VideoLLM-online: Online Video Large Language Model for Streaming Video

Joya Chen[1] Zhaoyang Lv[2] Shiwei Wu[1] Kevin Qinghong Lin[1] Chenan Song[1]
Difei Gao[1] Jia-Wei Liu[1] Ziteng Gao[1] Dongxing Mao[1] Mike Zheng Shou[1]✉
[1]Show Lab, National University of Singapore [2]Reality Labs Research, Meta

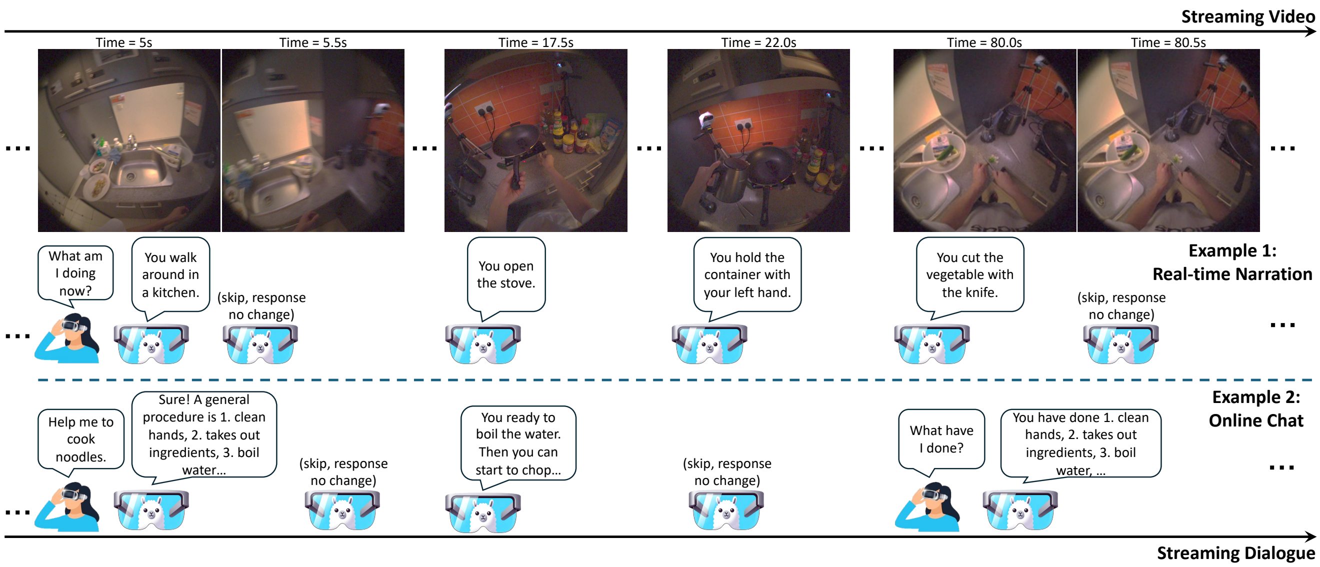

Figure 1. Zero-shot examples of our VideoLLM-online applied to an egocentric video stream from Ego-Exo4D dataset [27]. Our model is designed for temporally aligned, long-context, real-time dialogue in continuous video streams, shedding light on the future always-on, contextual AI assistants (*e.g.*, smart AR glasses). Model responses are appropriately simplified for better visualization.

Abstract

Recent Large Language Models (LLMs) have been enhanced with vision capabilities, enabling them to comprehend images, videos, and interleaved vision-language content. However, the learning methods of these large multimodal models (LMMs) typically treat videos as predetermined clips, rendering them less effective and efficient at handling streaming video inputs. In this paper, we propose a novel Learning-In-Video-Stream (LIVE) framework, which enables temporally aligned, long-context, and real-time dialogue within a continuous video stream. Our LIVE framework comprises comprehensive approaches to achieve video streaming dialogue, encompassing: (1) a training objective designed to perform language modeling for continuous streaming inputs, (2) a data generation scheme that converts offline temporal annotations into a streaming dialogue format, and (3) an optimized inference pipeline to speed up interactive chat in real-world video streams. With our LIVE framework, we develop a simplified model called VideoLLM-online and demonstrate its significant advantages in processing streaming videos. For instance, our VideoLLM-online-7B model can operate at over 10 FPS on an A100 GPU for a 5-minute video clip from Ego4D narration. Moreover, VideoLLM-online also showcases state-of-the-art performance on public offline video benchmarks, such as recognition, captioning, and forecasting. The code, model, data, and demo have been made available at showlab.github.io/videollm-online.

✉Corresponding Author.

1. Introduction

Building the future of always-on contextual AI agent that can promptly answer any human question, digitizing inputs as episodic memories and forecasting future plans given any query in online continuous setting represents a "holy grail" mission in AI research. Powered by advancements in large language models (LLMs) [8, 34, 60, 61, 63, 72], recent large multimodal models (LMMs) have unveiled impressive capabilities such as vision-language dialogue [6, 13, 40, 41, 48, 49, 93], spatial understanding [6, 37, 51, 64, 83, 87], processing diverse modalities [19, 28, 55, 77, 85]. Seminal exemplars, like OpenAI's GPT-4V [62], are progressively evolving into highly versatile human AI assistants.

It is time to envision an always-on, contextual, J.A.R.V.I.S-like multimodal assistant that supports free-form user-assistant dialogue within the video stream, which we term "video streaming dialogue". Unlike existing LMMs for video understanding (*i.e*., VideoLLMs) [42, 43, 52, 56, 69, 81, 86] that work offline with manually selected short-video clips, an online assistant should continuously receive video frames with visual content that is constantly refreshed. This paradigm shift presents new challenges. First, the user query may come with **temporally aligned** requirements (*e.g*., "alert me when it's time to flip the steak"), thus the VideoLLM should scan every incoming frame to avoid event missing, instead of only yielding video-level responses. Second, to answer questions regarding summarization and planning, the VideoLLM must retain the **long-context** historical vision and language, which poses huge risk of exceeding maximum context window of LLMs, as well as introduces considerable burden to causal decoding speed and GPU memory. Third, the VideoLLM should generate the answer in **real-time**, keeping pace with the video stream for always-on scenario. These abilities, however, are even partially overlooked by the most advanced AI assistants [3, 62].

One possible path towards such an online VideoLLM, inspired by current interleaved vision-language models [3, 6, 40, 62, 82], is to employ a multi-turn dialogue format to achieve per-frame chatting within a video stream. This can be accomplished by facilitating very frequent user interactions, utilizing the visual frame as query at each timestamp to obtain the answer. We follow this to perform prompt engineering for GPT-4V [44, 62], but the results are disappointing: GPT-4V tends to output lengthy content at every frame, leading to significant delays, making it impractical for real-time streaming video. We also explore training baseline models for per-frame chatting. Unfortunately, this approach evidently diminishes the language modeling capability, likely due to harmful language modeling on an excessive number of redundant frames.

We propose Learning-In-Video-strEam (LIVE), a comprehensive framework that encompasses learning, data, and inference methods to develop an online video assistant. Unlike per-frame dialogue approach, LIVE introduces a novel training objective termed *Streaming EOS (End-Of-Sequence) prediction* that enables the model to learn when to response or remain silent in a video stream. This objective differs from next-token prediction since EOS tokens here will not appear in the input/output sequence. However, it can work well with the autoregressive loss to train an online VideoLLM. This design reduces unnecessary context, helping the model to manage much longer streaming videos. Nevertheless, the training still requires data from user queries and assistant responses within video streams, which is scarce in popular video datasets usually used for training offline video models. To address this issue, LIVE presents a streaming dialogue generation scheme that converts offline annotations into online dialogues to support free-form chatting. To enhance inference efficiency, LIVE leverages continuous key-value caching for streaming assistance, and parallelizes the fast visual encoding and slow language decoding to prevent bottlenecks, thus moving towards real-time application.

With LIVE framework, we build a simple VideoLLM-online model upon CLIP [65] vision encoder and Llama-2 [73] language model. To evaluate the performance of video streaming dialogue, we utilize the language perplexity metric and design two new metrics to comprehensively assess the model's capabilities in language modeling, temporal responsiveness, and overall fluency. Experiments with real-time Ego4D narration [26] demonstrate that our method shows advantages in all that metrics, with higher speed and lower memory cost. Furthermore, our model achieves state-of-the-art results on numerous offline benchmarks, such as short- and long-term activity recognition and forecasting on the COIN and Ego4D LTA benchmarks [26, 71]. In addition, our model has good speed/memory efficiency, which allows continuous video narration on Ego4D video clips with memory cost less than 20 GB and average speed higher than 10 FPS on a single A100 GPU, paving the way for future real-world usage.

2. Related Work

Visual Dialogue. Before transformers [74] become mainstream in vision, visual dialogue methods [1, 16] tend to employ a visually enhanced encoder, with a discriminative head to select candidate answers or a recurrent architecture to generate multi-turn responses. For the encoder, a variety of attention mechanism-based approaches [22, 59, 66] have been proposed to learn the interactions between the image, the answers, and the dialogue history. There have also been explorations of encoder-only BERT [18] models for visual dialogue [57, 76]. However, most of them rely solely on a single image/video at the beginning of the conversation, followed by multi-turn pure language dialogue, which makes them less flexible than the current interleaved vision-language dialogue systems.

Large Multimodal Models. The advent of large language models (LLMs) [8, 60, 63] has inspired a series of LLM multimodal variants, *i.e*. large multimodal models (LMMs). Early LMMs [2, 13, 41, 49, 93] achieve image dialogue by projecting the image encoding (*e.g*., from CLIP [65]) to align with LLM embedding space. Then, lots of efforts [3, 6, 40, 62, 82] explore more free-form interleaved vision-text chatting, spatial understanding [6, 11, 12, 37, 51, 83, 87], video comprehension [42, 43, 52, 56, 69, 81, 86], etc. However, when it comes to online scenario, there is less exploration on how LLMs can fulfill the temporal alignment, long-context, and real-time requirements for streaming video inputs. Our research bridges this gap, offering comprehensive solutions across model training, data, and inference to study the problem.

Online Video Understanding. Typical video understanding benchmarks, such as action recognition [9], temporal action localization [29], video question answering [38], and video dialogue [1], typically allow models to access entire video frames to make predictions, a setting referred to as ”offline”. However, such setting does not align well with many real-time demands (*e.g.*, autonomous driving, AR glasses). Instead, there is a growing focus on “online” video understanding problems like online action detection [24] and anticipation [35], which aim to identify the current/future action at each timestamp without seeing the future. Our study pioneers LMMs for online video understanding. Unlike previous online action detection [75, 89] or anticipation models [25, 88], which primarily address one task with a highly customized model, we aim to propose a general solution to achieve free-form dialogue along the online video stream, enabling a model to flexibly handle diverse tasks.

Efficient Token Decoding. Efficient token decoding for LLMs and LMMs is essential for applying them to real-time online services. To accelerate that, a diversity of strategies has been proposed, such as parallelism on batch dimension [67] or cached key-value sequence [15], computation/memory management optimization [23, 30, 36, 67, 84], and even some lossy approaches [50, 78]. Our focused streaming video scenario has less concerns in large batch processing, but expects faster decoding to avoid excessive frame skipping. We have considered this in training objectives, and further propose some inference schemes to accelerate the decoding efficiency.

3. Method

In this section, we present Learning-In-Video-strEam (LIVE) framework that enables LMMs to provide temporally aligned response, handle long-context streaming video, and run efficiently towards real-time usage. We will start from the problem definition of “video streaming dialogue”, analyze its challenges, and introduce our approach to solve the problem.

3.1. Video Streaming Dialogue

Problem Formulation. Though huge successes have been witnessed in large multimodal models (LMMs), the assistance scenario like a smart AR glass helping the user with cooking, is still far from the capabilities of current LMMs, even for the most advanced version, GPT-4V [62]. For example, despite carefully prompting GPT-4V similarly to MMVid [44]—to perform per-frame dialogue for handling streaming video inputs—the redundancy in responses between frames, a limited context window of 10∼50 frames, and slow speed, collectively render the current GPT-4V unsuitable for online video understanding, as our prompting analysis demonstrates (see supplementary material).

To bridge the gap, we define a problem termed “video streaming dialogue”. Given the context sequence before time $t = t_1$, denoted as $\texttt{[Ctx}^{t<t_1}\texttt{]}$, which may encompass previous vision-language content (*e.g.*, historical user queries, video frames, assistant responses), and an ongoing continuous video stream from t_1 to t_2, denoted as $\texttt{[Frame}^{t_1\le t\le t_2}\texttt{]}$, our goal is (1) to determine whether the current time t_2 is suitable for language modeling; (2) to carry out language modeling

$$\max P(\texttt{[Txt}^{t_2}_{i+1}\texttt{]}\,|\,\texttt{[Ctx}^{<t_1}\texttt{]},\texttt{[F}^{t_1\le t\le t_2}\texttt{]},\texttt{[Txt}^{t_2}_{\le i}\texttt{]}) \tag{1}$$

if t_2 is determined, where $\texttt{[Txt}^{t}_{i}\texttt{]}$ denotes the ideal language token in the i-th position at timestamp t. $\texttt{[F]}$ is the abbreviation of $\texttt{[Frame]}$.

In the following, we analyze if existing techniques have been enough to solve the problem.

Interleaved/Per-frame Dialogue are Suboptimal. First, we investigate whether the popular approach of interleaved vision-language chatting can address this problem. Related to our formulation above, such a method learns language modeling after given frames between timestamps t_1 and t_2. However, if this approach is adopted during inference, it necessitates the manual selection of timestamps t_1 and t_2, which does not align with the concept of video streaming dialogue. Current VideoLLMs [42, 43, 52, 56, 69, 81, 86] represent a simplified version of this approach, engaging in single- or multi-turn pure language dialogue following video clip inputs.

If we consider a more free-form format, multi-turn interleaved vision-language chatting [3, 6, 40, 62, 82], we can get a per-frame chatting solution that might be hopeful to solve video streaming dialogue, which performs per-frame language modeling

$$\max P(\texttt{[Txt}^{t}_{i+1}\texttt{]}\,|\,\texttt{[Ctx}^{<t_1}\texttt{]},\texttt{[Frame}^{t}\texttt{]},\texttt{[Txt}^{t}_{\le i}\texttt{]}) \tag{2}$$

for every frame from timestamp $t_1 \le t \le t_2$. Since t_2 is necessary to output answer, so we can simply learn short text (*e.g.*, “this is not the time to answer”) between $t_1 \le t < t_2$. However, this approach imposes a significant burden on processing speed. For every frame, performing the slow, recurrent, and lengthy next-token prediction with a billion-scale language model makes it extremely hard to achieve real-time video streaming dialogue. Then, this will lead to unavoidable frame skipping, which is unexpected and detrimental for temporal alignment. Furthermore, it taxes the limited context window of the language model, presenting challenges for modeling long contexts and managing GPU memory efficiently.

Streaming EOS Prediction. To solve the problem mentioned above, we first consider a more efficient per-frame chatting method: simply assigning the End-of-Sequence (EOS) token as the content for chatting between $t_1 \le t < t_2$. However, this approach remains suboptimal. The dialogue prompt template (*e.g.*, `[INST]`, `[/INST]`, space tokens in Llama [72, 73]) still consumes a considerable number of tokens per frame, which is less favorable due to the numerous frames in streaming video. Furthermore, the excessive number of EOS tokens in the sequence can significantly increase the language model's perplexity, as we have observed in our experiments.

Instead, we propose a novel training objective named "streaming EOS prediction" to address this issue. We still assume t_2 is essential for decoding language; thus, we normally learn language modeling here:

$$\max P(\texttt{[Txt}^{t_2}_{i+1}\texttt{]}|\texttt{[Ctx}^{<t_2}\texttt{]}, \texttt{[Frame}^{t_2}\texttt{]}, \texttt{[Txt}^{t_2}_{\le i}\texttt{]}). \tag{3}$$

However, for timestamps $t_1 \le t < t_2$, which are redundant for producing answers, we directly learn the model to predict EOS token on the frame tokens, *i.e.*

$$\max P(\text{EOS}|\texttt{[Ctx}^{<t}\texttt{]}, \texttt{[Frame}^{t}\texttt{]}), \text{where } t_1 \le t < t_2. \tag{4}$$

In this way, we "skip" a dialogue turn and learn to determine when it is appropriate to decode language for streaming inputs. During inference, if EOS is predicted on a frame, then we can directly ask the next frame to input. Meanwhile, the EOS token is not appended to the context to prevent it from affecting the language modeling. Therefore, this task is not about next-token prediction; however, it can work with autoregressive loss to train a video streaming dialogue model. In the following, we first introduce how can we get the data of streaming video and timestamped language annotations, then present the details about the model and the training procedure.

We also note that the EOS token mentioned here is not limited to the real EOS token used in language models (*e.g.*, `</s>` in Llama). It is permissible to use any token or to introduce a new token, provided that it is specified in the system prompt. We use this term solely for simplicity.

3.2. Data

Online Annotations to Video Streaming Dialogue. Some video datasets, such as Ego4D narrations [26], are inherently collected in a streaming manner, with annotators providing real-time narrations as they watch a 5-minute long video clip. However, prior research [45, 90] has predominantly focused on learning from short, discrete clips (*e.g.* only 32 frames), rather than in a continuous streaming context. For this dataset, we followed the same instructions provided to human annotators [26] as our model training prompt. This prompt instructs the model to simulate human annotators to streamingly generate narrations in a 5-minute video (about 600 frames in 2 FPS). For demo purposes (not for experiments), we also utilize Llama-2-13B-Chat [73] to rephrase the narration text, correcting grammatical errors and typos, and converting it into a more understandable version (*e.g.*, changing "C does..." to "You do...").

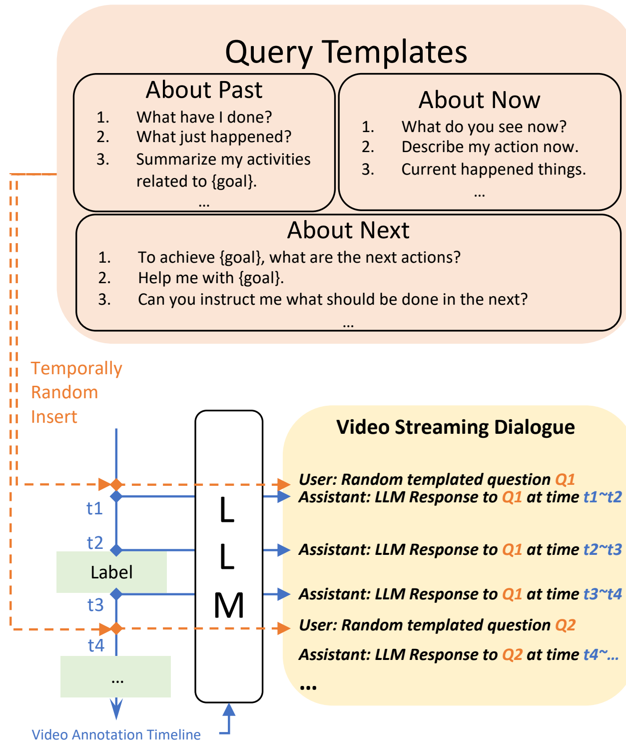

Figure 2. **The streaming dialogue data generation method in our LIVE framework**. We randomly insert templated questions into the video timeline and "expose" the ground-truth video annotations (along with their timestamps) to LLMs, prompting them to answer the queries within a period of time.

Offline Annotations to Video Streaming Dialogue. Despite the Ego4D narration data being collected in a streaming manner, most prevalent video datasets [14, 26, 54, 71] are used to train offline models and only feature temporal segment annotations paired with basic language descriptions (*e.g.*, activities, narrations). To bridge this gap, we propose a method for synthesizing dialogue data from these sources. As shown in Figure 2, our key idea is use LLM to generate user-assistant dialogues based on video annotations, involving the following steps:

- First, we prepare a question template library containing various queries about the past, present, and future tenses of the video, totaling N queries. We randomly sample one question from the library, denoted as Q_i.
- Then, we obtain the video annotation timeline from the offline dataset. This usually includes timestamped language descriptions, which we organize into a language prompt, e.g., "time $t_a \sim t_b$: boiling the water; time $t_c \sim t_d$: cutting the vegetables.", denoted as A. We consider all the state change critical timestamps as the ideal response times. For this example, t_a, t_b, t_c, and t_d are all considered response times.
- Third, we prompt the large language model to generate responses at every critical timestamp, *e.g.*, t_a, t_b, t_c, t_d, according to Q_i and A. We can repeat this procedure for each Q_i until all queries have been processed. The responses are saved for loading during training.
- Finally, during training, we (1) randomly sample a query and load its responses at critical timestamps, (2) randomly insert a query into a video timestamp t_r, (3) discard the responses that occur before t_r, and add a response at t_r. Here different queries can be inserted into one video, which only requires discarding the responses of the previous query after the new query insertion timestamp.

In this way, we can generate temporally varied and free-form dialogue data within a video stream. We have prepared 50 questions each for past, current, and future events, totaling $N = 150$ queries. We use Llama-2-13B-Chat to generate the responses and insert a maximum of 3 queries per training sample. The offline datasets we used are COIN [71] and Ego4D GoalStep [70] (for demo usage), which belong to the categories of egocentric and instructional video datasets, aligning with our aim to develop online video assistants. Here, we do not consider online action detection benchmarks (e.g., THUMOS14 [33], TVSeries [24]) because they are closed-set online classification benchmarks, and their labels are too brief, which may cause language models to generate hallucinatory responses. Please refer to the supplementary material for the generated dialogue example.

3.3. Model Training

Model Architecture. We illustrate the model architecture in Figure 3. Similar to LLaVA [48, 49], it comprises three key components: an image encoder, an MLP projector, and a language model. For the image encoder, we utilize the CLIP ViT-L [20, 65] encoder (pretrained on DataComp-1B [21]) to extract video frame embeddings at 2 FPS. Each video frame embedding has a shape of $(1 + h_p \times w_p) \times c$, where $(1 + h_p \times w_p)$ denotes the CLS token and average pooled spatial tokens.[2] The extracted frame embeddings from the image encoder are then fed into MLP projector to frame tokens, as in LLaVA-1.5 [48]. Then frame tokens are interleaved with language tokens as input to an LLM, Llama-2-7B-Chat [73]. Finally, we incorporate LoRA [31] in every linear layer of the LLM for efficient tuning.

Training Loss. As described in Section 3.1, our learning objective is twofold. The first part focuses on autoregressive language modeling, aiming to maximize the joint probability of input text sequences. The second training objective involves streaming EOS prediction, which requires the model to remain silent when it is unnecessary to output responses. With these two training objectives, we have language modeling (LM) loss and streaming loss terms to minimize, both employing cross-entropy loss:

$$L = \frac{1}{N}\sum_{j=1}^{N}(\underbrace{-\log l_{j+1}P_j^{\texttt{[Txt}_{j+1}\texttt{]}}}_{LMLoss} - \underbrace{w\log f_j P_j^{\texttt{[EOS]}}}_{StreamingLoss}), \quad (5)$$

where l_j and f_j denote condition indicators. l_j is 1 if the j-th token is a language response token, and 0 otherwise. f_j is 1 if (1) the j-th token is the *last* token of a frame[3], and (2) $l_{j+1} = 0$. In essence, the streaming EOS loss is applied to frames before responding. $P_j^{\texttt{[Txt}_{j+1}\texttt{]}}$ denotes the probability on $j+1$-th text token, output from the language model head of the j-th token, and $P_j^{\texttt{[EOS]}}$ represents that probability for the EOS token. w is a balance term, set to $w = 1$ by default. As shown in Figuer 3, we visualize the ranges of language loss and streaming loss in an input sequence when we only use 1 token for each frame.

3.4. Inference

Probability Correction. The prevalence of EOS token will bias the model towards EOS token prediction. To address this, we introduce a threshold θ to correct the output probability on frame tokens: EOS will not be considered as the next token if $P_j^{\texttt{[EOS]}} < \theta$. In practical usage, we find that setting θ to 0.5 yields much better results than no threshold here.

Continuous Key-Value Cache. As shown in Figure 4, During inference, the video is input as a frame-by-frame stream, with a default FPS 2. Our model takes the current frame as input and generates tokens on-the-fly. In the whole process, we use the key-value cache trick to accelerate token decoding, thus we do not need to manually append the generated

[2] The experiments described in this paper are conducted without extra spatial tokens (*i.e.*, $h_p = w_p = 0$), which is the most efficient setup and can handle half-hour videos within a 4096 context window. Our released models for demo usage include $1 + h_p \times w_p = 1 + 3 \times 3 = 10$ tokens, offering better detail in dialogue but supporting shorter maximum video lengths. Despite more tokens used per frame, all models can still run at over 10 FPS for 5-minute Ego4D narration streams.

[3] When a frame has multiple patch tokens, loss is only on the last one.

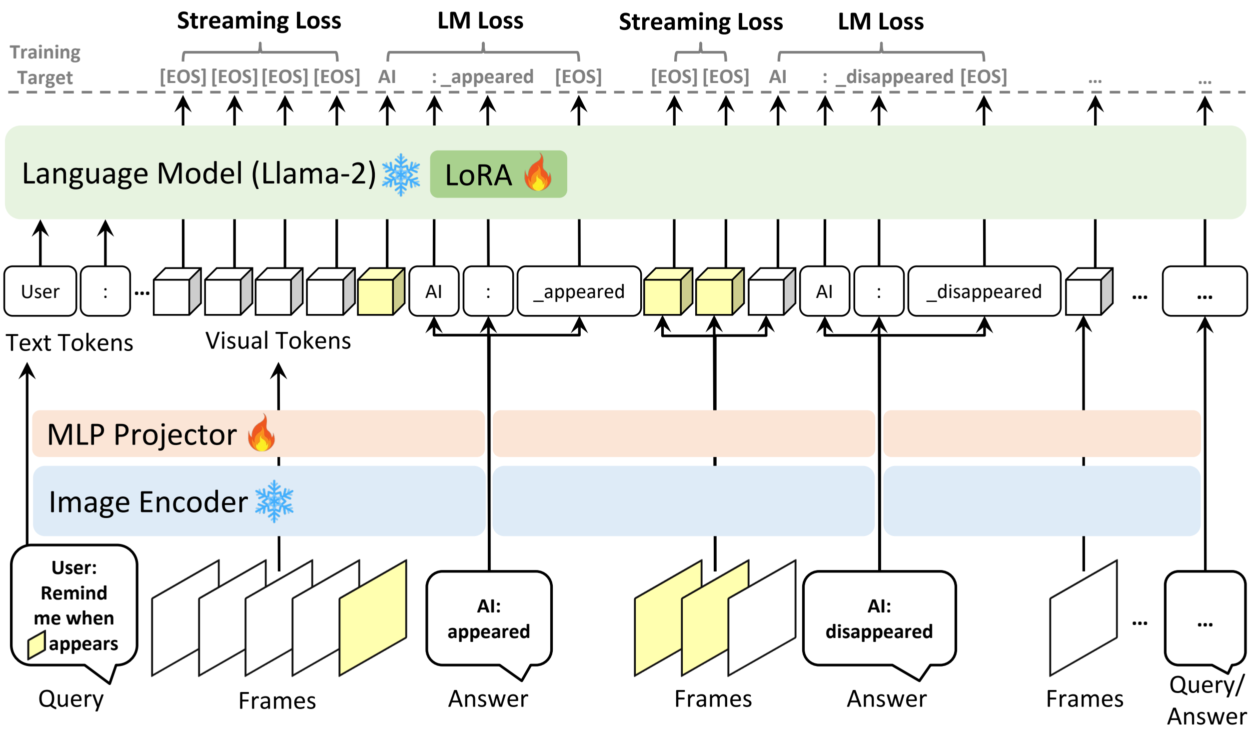

Figure 3. **The training method in our LIVE framework**. We organize the user-assistant dialogue data and video frames in temporal order as the input sequence. To learn the model when to answer or keep silent in a video stream, we employ not only the standard language modeling (LM) loss but also introduce a streaming EOS prediction loss. This additional loss supervises the model when it is necessary to generate language, enabling it to produce temporally aligned responses and reduces the redundant dialogue history.

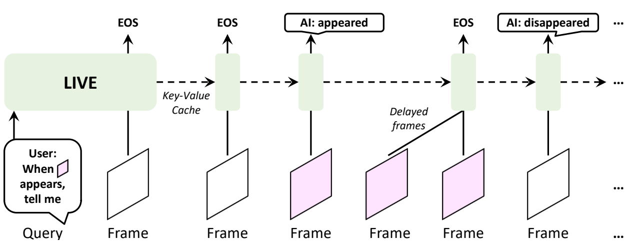

Figure 4. **Inference pipeline in our LIVE framework**. During inference, video frames serve as streaming inputs. Our model maintains a continuous key-value cache as the input progresses to speed up the inference. Furthermore, we parallelize the fast video frame encoder and the slower language model to avoid the bottleneck in the latter. Video frame tokens can be always encoded and buffered, no need to wait the language decoding.

tokens for the next frame. As our training encourages the model to keep silence, this continuous inference would be efficient, providing the possibility to pace with the video stream speed.

Parallelization of encoding and decoding. Our video frame encoder utilizes CLIP ViT-L (307M), which is significantly smaller than the 7B LLM. This size discrepancy leads to a speed mismatch, potentially resulting in frame skipping when the LLM decodes long sentences. To mitigate this issue, we parallelize the processes and establish a buffer for video frame tokens. Once the language model completes its decoding, it fetches all frame tokens from the buffer and inputs them in batch.

4. Experiments

4.1. Implementation Details

We implement VideoLLM-online-7B with our LIVE framework. It uses OpenCLIP-ViT-L [21, 65] as the video frame encoder, 2-layer MLP as multimodal connector, and Llama-2-7B-Chat [73] as the language model. We only use 1 token for each video frame, sampled by 2 FPS in the following experiments. We add LoRA [31] to all linear layers of Llama-2, with a rank of 128, scaling factor of 256. For video streaming dialogue experiments, we train 2 epochs for the model. For the downstreaming offline experiments, we train 5 epochs without pre-training for fair comparison with previous methods. By default, we set streaming loss weight $w = 1.0$ during training.

Method	Training Objective	Ego4D Narration Stream on Validation Set *LM-PPL*↓	*TimeDiff*↓	*Fluency*↑	#Training Token↓	Training Cost
No Training		498.5	6.50	0.1%	n/a	n/a
Interleaved Dialogue	Language Modeling	2.45	6.47	11.1%	**1694**	**12h**
Per-frame Dialogue for Streaming	Language Modeling (w/ EOS turns)	3.34	2.52	37.7 %	6737	22h
Streaming Dialogue (Ours)	Language Modeling + Streaming EOS	**2.43**	**2.32**	**42.6%**	**1694**	**12h**

(a) **Learning method for streaming dialogue**. Training with streaming dialogue method can achieve much better *TimeDiff* and *Fluency*, as well as maintain the language modeling quality. Meanwhile, the streaming dialogue can enjoy much more efficient training than per-frame dialogue for video streaming dialogue.

Streaming Loss	Ego4D Narration Stream Validation *LM-PPL*↓	*TimeDiff*↓	*Fluency*↑
Standard CE	**2.43**	**2.32**	**42.6%**
OHEM [68]	2.53	2.39	41.0%
Focal Loss [46]	2.59	2.44	39.4%

(b) **Streaming loss function**. Standard CE (cross-entropy) is enough for training streaming dialogue; there is no need to specifically to address the class imbalance on EOS token.

Weight τ	Ego4D Narration Stream Validation *LM-PPL*↓	*TimeDiff*↓	*Fluency*↑
$\tau = 0.5$	2.44	2.32	42.4%
$\tau = 1.0$	**2.43**	2.32	**42.6%**
$\tau = 2.0$	2.46	**2.31**	42.5%
$\tau = 3.0$	2.47	2.32	42.5%

(c) **Streaming loss weight**. Using slightly higher streaming loss weight ($\tau = 2.0$) achieves the best trade-off among various metrics.

Method	*Mem*↓	*FPS*↑
Interleaved	34.4G	1.5
Per-frame Streaming	24.9G	7.5
Streaming	**18.2G**	**13.5**

(d) **Generation memory/speed**. Streaming dialogue method has much better efficiency.

Table 1. **Ablation experiments on Ego4D Narration Stream**. We train VideoLLM-online on Ego4D [26] narration stream training set and evaluate on its validation set. The comparison is based on our designed metrics: the ratio of strictly correct prediction tokens (*Fluency*), language modeling perplexity (*LM-PPL*) and time difference (*TimeDiff*) metrics. #Training Token denotes the average token length during training. *TimeDiff* refers to difference in second. Default settings are highlighted.

Method	Not use HowTo100M	COIN Benchmark Top-1 Accuracy↑ Step	Task	Next	Proc.	Proc.+
ClipBERT [39]	✓	30.8	65.4	-	-	-
TimeSformer [7]	✗	46.5	85.3	34.0	17.0	40.1
Paprika [92]	✗	51.0	85.8	43.2	-	-
DistantSup [47]	✗	54.1	90.0	39.4	-	41.3
VideoTF [58]	✗	56.5	91.0	42.4	40.2	46.4
ProcedureVRL [91]	✗	56.9	90.8	46.8	-	-
VideoTaskGraph [5]	✗	57.2	90.5	40.2	-	-
VideoLLM-online	✓	**59.8**	**92.1**	**48.1**	**47.9**	**52.9**

(a) Results on COIN benchmarks (left to right): step recognition, task recognition, next forecasting, procedure forecasting, procedure forecasting with a goal.

Method	Not use EgoVLP	End-to-end?	Ego4D LTA ED@Z=20↓ Verb	Noun	Action
CLIP [17]	✓	✓	0.739	0.769	0.941
EgoT2 [79]	✓	✓	0.722	0.764	0.935
I-CVAE [53]	✓	✓	0.753	0.749	0.931
HierVL [4]	✓	✓	0.724	0.735	0.928
VideoLLM [10]	✗	✓	0.721	0.725	0.921
VideoLLM-online	✓	✓	**0.697**	**0.698**	**0.897**
Palm [32]	✗	✗	0.696	0.651	0.886
AntGPT [88]	✗	✗	0.650	0.650	0.877

(b) Results on Ego4D LTA benchmark, evaluated on public server. ED@Z=20 denotes editing distance for future 20 actions.

Table 2. Experiments on COIN [71] and Ego4D [26] benchmarks. VideoLLM-online is finetuned on their training set, and strictly evaluated on the test set by generated string comparison with the ground-truth text. It achieves best results among end-to-end models.

4.2. Evaluation Setting

Datasets. We use (1) assistance-related, instructional video dataset COIN [71] and (2) continuous, egocentric video dataset Ego4D [26] in various settings:

- **Ego4D Narration Stream**: We also leverage the dense Ego4D timestamp-narration to create a streaming set. The goal is to generate narrations timely like Ego4D human annotators [26]. We follow the division of the training, validation, and test set in EgoVLP [45].
- **COIN+Ego4D Stream**: To further evaluate the potential of model's performance to free-form dialogue, we construct a simple COIN+Ego4D Stream set, constructed from COIN annotations using our data generation methods, and the above Ego4D Narration Stream. The model should remind the user when an action starts, summarizes the action when it ends, as well as forecasts the next action. We use the same training/testing splits as COIN benchmarks. See supplementary material for the results.
- **COIN Benchmarks**: Following on previous studies [47, 58, 80, 91], we evaluate our model on six common benchmarks of the COIN dataset: step recognition, step forecasting, task summarization, procedure forecasting, procedure forecasting with a goal.
- **Ego4D long-term action anticipation (LTA) benchmark**: This benchmark requires to predict next $Z = 20$ actions (verbs and nouns) for the given video of previous 8 steps. We use the standard Ego4d v2 splits as in previous studies [32, 88].

Evaluation metrics. We use the following metrics to evaluate the model as an online video assistant:

- **Language modeling perplexity (LM-PPL).** This indicates the language modeling capability at a given timestamp. A lower LM-PPL is better for correct answering.
- **Time Difference (TimeDiff).** To evaluate the temporal

alignment capability of an online assistant, we calculate the discrepancy between the timestamp of its response and the expected timestamp for each response. We average TimeDiff each turn as the metric.

- **Fluency.** Individual *LM-PPL* or *TimeDiff* do not entirely evaluate both language and temporal effectiveness in a streaming dialogue. We introduce the Fluency metric, which evaluates the proportion of consecutive successful token prediction within a dialogue turn. As the token also including language tokens, Fluency can comprehensively reflect the aspects of LM-PPL and TimeDiff.

Baselines. To our best knowledge, we are the first one to tackle producing temporal aligned, free-form language answer with streaming video settings. To better understand the challenges, we build baseline models for video-text interleaved dialogue, per-frame dialogue, as we described in Section 3.1, with the same model architecture and training details to VideoLLM-online, differing only in their training objective and multi-turn formulation.

4.3. Ablation Study

Learning Method. Table 1a shows the ablation studies on learning methods in a streaming setting. Both vision-language interleaved and streaming methods exhibit low perplexity loss, indicating that our proposed objective does not hurt language modeling capability. However, learning with per-frame for streaming will produce significant higher *LM-PPL* than others, which might be attributed to the too more single EOS token in answering that affects the original language modeling.

When we turn to online metrics of *TimeDiff* and *Fluency*, streaming dialogue method yields much better results than others. In our observation, the first interleaved dialogue method always outputs language after every frame, and the second mutli-turn for streaming dialogue approach tends to answer EOS token after every frame, which decreases their performance for streaming video inputs. Furthermore, per-frame for streaming dialogue method will significanly slow down the training speed due to its lengthy prompts, while our method has no negative impact on the efficiency.

Streaming Loss. We continue to investigate the most suitable strategy to learn the streaming objective. As shown in Table 1b and Table 1c, we find a default setting works surprisingly well (CE loss, $\tau = 1.0$), which demonstrates there is no need to apply more advanced loss (*e.g*. Focal Loss [46]) to address the imbalance on EOS token.

Inference Efficiency. In Table 1d, we test the inference efficiency on Ego4D narration stream validation set (5 minute), and report the memory cost and average FPS on a single A100 GPU. The first interleaved dialogue method, which will output language after every video frame, has huge memory cost and slow generation speed. The second one, per-frame dialogue for streaming that formulates all in a multi-turn dialogue, show better efficiency than the first one since it can cost less tokens in redundant frames. However, this approach still lags significantly behind our streaming dialogue approach, which does not cost extra tokens in redundant frames thus maintain smaller key-value cache. We observe for most Ego4D videos, our model can run larger than 10 FPS, providing possibility for AI assistants working in real-time video stream.

4.4. Results

Offline language modeling. We show our model can perform well on traditional temporal summarization and forecasting problems. As shown in Table 2(a), our model achieves state-of-the-art performances in step/task summarization and next step/procedure forecasting benchmarks of COIN dataset [71]. Furthermore, we also obtain the best performance among end-to-end models evaluated on Ego4D LTA. We observe the results of Palm [32] and AntGPT [88] are better than us, but they used egocentric pre-trained visual feature [45], and integrates lots of complex cascading methods to improve the forecasting results. Our VideoLLM-online, however, directly outputs language as the results, which performs better than the similar end-to-end VideoLLM [10].

Visualization. See Figure 1, we visualize two representative examples, real-time narration and online dialogue. The most distinctive characteristics of our approach are: (1) the dialogue process goes along with the streaming video input, rather than chatting based on the full video. (2) The response will be "muted" when it is unnecessary, significantly improving the overall speed of video streaming dialogue.

5. Conclusion

We propose Learning-In-Video-strEam (LIVE), a novel framework empowering LLMs to handle streaming video, to produce temporal aligned answers, hold long-context video duration, and have high inference efficiency. We use LIVE to train a simple VideoLLM-online model, which not only achieves superior capability in online/offline vision-language tasks, but also enable fast inference for an online video streaming setting. We believe enabling such abilities will be an important step to move towards always-on online assistant. In future work, to make our VideoLLM-online be more general and improve its spatial capability in zero-shot prediction for downstream applications, we will explore suitable pre-training data source, and develop models that can employ more spatial tokens but without many trade-off on speed and memory cost.

Acknowledgment This work is sponsored by Project Aria Team, Meta. The datasets and processing were acquired and all models were trained at the National University of Singapore (NUS) by NUS authors.

References

[1] Huda AlAmri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K. Marks, Chiori Hori, Peter Anderson, Stefan Lee, and Devi Parikh. Audio visual scene-aware dialog. In *CVPR*, pages 7558–7567, 2019. 2, 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 2

[3] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2023. 2, 3

[4] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *CVPR*, pages 23066–23078, 2023. 7

[5] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. In *NeurIPS*, 2023. 7

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv:2308.12966*, 2023. 2, 3

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 7

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901, 2020. 1, 2

[9] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017. 3

[10] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, et al. Videollm: Modeling video sequence with large language models. *arXiv preprint arXiv:2305.13292*, 2023. 7, 8

[11] 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. 3

[12] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 3

[13] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C.H.Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv:2305.06500*, 2023. 2

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *Int. J. Comput. Vis.*, 130(1):33–55, 2022. 4

[15] Tri Dao, Daniel Haziza, Francisco Massa, and Grigory Sizov. Flash-decoding for long-context inference. https://pytorch.org/blog/flash-decoding/, 2023. 3

[16] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, pages 1080–1089, 2017. 2

[17] Srijan Das and Michael S. Ryoo. Video + clip baseline for ego4d long-term action anticipation. *arXiv:2207.00579*, 2022. 7

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 2

[19] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv:2309.11499*, 2023. 2

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5

[21] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah M. Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen

Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. *arXiv:2304.14108*, 2023. 5, 6

[22] Zhe Gan, Yu Cheng, Ahmed El Kholy, Linjie Li, Jingjing Liu, and Jianfeng Gao. Multi-step reasoning via recurrent dual attention for visual dialog. In *ACL*, pages 6463–6474, 2019. 2

[23] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv:2310.01801*, 2023. 3

[24] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *ECCV*, pages 269–284, 2016. 3, 5

[25] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *Arxiv*, 2106.02036, 2021. 3

[26] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. In *CVPR*, pages 18973–18990, 2022. 2, 4, 7

[27] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J. Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina González, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, and et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. *arXiv:2311.18259*, 2023. 1

[28] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv:2309.03905*, 2023. 2

[29] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 3

[30] Ke Hong, Guohao Dai, Jiaming Xu, Qiuli Mao, Xiuhong Li, Jun Liu, Kangdi Chen, Hanyu Dong, and Yu Wang. Flashdecoding++: Faster large language model inference on gpus. *arXiv:2311.01282*, 2023. 3

[31] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5, 6

[32] Daoji Huang, Otmar Hilliges, Luc Van Gool, and Xi Wang. Palm: Predicting actions through language models @ ego4d long-term action anticipation challenge 2023. *arXiv:2306.16545*, 2023. 7, 8

[33] Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos ”in the wild”. *Comput. Vis. Image Underst.*, 155:1–23, 2017. 5

[34] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. 1

[35] Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, pages 201–214, 2012. 3

[36] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. *arXiv:2309.06180*, 2023. 3

[37] 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. 2, 3

[38] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: localized, compositional video question answering. In *EMNLP*, pages 1369–1379, 2018. 3

[39] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, pages 7331–7341, 2021. 7

[40] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv:2305.03726*, 2023. 2, 3

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

[42] 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. 2, 3

[43] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023. 2, 3

[44] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. MM-VID: advancing video understanding with gpt-4v(ision). *arXiv:2310.19773*, 2023. 2, 3

[45] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. *arXiv:2206.01670*, 2022. 4, 7, 8

[46] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017. 7, 8

[47] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *CVPR*, pages 13843–13853, 2022. 7

[48] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023. 2, 5

[49] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023. 2, 5

[50] Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Ion Stoica, Zhijie Deng, Alvin Cheung, and Hao Zhang. Online speculative decoding. *arXiv:2310.07177*, 2023. 3

[51] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv:2306.09093*, 2023. 2, 3

[52] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv:2306.05424*, 2023. 2, 3

[53] Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action anticipation. In *WACV*, pages 6037–6046, 2023. 7

[54] 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*, pages 2630–2640, 2019. 4

[55] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. Anymal: An efficient and scalable any-modality augmented language model. *arXiv:2309.16058*, 2023. 2

[56] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv:2305.15021*, 2023. 2, 3

[57] Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *ECCV*, pages 336–352, 2020. 2

[58] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. *arXiv:2303.13519*, 2023. 7

[59] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. Efficient attention mechanism for visual dialog that can handle all the interactions between multiple inputs. In *ECCV*, pages 223–240, 2020. 2

[60] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt/, 2023. 1, 2

[61] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023. 1

[62] OpenAI. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. 2, 3

[63] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 1, 2

[64] 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. 2

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 2, 5, 6

[66] Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G. Schwing. Factor graph attention. In *CVPR*, pages 2039–2048, 2019. 2

[67] Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. Flexgen: High-throughput generative inference of large language models with a single gpu. *arXiv:2303.06865*, 2023. 3

[68] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016. 7

[69] 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, 3

[70] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *NeurIPS*, 2023. 5

[71] 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*, pages 1207–1216, 2019. 2, 4, 5, 7, 8

[72] 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. 1, 4

[73] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023. 2, 4, 5, 6

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 6000–6010, 2017. 2

[75] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *ICCV*, pages 7545–7555, 2021. 3

[76] Yue Wang, Shafiq R. Joty, Michael R. Lyu, Irwin King, Caiming Xiong, and Steven C. H. Hoi. VD-BERT: A unified vision and dialog transformer with BERT. In *EMNLP*, pages 3325–3338, 2020. 2

[77] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv:2309.05519*, 2023. 2

[78] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv:2309.17453*, 2023. 3

[79] Zihui Xue, Yale Song, Kristen Grauman, and Lorenzo Torresani. Egocentric video task translation. In *CVPR*, pages 2310–2320, 2023. 7

[80] 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*, pages 13623–13633, 2023. 7

[81] 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*, pages 10714–10726, 2023. 2, 3

[82] Zhewei Yao, Xiaoxia Wu, Conglong Li, Minjia Zhang, Heyang Qi, Olatunji Ruwase, Ammar Ahmad Awan, Samyam Rajbhandari, and Yuxiong He. Deepspeed-visualchat: Multi-round multi-image interleave chat via multi-modal causal attention. *arXiv:2309.14327*, 2023. 2, 3

[83] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv:2310.07704*, 2023. 2, 3

[84] Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for {Transformer-Based} generative models. In *OSDI*, pages 521–538, 2022. 3

[85] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv:2309.02591*, 2023. 2

[86] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv:2306.02858*, 2023. 2, 3

[87] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv:2307.03601*, 2023. 2, 3

[88] Qi Zhao, Ce Zhang, Shijie Wang, Changcheng Fu, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos? In *ICLR*, 2024. 3, 7, 8

[89] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *ECCV*, pages 485–502, 2022. 3

[90] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, pages 6586–6597, 2023. 4

[91] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *CVPR*, pages 14825–14835, 2023. 7

[92] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *CVPR*, pages 10727–10738, 2023. 7

[93] 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. 2