captions. Specifically, we first obtain detail captions of all n frames by prompting a M-LLM. Second, we feed the captions of all frames together as well as the question to a strong text-only LLM. Then the LLM can temporally reason the helpfulness of all frames.

We find it challenging to generate floating-point scores for an extensive list of frames for an LLM. Consequently, we ask the model to produce a list of the index of most helpful frames (see appendix Section A for details). Frames included in this list are assigned a score of 1, while those excluded receive a score of 0. Figure 3 (b) shows the pipeline.

While temporal pseudo labels can capture temporal relations in videos, it may suffer from information loss and model hallucination due to its two-stage evaluation process. Therefore, we combine two methods into the final pseudo-labels by averaging the scores obtained from spatial and temporal pseudo labels.

3.4. Training of the Frame Selector

We consider a two-stage training procedure. In stage 1, we freeze the pre-trained vision and LLM backbones and train the parameters of the alignment projector g_a, the learned score query X_{score} and the score projector g_s. Figure 2 (b) shows the trainable modules in the red boxes and the frozen modules in the blue boxes. Stage 1 training is optimized over the two tasks alternatively. We use below two tasks:

- **Visual instruction following** Recall r in Equation 2 is the generated response from the LLM, the objective is the cross entropy loss between r and the ground truth response. This tasks trains the projector g_a to aligns the visual features with the pre-trained LLM embedding space.
- **Importance score prediction** Recall s in Equation 3.2 is the importance score for n frames, the objective is the binary cross entropy loss between s and the pseudo labels generated from Section 3.3. This task provides a good initialization for the score query and the score projector.

In stage 2, we only train the model with the importance score prediction task. Besides the alignment projector g_a, the learned score query q and the score projector g_s, we also include the Low Rank Adaptation (LoRA) weights of the LLM as the trainable parameters to adapt the LLM to the frame selection task.

4. Experiments

We begin by outlining our experimental setup in Section 4.1. Then we demonstrate that our frame selector improves the performance of well-trained M-LLMs without changing their parameters by selecting better frames in Section 4.2. We also conduct ablation studies to demonstrate the effectiveness and efficiency of our frame selection framework and results are in Section 4.3. At the end, we showcase some qualitative examples of frames selected from the video according to the question in Section 4.4.

4.1. Experiment Setup

Training data We compile the training dataset from three sources: 1) 800K data from VideoChat2 [20], 2) 125K data from the TimeIT dataset [38], and 3) 178K data from the LLaVA-Video-178K dataset [63]. For the visual instruction tuning task, we use the entire training dataset. For the importance score prediction task, we use 400K video QA data where the video length exceeds 5 seconds. **Pseudo label generation** we utilize Qwen2-VL-7B [46] to generate importance scores for each frame in the spatial pseudo labeling and concise captions for all frames in the temporal pseudo labeling. We use GPT-4o mini to propose the most helpful frames given the concise captions in the multi-frame evaluation. **Evaluation benchmarks** Since our framework selects frames for video question answering, we evaluated its performance on benchmarks consisting of relatively longer videos, including open-ended video QA on ActivityNet-QA [58], and multi-choice QA on NExT-QA [54], VideoMME [9], EgoSchema [32], LongVideoBench [53].

Implementation details We use the pre-trained SigLIP ViT-Large [60] as the visual encoder and Qwen2.5 1.5B [42] as the backbone of the LLM. We uniformly sample 128 frames from the video as the input for the visual encoder. The output from the visual encoder have a size of $128 \times 16 \times 16$. After the alignment projector and the spatial pooling layer, the size of the visual tokens is $128 \times 3 \times 3$. In Stage 1, we use a batch size of 128, a learning rate of 10^{-3}, and a warm-up ratio of 0.03 to train the model for two epochs. In Stage 2, a batch size of 128, a learning rate of 10^{-5}, a warm-up ratio of the first 3% iterations, and a cosine learning rate scheduler are used to train the model for five epochs.

4.2. Comparison with SOTA Video-LLMs

We choose two strong video M-LLMs, PLLaVA [55] and LLaVA-NeXT-video [62] and two (multi-)image based M-LLM Idefics [15] and Qwen2-VL [46], to be the baselines to illustrate how our frame selector enhances the video question-answering (QA) performance of these models. For each model, we compare the performance of using uniformly sampled frames as inputs versus using frames selected by our frame selector. The number of frames seen by the video M-LLM is the same during the comparison.

Model	Model Size	ActivityNet-QA
Video-ChatGPT [31]	7B	35.2 / 2.8
Chat-UniVi [13]	7B	46.1 / 3.3
LLaMA-VID [21]	7B	47.4 / 3.3
LLaMA-VID [21]	13B	47.5 / 3.3
Video-LLaVA [23]	7B	45.3 / 3.3
MiniGPT4-Video [4]	7B	46.3 / 3.4
SlowFast-LLaVA [56]	7B	55.5 / 3.4
SlowFast-LLaVA [56]	34B	59.2 / 3.5
Tarsier [45]	7B	59.5 / 3.6
Tarsier [45]	34B	61.6 / 3.7
PLLaVA [55]	7B	56.3 / 3.5
PLLaVA [55]	34B	60.9 / 3.7
LLaVA-NeXT-Video [62]	7B	53.5 / 3.2
LLaVA-NeXT-Video [62]	34B	58.8 / 3.4
PLLaVA + Selector	7B + 1.5B	57.6 (1.3↑) / 3.5
PLLaVA + Selector	34B + 1.5B	62.3 (1.4↑) / 3.6
LLaVA-NeXT-Video + Selector	7B + 1.5B	55.1 (1.6↑) / 3.4
LLaVA-NeXT-Video + Selector	34B + 1.5B	60.2 (1.4↑) / 3.5

Table 1. Comparison of open-ended question answering evaluation on ActivityNet QA. Results with the "**+ Selector**" are ours.

Model	Model Size	NExT-QA
SlowFast-LLaVA [56]	7B	64.2
SlowFast-LLaVA [56]	34B	72.0
Tarsier [45]	7B	71.6
Tarsier [45]	34B	79.2
LLaVA-NeXT-Video [62]	7B	62.4
LLaVA-NeXT-Video [62]	34B	68.1
Idefics2 [15]	8B	68.0
Qwen2-VL [46]	7B	77.6
LLaVA-NeXT-Video + Selector	7B + 1.5B	63.4 (1.0↑)
LLaVA-NeXT-Video + Selector	34B + 1.5B	69.3 (1.2↑)
Idefics2 + Selector	8B + 1.5B	69.1 (1.1↑)
Qwen2-VL + Selector	7B + 1.5B	78.4 (0.8↑)

Table 2. Comparison of multi-choice question answering evaluation on NExT-QA. Results with the "**+ Selector**" are ours.

Table 1, Table 2, Table 3 and Table 4 present the comparison on ActivityNet-QA [58], NExT-QA [54] and EgoSchema [32], VideoMME [9] respectively. Following existing practice [31], performance on ActivityNet-QA is measured as "accuracy/correctness" metrics, and both metrics are evaluated using GPT-3.5, with higher values indicating better performance. Performances on NExT-QA and EgoSchema are the accuracy of of multi-choice questions where each question has 5 options. We prefill the word "Option" as the initial token to generate, and then use the next generated token as the prediction result.

Model	Model Size	EgoSchema
SlowFast-LLaVA [56]	7B	47.2
SlowFast-LLaVA [56]	34B	55.8
Tarsier [45]	7B	56
Tarsier [45]	34B	68.6
LLaVA-NeXT-Video [62]	7B	45.8
LLaVA-NeXT-Video [62]	34B	48.6
Idefics2 [15]	8B	56.6
Qwen2-VL [46]	7B	64.6
LLaVA-NeXT-Video + Selector	7B + 1.5B	47.2 (1.3↑)
LLaVA-NeXT-Video + Selector	34B + 1.5B	50.6 (2.0↑)
Idefics2 + Selector	8B + 1.5B	57.9 (1.3↑)
Qwen2-VL + Selector	7B + 1.5B	65.9 (1.1↑)

Table 3. Comparison of multi-choice question answering evaluation on EgoSchema. Results with the "**+ Selector**" are ours.

Model	short	medium	long	average
Qwen2-VL (baseline)	69.1	53.0	51.6	58.1
Qwen2-VL + our selector	69.6	54.1	51.9	58.7 (0.6↑)

Table 4. Performances on VideoMME. **"+ Selector"** are ours.

4.3. Ablation Studies

We conduct ablation studies on several components of the frame selection framework. First, we analyze the performance of the following frame selection methods in the video QA tasks:

- **Uniform sampling**: the default uniform sampling.
- **Pseudo labels from CLIP similarity**: define the importance score of a frame as the image-text similarity between the frame and the text question, computed using a CLIP model. Then sample frames using Algorithm 1.
- **Pseudo labels from SeViLA**: define the importance score of a frame as in SeViLA [57] to sample frames.
- **Spatial pseudo labels**: define the importance score using the spatial pseudo labels only.
- **Spatial & temporal pseudo**: define the importance score as the average of the spatial and temporal pseudo labels.
- **Trained selector**: define the importance score of a frame using the output of our trained frame selector.

Not using video-LLM due to worse pseudo-label quality. We refer to *Table 3* in [8] for justification. In the temporal reasoning tasks, although video-LLMs are trained with frames sampled from the same video, they fall behind image-trained M-LLMs. It implies image-trained M-LLMs could lead to better temporal pseudo-labels for our method. We tried to use LLaVA-NeXT-Video to generate pseudo-labels. It is more time-consuming but worse pseudo-labels quality. On NExT-QA, using frames labeled by Qwen2-VL and LLaVA-NeXT-Video are 63.9% and 62.8% respectively.

Table 5 presents the video QA performance of LLaVA-NeXT-Video 7B on two benchmarks, respectively. **Uniform** and **CLIP** sampling serve as simple baselines for frame selection, while other methods utilize M-LLM reasoning. Both **SeViLA** and our **Spatial** are single-frame-based selection methods, with the latter achieving superior performance due to enhanced reasoning during multimodal LLM inference. **Spatial & Temporal** further improves upon **Spatial**, demonstrating the importance of temporal reasoning in frame selection. However, the computational cost to generate such pseudo labels are extremely high as it needs to prompt an M-LLM densely. The light-weight selector's performance matches the performance of frame selection using pseudo labels, validating the effectiveness of the selector architecture.

Selection Method	ANet-QA	NExT-QA
Uniform sampling	53.5	62.4
Scores from CLIP similarity	53.7	62.2
Pseudo labels from SeViLA	54.0	63.2
Spatial pseudo labels	54.2	63.6
Spatial & temporal pseudo	55.5	63.9
Scores from trained selector	55.1	63.4

Table 5. Performance of LLaVA-NeXT-Video 7B on ActivityNet (ANet) and NExT QA with different frame selection methods

# frames	Acc@C	Acc@T	Acc@D	Acc	Speed (s)
4	67.2	61.2	73.9	66.4	0.56
8	68.7	62.5	76.9	68.1	0.92
16	69.1	63.6	76.8	68.7	1.71
32	69.5	64.3	78.4	69.3	3.40
$128 \to 4$	68.5	64.5	75.7	68.5	0.76
$128 \to 8$	69.3	64.9	77.5	69.3	1.12
$128 \to 16$	69.4	64.8	78.5	69.5	1.91
$128 \to 32$	69.2	65.6	78.7	69.6	3.50

Table 6. Performance and inference speed of LLaVA-NeXT-Video 34B on NExT-QA with different number of input frames. First 4 rows: uniform sampling, last 4 rows: sampling using the selector.

We further show that the video QA system can use fewer frames to reason videos with the help of our frame selector. Table 6 shows the performance of LLaVA-NeXT-Video on NExT-QA taking different number of frames as input and the inference speed of LLaVA-NeXT-Video with different number of input frames. The inference speed was measured with float16 precision using a batch size of 1 on a single A100 GPU, employing the Hugging Face implementation. When taking the same number of frames as input, our framework incurs additional inference costs to select frames using the M-LLM selector. However, this increase in inference time is not significant thanks to the efficient design of the frame selector. Moreover, the frames selected by the M-LLM selector are more useful for answering the question. Therefore, **the model using a selector with n-frame input can achieve similar video QA performance as the model without a selector using $2n$-frame input**. For example, the configuration $128 \to 4$ outperforms 8-frame uniform sampling with a faster inference speed.

# tokens / frame	ActivityNet-QA	NExT-QA	EgoSchema
no selector	53.5	62.4	45.8
1	53.2	62.7	46.6
9	55.1	63.4	47.2
25	55.3	63.6	47.3

Table 7. Performance of **LLaVA-NeXT-Video 7B** with different number of tokens per frame used in the selector.

Backbone size	ActivityNet-QA	NExT-QA	EgoSchema
no selector	53.5	62.4	45.8
0.5 B	53.8	62.8	46.4
1.5 B	55.1	63.4	47.2
7 B	55.5	64.0	47.9

Table 8. Performance of LLaVA-NeXT-Video 7B with different size of the selector's base LLM.

model	# frames	Uniform	Selector
LLaVA-Next Video 34B	4	45.3	49.5
	8	46.9	49.9
	16	48.1	49.8
	32	49.7	50.0
Qwen2-VL 7B	4	48.0	55.0
	8	50.9	56.0
	16	53.6	56.5
	32	53.3	57.0

Table 9. Performance of LLaVA-NeXT-Video 34B Qwen2-VL 7B on LongVideoBench with different input frames.

As discussed in Section 3.2, one advantage of our framework is that the frame selector features a lightweight design. We examines two key hyperparameters that influence the computational efficiency: the number of tokens to represent a frame and the size of the base LLM. By default, we use Qwen2.5 1.5b as the LLM backbone and use 9 tokens for video frame representation. Table 7 and Tabel 8 present the ablation studies examining these two factors. "no selector" indicates sample the video frames uniformly.

We further demonstrate the effectiveness of our frame selector for long video question answering (QA). LongVideoBench [52] is a recently introduced benchmark for long-context video-language understanding, with an average video duration of 473 seconds. In Table 9, we report the performance of LLaVA-Next-Video 34B and Qwen2-VL 7B with different numbers of input frames, using uniform sampling and sampling with our frame selector. For both M-LLMs, the performance with n input frames sampled using the selector surpasses that of $2n$ input frames with uniform sampling, demonstrating the effectiveness of the frame selector.

4.4. Visualization of selected frames

Qualitative Results: Figure 4 presents two examples of selected frames from the video conditioned the question. Each question involves two events and thus requires temporal reasoning. Estimating the importance of a single frame is challenging without reference to prior or subsequent frames. The frame selector effectively identifies frames containing the answers to the questions. More results in Appendix Section C .

Evaluate on Video Grounding Benchmarks: We compare the moment retrieval performance with SeViLa on QVHighlights. Ours achieves 43.9% R1@5 and 32.3% R1@7 while SeViLa achieves 54.5% R1@5 and 36.5% R1@7. Although our frame selector has lower performance than SeViLa, it is not specifically trained on QVHighlights. In other words, the comparable performance proves the overall correctness of the frame selection.

Figure 4. Visualization of the frame selection results.

5. Conclusion

In this paper, we propose a lightweight M-LLM-based frame selector to improve both performances and efficiency in video QA. This selector is question-aware and takes dense video frames and the question as input, selecting the most relevant frames. We can then use any multi-image or video M-LLMs with the selected frames to complete the question-answering. To train the frame selector, we introduce spatial and temporal pseudo-labeling due to the limited public annotations for video frame-level importance. Our experiments on two medium-length video QA benchmarks (ActivityNet QA and NExt-QA) and two long-video QA benchmarks (EgoSchema and LongVideoBench, VideoMME) demonstrate the effectiveness of our proposed method.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv:2303.08774*, 2023. 2

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

[3] Anthropic. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024. Accessed: 2024-09-18. 1, 2

[4] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024. 1, 6

[5] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022. 2

[6] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 2

[7] Ran Cui, Tianwen Qian, Pai Peng, Elena Daskalaki, Jingjing Chen, Xiaowei Guo, Huyang Sun, and Yu-Gang Jiang. Video moment retrieval from text queries via single frame annotation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1033–1043, 2022. 2

[8] Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37:89098–89124, 2024. 7

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

[10] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14773–14783, 2023. 2

[11] Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. *arXiv preprint arXiv:2411.14794*, 2024. 3

[12] Wei Han, Hui Chen, Min-Yen Kan, and Soujanya Poria. Self-adaptive sampling for accurate video question answering on image text models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2522–2534, 2024. 3

[13] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024. 6

[14] Sungdong Kim, Jin-Hwa Kim, Jiyoung Lee, and Minjoon Seo. Semi-parametric video-grounded text generation. *arXiv preprint arXiv:2301.11507*, 2023. 2

[15] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024. 2, 6, 7

[16] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

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

[18] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 3

[19] 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. MVBench: A comprehensive multi-modal video understanding benchmark. *arXiv:2311.17005*, 2023. 2

[20] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 6

[21] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2025. 2, 3, 6

[22] Jianxin Liang, Xiaojun Meng, Yueqian Wang, Chang Liu, Qun Liu, and Dongyan Zhao. End-to-end video question answering with frame scoring mechanisms and adaptive sampling. *arXiv preprint arXiv:2407.15047*, 2024. 3

[23] 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. 2, 6

[24] Haogeng Liu, Qihang Fan, Tingkai Liu, Linjie Yang, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. Video-teller: Enhancing cross-modal generation with fusion and decoupling. *arXiv preprint arXiv:2310.04991*, 2023. 2

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2

[26] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, ocr, and world knowledge, 2024. 1, 2

[27] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV*, pages 319–335. Springer, 2022. 2

[28] Yuan Liu, Zhongyin Zhao, Ziyuan Zhuang, Le Tian, Xiao Zhou, and Jie Zhou. Points: Improving your vision-language model with affordable strategies. *arXiv preprint arXiv:2409.04828*, 2024. 2

[29] Haoyu Lu, Mingyu Ding, Nanyi Fei, Yuqi Huo, and Zhiwu Lu. LGDN: Language-guided denoising network for video-language modeling. In *Advances in Neural Information Processing Systems*, 2022. 2

[30] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023. 2

[31] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024. 2, 6

[32] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023. 6

[33] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. MM1: Methods, analysis & insights from multimodal llm pre-training. *arXiv:2403.09611*, 2024. 2

[34] TB OpenAI. Chatgpt: Optimizing language models for dialogue. *OpenAI*, 2022. 1

[35] Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryoo, Donghyun Kim, and Michael S Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*, 2024. 2

[36] Tianwen Qian, Ran Cui, Jingjing Chen, Pai Peng, Xiaowei Guo, and Yu-Gang Jiang. Locate before answering: Answer guided question localization for video question answering. In *IEEE transactions on multimedia*, 2022. 2

[37] Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. Understanding long videos in one multimodal language model pass. *arXiv preprint arXiv:2403.16998*, 2024. 3, 4, 12

[38] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024. 6

[39] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. MovieChat: From dense token to sparse memory for long video understanding. *arXiv:2307.16449*, 2023. 2

[40] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13581–13591, 2024. 3

[41] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023. 1, 2

[42] Qwen Team. Qwen2.5: A party of foundation models, 2024. 6

[43] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 2

[44] 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. 1, 2

[45] Jiawei Wang, Liping Yuan, and Yuchen Zhang. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 6, 7

[46] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 6, 7

[47] Xijun Wang, Junbang Liang, Chun-Kai Wang, Kenan Deng, Yu Lou, Ming C Lin, and Shan Yang. Vila: Efficient video-language alignment for video question answering. In *European Conference on Computer Vision*, pages 186–204. Springer, 2024. 3

[48] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent, 2024. 3

[49] Zixu Wang, Yujie Zhong, Yishu Miao, Lin Ma, and Lucia Specia. Contrastive video-language learning with fine-grained frame sampling. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 694–705. Association for Computational Linguistics, 2022. 2

[50] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv:2405.19209*, 2024. 3

[51] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language model. In *European Conference on Computer Vision*, pages 408–424. Springer, 2025. 2

[52] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer, 2025. 2, 8

[53] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024. 6

[54] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021. 6

[55] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. PLLaVA: Parameter-free llava extension from images to videos for video dense captioning. *arXiv:2404.16994*, 2024. 1, 2, 3, 6

[56] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 6, 7

[57] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 4, 7, 12

[58] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9127–9134, 2019. 6

[59] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 1

[60] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. 6

[61] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 1, 3

[62] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 6, 7

[63] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. 6