LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models

Yanwei Li[1⋆] Chengyao Wang[1⋆] Jiaya Jia[1,2]

CUHK[1] SmartMore[2]

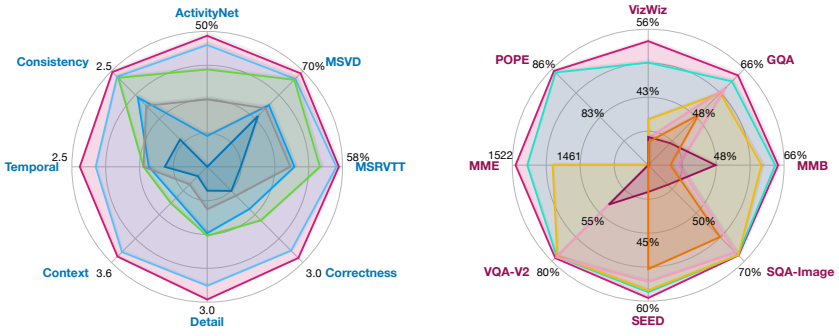
Abstract. In this work, we present a novel method to tackle the token generation challenge in Vision Language Models (VLMs) for video and image understanding, called LLaMA-VID. Current VLMs, while proficient in tasks like image captioning and visual question answering, face computational burdens when processing long videos due to the excessive visual tokens. LLaMA-VID addresses this issue by representing each frame with two distinct tokens, namely context token and content token. The context token encodes the overall image context based on user input, whereas the content token encapsulates visual cues in each frame. This dual-token strategy significantly reduces the overload of long videos while preserving critical information. Generally, LLaMA-VID empowers existing frameworks to support hour-long videos and pushes their upper limit with an extra context token. It is demonstrated to surpass previous methods on most of video- or image-based benchmarks. Code and models are available at https://github.com/dvlab-research/LLaMA-VID.

1 Introduction

Large Language Models (LLMs) [44, 54, 66], through their capacity to generate contextually accurate responses, have significantly advanced the field of AI. Drawing from the strengths of LLMs, Vision Language Models (VLMs) [13,33,45] have been developed to extend these capabilities to visual data, demonstrating their adeptness in tasks like image captioning and visual question answering. However, a substantial challenge emerges in the context of long video, where an excessive number of tokens are required to represent consecutive frames. The computational demands escalate with the video length, thereby constraining the practical application of VLMs for extensive videos.

Recently, several approaches have been proposed to handle videos, moving beyond image-only VLMs. These methods aim to alleviate the token issue by utilizing representative queries [30,64] or applying temporal compression [38,39,52]. Despite these efforts, the challenge of long videos remains unresolved. The primary obstacle stems from the excessive number of tokens required for each video frame. For instance, models like BLIP [13,29] and LLaVA [33] require 32 and over 256 tokens respectively for a single image. A video containing 10K frames would thus necessitate over 320K tokens, exceeding the capacity of current VLMs.

⋆ equal contribution



(a) Video-based benchmarks **(b)** Image-based benchmarks

Fig. 1: The proposed LLaMA-VID achieves leading performance on most of benchmarks with 7B LLMs. The video-based and image-based benchmarks are noted in blue and purple color, respectively. Please refer to Tables 1, 2, and 5 for more details.

Furthermore, simple temporal compression can significantly damage the representation over long-term intervals. This drawback hampers their performance, thereby underscoring the need for a robust solution.

In this work, we present LLaMA-VID, a novel approach to effectively manage the token generation issue in long videos. Our core idea is to represent each video frame with two distinct tokens: *context token* and *content token*. The context token is designed to encode the overall context of the image based on user input, which efficiently condenses the broader picture into *a single token*. Simultaneously, the content token captures finer aspects of each frame. According to computational constraints, the length of content token can be extended to include more details, *e.g.*, 1 token/frame for video input and beyond 256 token/frame for single image. In this way, the overload of long videos can be significantly reduced without sacrificing critical information.

In particular, our method employs a dual-token generation strategy that is both efficient and effective. For each frame, we first extract image features using a pre-trained vision transformer [15], akin to other VLMs [13, 33]. The key question is how to generate the context-related token according to user instructions. We provide the solution by leveraging the cross-modality design [14,29] for instruction-guided queries, which carry the interactive intention from users. For *context token*, these queries interact with previously generated image features in the designed attention module, termed as context attention. To generate *content token*, the image features are average pooled to formulate tokens that adapt to different settings. For instance, global pooling is adopted to maintain efficiency for video input while details are preserved with more tokens for single image input. The context and content tokens are subsequently projected to the space of LLMs with simple linear layers for final prediction. Furthermore, to better sup-

port hour-long videos in VLMs, we construct an instruction dataset that contains 9K movie-level conversations for plot reasoning and detail understanding.

Generally, LLaMA-VID can be distinguished from two aspects. On one hand, with the dual-token paradigm, each frame can be efficiently encoded with only two tokens, which empowers existing LLMs to support long videos. On the other hand, the context token aggregates the most informative feature of each image, which further extends the upper limit of VLMs with an extra token.

The overall framework, dubbed LLaMA-VID, can be easily instantiated with various decoders and LLMs, as elaborated in Section 3. Extensive empirical studies are conducted in Section 4 to reveal the effectiveness of each component. Remarkably, our model can complete training within 2 days on a single machine with 8×A100 GPUs, and it outperforms previous leading methods on most of video- and image-based benchmarks, as shown in Figure 1.

2 Related Work

In this section, we first review large language models and delve into recent advances in vision language models.

2.1 Large Language Models

The field of Natural Language Processing (NLP) has witnessed tremendous advancements with the evolution of LLMs. Transformer [55] marked a pivotal milestone, with subsequent language models [14, 35, 66] demonstrating remarkable capabilities. GPT [6] revolutionized this field by utilizing generative pre-trained transformers for auto-regressive prediction, which is proved to be a potent language modeling paradigm. Recent groundbreaking works, such as ChatGPT [44], GPT-4 [45], and LLaMA [54], have pushed the boundaries even further. Trained on vast amounts of text data, these models exhibit exceptional capabilities in complex linguistic tasks. To leverage the potential of pre-trained LLMs, instruction tuning [46, 57] is a crucial component for high-quality output. This strategy is widely adopted in open-source models like Alpaca [53] and Vicuna [12], which improve over LLaMA [54] using specially designed instruction pairs. There are also researches [58, 61] that utilize the reasoning ability of LLMs and invoke pre-defined tools for visual applications. Different from them, we collect multi-modality instruction data that contains text, images, and videos in this work, which is employed to empower LLMs for long video processing.

2.2 Vision Language Models

The advancements in computer vision and NLP have led to the emergence of vision-language models (VLMs) that integrate vision models with language models for cross-modality understanding [10, 59] and reasoning [19, 27, 37]. Pioneering large-scale VLMs like CLIP [47] and ALIGN [24] have extended language models to vision-language tasks. The recent progress has seen an increasing focus

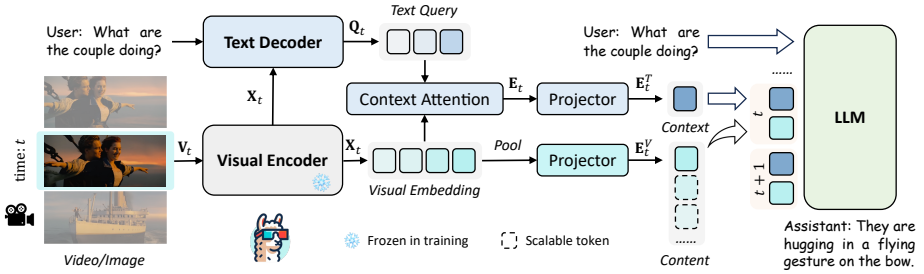


Fig. 2: The framework of LLaMA-VID. With user directive, it operates by taking either a *single image* or *video frames* as input, and generates responses from LLM. The process initiates with a visual encoder that transforms input frames into the visual embedding. Then, the text decoder produces text queries based on the user input. In context attention, the text query aggregates text-related visual cues. For efficiency, an option is provided to downsample the visual embedding to various token sizes, or even to a single token. The text-guided *context token* and the visually-enriched *content token* are then formulated using a linear projector to represent each frame at time t. Finally, the LLM takes the user directive and all visual tokens as input and gives responses.

on leveraging the power of LLMs. Notably, Flamingo [2] and BLIP-2 [29] utilize web-scale image-text pairs for cross-modality alignment, thereby enhancing learning performance. To further exploit the potential of such pre-trained models, InstructBLIP [13] and MiniGPT-4 [67] construct high-quality instruction pairs based on BLIP-2 and achieve superior results. Simultaneously, LLaVA [33] employs a simple linear projector with a few learnable parameters to align the image and text space of LLaMA. Given the tailored instruction data, this straightforward approach demonstrates strong capabilities. To support video understanding in LLMs, several studies [30, 52, 64] attempt to utilize BLIP-2 for video embedding or text-only caption [63] extraction, while Video-ChatGPT [39] proposes spatial and temporal pooling for video features. However, given the substantial number of tokens required for each frame, LLMs encounter significant challenges when processing extensive video sequences. It prevents previous work from representing long video sequences that exceed a duration of one hour in LLMs. To solve the issue, we propose to efficiently encode each frame with only 2 tokens, which supports long video understanding in existing LLMs.

3 LLaMA-VID

The framework of LLaMA-VID is conceptually simple: encoder and decoder are adopted to produce visual embedding and text-guided features, respectively; context token and content token are transformed with the tailored token generation strategy; instruction tuning is designed to unleash the potential of LLMs for image and video.

3.1 Encoder and Decoder

The proposed LLaMA-VID can be utilized to interact with single image or long videos. For clarity, we assume the input image is captured from a video sequence, as presented in Figure 2. Given a video frame $\mathbf{V}_t \in \mathbb{R}^{H\times W\times 3}$ at time t, a transformer-based visual encoder is first employed to produce the visual embedding $\mathbf{X}_t \in \mathbb{R}^{N\times C}$. Here, $N = H/p \times W/p$ and C indicate the number of image patches and embedding channels, respectively. The patch size p is typically set to 14 for ViT-based backbones [15, 16, 47]. Meanwhile, we take the user instruction as input and generate the text-guided query $\mathbf{Q}_t \in \mathbb{R}^{M\times C}$ with the produced $\mathbf{X}_t$, where M denotes the number of queries. As depicted in Figure 2, this cross-modality interaction predominantly occurs in the text decoder, which can be easily instantiated with BERT [14] or QFormer [13], as compared in Table 8. In this way, the text query $\mathbf{Q}_t$ contains highlighted visual cues that are most related to the user instruction.

3.2 Token Generation

With the text query $\mathbf{Q}_t$ and visual embedding $\mathbf{X}_t$, we can easily generate representative tokens for LLMs. Specifically, context attention is designed to aggregate text-related visual features and condense them to a single context token. As shown in Figure 2, it takes $\mathbf{Q}_t$ and $\mathbf{X}_t$ as input and formulates the context-related embedding $\mathbf{E}_t \in \mathbb{R}^{1\times C}$ as

$$\mathbf{E}_t = \text{Mean}(\text{Softmax}((\mathbf{Q}_t \times \mathbf{X}_t^{\mathsf{T}})/\sqrt{C}) \times \mathbf{X}_t), \tag{1}$$

where the Softmax function and Mean operation are conducted along the N and M dimensions, respectively. Unlike QFormer [13] that adopts 32 visual queries as LLMs tokens, we only utilize the text query $\mathbf{Q}_t$ to aggregate the visual features with high-response scores to input instructions. As a result, the most crucial visual cues related to user input are efficiently preserved in the condensed embedding $\mathbf{E}_t$. The effectiveness of this context-related token generation is demonstrated in Table 6 and Figure 6. Subsequently, a linear projector is utilized to transform the embedding $\mathbf{E}_t$ into the context token $\mathbf{E}_t^T \in \mathbb{R}^{1\times C}$, which aligns with the language space of LLMs. Meanwhile, we employ an adaptive pooling strategy for the visual embedding according to computational constraints to produce the content token $\mathbf{E}_t^V \in \mathbb{R}^{n\times C}$, where $n \in [1, N]$. For instance, we maintain the original resolution of visual embedding $\mathbf{X}_t$ when input single image, while we downsample $\mathbf{X}_t$ to 1 token for long videos. This approach significantly reduces the overload of LLMs for each frame, thereby supporting hour-long videos effectively. Finally, the generated context token $\mathbf{E}_t^T$ and the content token $\mathbf{E}_t^V$ are concatenated to represent the frame at time t. Along with frames at other timestamps, the entire video sequence is translated into the language space in token format, which is then used to generate responses from LLMs. The whole process is summarized in Algorithm 1.

Algorithm 1 Pseudo Code for Token Generation.

```
# B: batch size; C: channel size; n: content shape
# M: query length; N: shape of flatten image pacthes;
# text_q: text query in shape (B, M, C)
# vis_embed: visual embedding in shape (B, N, C)

# Key part 1: calculate context-related embedding
ctx_embed = text_q @ vis_embed.transpose(-1,-2)
ctx_embed = ctx_embed / (vis_embed.shape[-1]**0.5)
ctx_embed = (ctx_embed.softmax(-1)@vis_embed).mean(1)
ctx_embed = self.ctxproj(ctx_embed[:,None])

# Key part 2: calculate visual embedding
cur_shape = int(vis_embed.shape[1]**0.5)
vis_embed = vis_embed.reshape(B, cur_shape, -1, C)
vis_embed = F.avg_pool2d(vis_embed.permute(0,3,1,2), kernel_size=cur_shape//n,
    stride=cur_shape//n)
vis_embed = vis_embed.permute(0,2,3,1).flatten(1,2)
vis_embed = self.visproj(vis_embed)

# concat token in shape (B, n+1, C), n in [1,N]
final_token = torch.cat([ctx_embed, vis_embed], dim=1)
```

F: torch.nn.functional; `ctxproj`, `visproj`: predefined linear projectors.

3.3 Training Strategy

Training strategy, particularly instruction tuning, has proven to be crucial in LLMs [12, 53, 54] and VLMs [13, 32, 33]. Considering training efficiency, in this work, we divide the training procedure into three stages, *i.e.*, modality alignment, instruction tuning, and long video tuning.

Modality Alignment. As shown in Figure 2, each video frame is projected into the space of LLMs in each forward pass. Therefore, it is crucial to ensure visual features are well aligned with the language space. To this end, we construct a compact dataset that contains 790K high-quality image- and video-caption pairs.

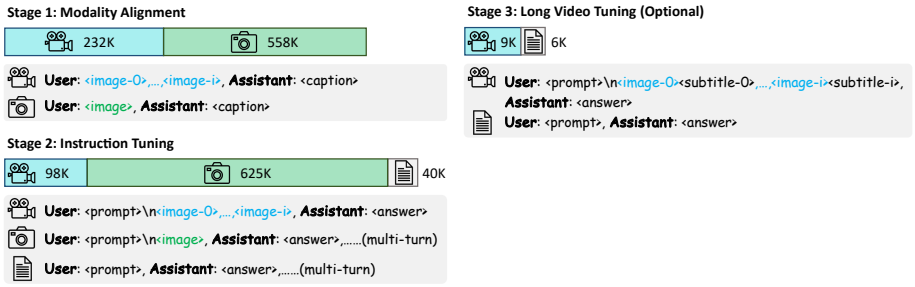


Fig. 3: Multimodal data distribution and instruction format for each model training stage. `<image>` and `<image-i>` denote the token for single image and the i-th video frame, respectively.

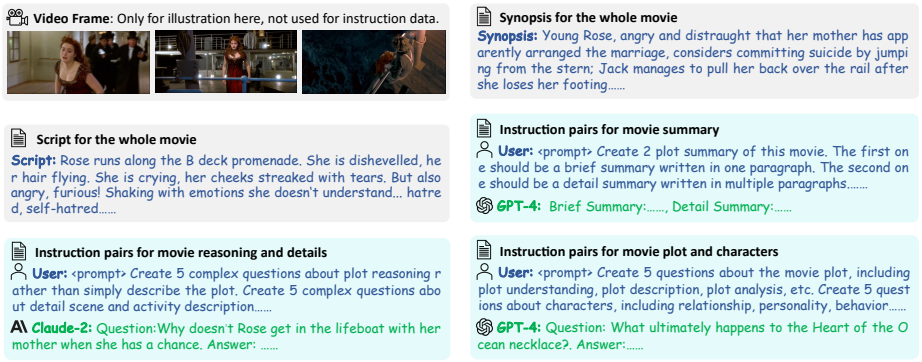


Fig. 4: An example to construct instruction pairs for the movie Titanic. Given the movie synopsis and script, we utilize the developed LLMs like GPT-4 [45] and Claude-2 [3] to generate movie summaries, plot-related QA pairs, and general reasoning pairs.

As illustrated in Figure 3, it mainly consists of 558K image-caption pairs from the LLaVA-filtered CC3M dataset [49] and 232K video-caption pairs sampled from the WebVid 2.5M dataset [5]. The instruction format for each modality is presented in Figure 3. In this stage, we optimize the projectors in Figure 2, while freezing the pre-trained modules like the visual encoder and text decoder.

Instruction Tuning. To enhance the multi-modality understanding of LLMs, we build the instruction pairs from [32] and [39]. In particular, the constructed dataset mainly involves content from three sources, *i.e.*, 40K text conversations from ShareGPT [1], 625K single- or multi-turn visual QA pairs from [19, 22, 25, 26, 33, 41–43, 48, 50], and 98K video QA pairs from [7]. For the instruction, we adopt different formats for text, image, and video input, as shown in Figure 3. The input prompt `<prompt>` and answer `<answer>` vary with datasets. Please refer to [32] and [39] for more details. Meanwhile, the image token `<image-i>` is randomly inserted at the beginning or end of the user input during our training. In instruction tuning, all the modules are optimized except the visual encoder.

Long Video Tuning. To further unleash the potential for hour-long videos, we construct 15K long QA pairs, including 9K conversions in movie scenes and 6K data sampled from LongLoRA [11] for token expanding. Specifically, we utilize more than 400 long movies and corresponding scripts in MovieNet [21] to build the training set. The key components for instruction generation are visualized in Figure 4. Generally, the generated dataset includes QA pairs from three aspects: video summary, movie plot, and detail reasoning. For video summaries, we collect movie synopses to produce brief and detailed summaries for each movie using developed LLMs like GPT-4 [45]. It brings about 1K summary-level instruction pairs in total. For plot-level data, we take the entire movie synopsis as input and leverage GPT-4 [45] to generate plot-related and character-related QA pairs. These include plot understanding, description, analysis, character relationship,

personality, and behavior. In particular, we generate 5 plot-related pairs and 5 character-related pairs for each movie, resulting in 4K plot-level QA data. As for detail-level data, we feed the long movie script into Claude-2 [3] and generate 5 plot-related reasoning pairs and 5 detail-related descriptions for each movie, which brings 4K pairs in total. With long videos and the generated pairs, we perform instruction tuning by concatenating visual tokens and subtitle tokens for each frame, as depicted in Figure 3. In this way, LLaMA-VID can well support 64K tokens with more than 3-hour video as input.

4 Experiments

In this section, we provide the experimental setup and comparisons with leading methods on several benchmarks.

4.1 Experimental Setup

Implementation Details. In this work, we instantiate the model with the pre-trained EVA-G [16] for visual encoder and QFormer [13] for text decoder by default. During training, we keep the visual encoder fixed in all stages and freeze the text decoder, as well as the LLM, in the modality alignment stage, except for the BERT module in Table 8 that is not pre-trained. Following the strategy in [32], we optimize trainable parameters with the designed data and instructions in Figure 3, running for 1 epoch in each stage. For video input, we extract frames at 1 FPS. All models are trained using 8×NVIDIA A100 GPUs.

Datasets. In this study, we construct the training set mainly from [5,21,32,39], as illustrated in Section 3.3. Moreover, we report results on several video- and image-based benchmarks. In particular, for video input, we evaluate the zero-shot performance on the open-ended QA benchmarks like MSVD [8], MSRVTT [59], ActivityNet [7], EgoSchema [40], SSv2 (Something-Something v2) [18], and the newly-proposed generative performance benchmark [39]. As for image-based evaluation, we conduct experiments on several widely-adopted benchmarks, including GQA [22], MMB (MMBench) [36], MME [17], POPE [31], SEED [28], SQA^{I} [37], VQA^{T} (TextVQA) [51], VizWiz [20], and VQA^{v2} (VQA V2) [19].

4.2 Main Results

Results on Video-based Benchmarks. In Table 1, we provide a comparative evaluation of LLaMA-VID against various state-of-the-art methods across zero-shot video QA benchmarks: MSVD-QA [8], MSRVTT-QA [59], ActivityNet-QA [7], EgoSchema [40], and SSv2 [18]. Notably, the results are reported with only two tokens for each frame. It is evident that LLaMA-VID, employing Vicuna-7B and Vicuna-13B as the LLMs, consistently delivers superior performance across all datasets. On the MSVD-QA and MSRVTT-QA datasets, it

Table 1: Comparison with leading methods on 3 zero-shot video QA datasets. We report results with 2 tokens for each frame. For fair comparisons, our model is trained without long video tuning (stage 3) in Figure 3. Res indicates image resolution.

Method	LLM	Res.	MSVD-QA		MSRVTT-QA		ActivityNet-QA	
			Acc	Score	Acc	Score	Acc	Score
VideoLLaMA [64]	Vicuna-7B	224	51.6	2.5	29.6	1.8	12.4	1.1
LLaMA-Adapter [65]	LLaMA-7B	224	54.9	3.1	43.8	2.7	34.2	2.7
VideoChat [30]	Vicuna-7B	224	56.3	2.8	45.0	2.5	26.5	2.2
Video-ChatGPT [39]	Vicuna-7B	224	64.9	3.3	49.3	2.8	35.2	2.7
BT-Adapter [34]	Vicuna-7B	–	67.5	**3.7**	57.0	3.2	45.7	3.2
LLaMA-VID	Vicuna-7B	224	69.7	**3.7**	57.7	3.2	47.4	**3.3**
LLaMA-VID	Vicuna-13B	224	**70.0**	**3.7**	**58.9**	**3.3**	**47.5**	**3.3**

Table 2: Comparison with leading methods on the video generative benchmark [39]. We report results with 2 tokens for each frame. For fair comparisons, our model is trained without long video tuning (stage 3) in Figure 3. Res indicates image resolution. *Correct*, *Detail*, *Context*, *Temporal*, and *Consist* indicate evaluation metrics in [39].

Method	LLM	Res.	Correct.	Detail	Context	Temporal	Consist.
VideoLLaMA [64]	Vicuna-7B	224	1.96	2.18	2.16	1.82	1.79
LLaMA-Adapter [65]	LLaMA-7B	224	2.03	2.32	2.30	1.98	2.15
VideoChat [30]	Vicuna-7B	224	2.23	2.50	2.53	1.94	2.24
Video-ChatGPT [39]	Vicuna-7B	224	2.40	2.52	2.62	1.98	2.37
BT-Adapter [34]	Vicuna-7B	–	2.68	2.69	3.27	2.34	2.46
LLaMA-VID	Vicuna-7B	224	2.96	3.00	3.53	2.46	2.51
LLaMA-VID	Vicuna-13B	224	**3.07**	**3.05**	**3.60**	**2.58**	**2.63**

Table 3: Comparison with leading methods on the zero-shot benchmark EgoSchema [40]. We report results with 2 tokens for each frame on different settings.

Frame	FrozenBiLM [60]	mPLUG-Owl [62]	InternVideo [56]	**LLaMA-VID**
10	26.4%	29.6%	31.4%	**41.2**%
30	–	20.0%	31.8%	**41.7**%
90	26.9%	–	32.1%	**40.5**%
180	–	–	–	**41.7**%

achieves the accuracy of 69.7% and 57.7% with Vicuna-7B, surpassing the previous leading approach [34] with absolute gains of 2.2% and 0.7%, respectively. For the ActivityNet-QA dataset, LLaMA-VID attains top performance in accuracy and the highest score of 3.3. In Table 2, we also carry out experiments on the newly proposed video-based generative performance benchmark [39]. As for the long EgoSchema and SSv2 dataset, the proposed method surpasses other leading approaches in different frame settings, as presented in Table 3 and Table 4. Specifically, we evaluate the model on the SSv2 dataset similar with that in MSRVTT-QA. Our LLaMA-VID is validated to achieve the best performance across all the evaluation metrics, surpassing previous approaches by a large margin. In general, LLaMA-VID is proved to bring robust results on all benchmarks.

Table 4: Comparisons with zero-shot methods on SSv2 [18] dataset.

Method	LLM	Res.	Acc	Score
Video-ChatGPT [38]	Vicuna-7B	224	11.59	2.10
LLaMA-VID	Vicuna-7B	224	**20.12**	**2.14**

Table 5: Comparison with leading methods on 8 benchmarks. Here, we use the same training and instruction finetuning data as that in LLaVA-1.5. We report results with 1 context token and n content tokens, where n is kept the same with that in LLaVA-1.5, *i.e.*, $n = (336/14)^2 = 576$. Res indicates input image resolution. * and † denote the *train* subset is included for training and the data is not publicly available, respectively.

Method	LLM	Res.	GQA	MMB	MME	POPE	SEED	SQA^I	VizWiz	VQA^{v2}
InstructBLIP [13]	Vicuna-7B	224	49.2	36.0	–	–	53.4	60.5	34.5	–
IDEFICS-9B [23]	LLaMA-7B	224	38.4	48.2	–	–	–	–	35.5	50.9
Qwen-VL† [4]	Qwen-7B	448	59.3*	38.2	–	–	56.3	67.1	35.2	78.8*
Qwen-VL-Chat† [4]	Qwen-7B	448	57.5*	60.6	1487.5	–	58.2	68.2	38.9	78.2*
LLaVA-1.5 [32]	Vicuna-7B	336	62.0*	64.3	1510.7	85.9	58.6	66.8	50.0	78.5*
LLaMA-VID	Vicuna-7B	336	**64.3***	**65.1**	**1521.4**	**86.0**	**59.9**	**68.3**	**54.2**	**79.3***
BLIP-2 [29]	Vicuna-13B	224	41.0	–	1293.8	85.3	46.4	61.0	19.6	41.0
InstructBLIP [13]	Vicuna-13B	224	49.5	–	1212.8	78.9	–	63.1	33.4	–
Shikra [9]	Vicuna-13B	224	–	58.8	–	–	–	–	–	77.4*
IDEFICS-80B [23]	LLaMA-65B	224	45.2	54.5	–	–	–	–	36.0	60.0
LLaVA-1.5 [32]	Vicuna-13B	336	63.3*	**67.7**	1531.3	85.9	61.6	**71.6**	53.6	**80.0***
LLaMA-VID	Vicuna-13B	336	**65.0***	66.6	**1542.3**	**86.0**	**62.3**	70.0	**54.3**	**80.0***

Results on Image-based Benchmarks. As illustrated in Section 3.2, LLaMA-VID not only efficiently represents long video, but also expands the upper limit of VLMs with an additional context token. In Table 5, we perform comparisons with leading VLMs on 8 widely-adopted benchmarks. It should be noted that we maintain the same training data and image resolution as in LLaVA-1.5 [32] for fair comparisons. It is evident that LLaMA-VID outperforms other leading methods across most of benchmarks with different LLMs. In particular, with Vicuna-7B, LLaMA-VID attains the best results across all the datasets and surpasses LLaVA-1.5 with significant gains in GQA, MME, and VizWiz, where the improvement reaches up to 2.3%, 10.7, and 4.2%, respectively. With a larger Vicuna-13B, LLaMA-VID also outperforms other methods in 6 benchmarks and achieves top-2 in the other datasets. This demonstrates the generality of LLaMA-VID, which can be scaled up with a stronger foundation model.

Qualitative Results. In Figure 5, we apply LLaMA-VID to different types of data, including short videos and long movies. We represent each frame with 2 tokens for videos. LLaMA-VID demonstrates various capabilities with different inputs. Given a short trailer video, LLaMA-VID summarizes the overall plot and infers the movie name, producer, and the type of movie. It shows that the proposed LLaMA-VID can well recognize the typical characters with the produced tokens. As for a 3-hour movie, the proposed model adeptly describes the storyline and demonstrates plot-related reasoning and detailed understanding.

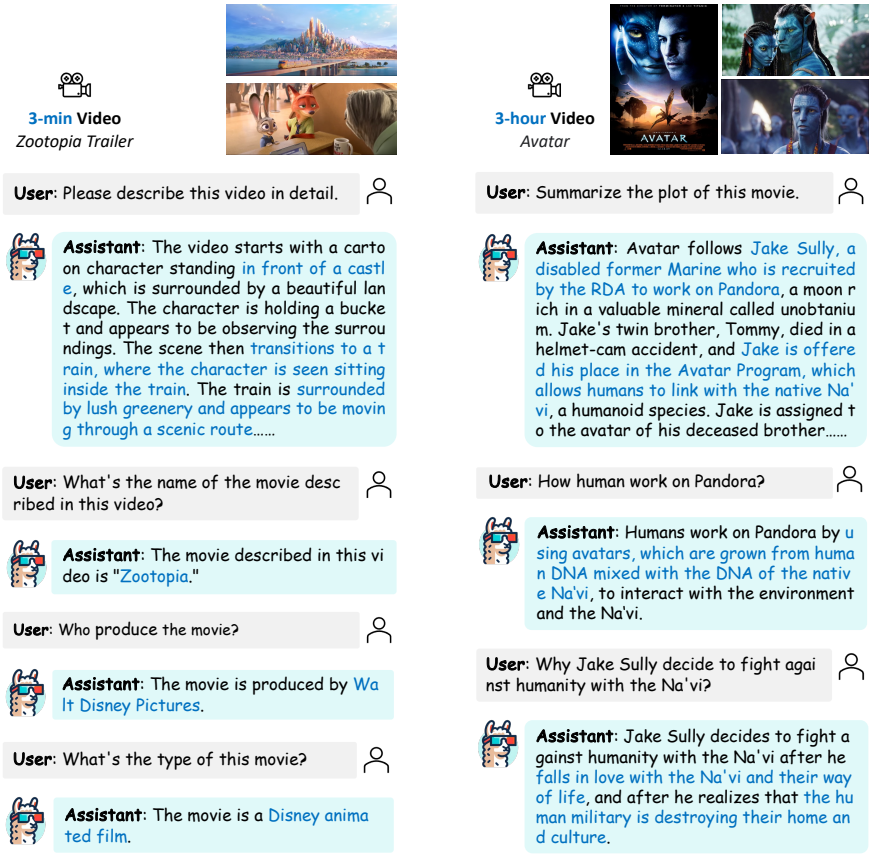


Fig. 5: Examples of LLaMA-VID with short video and hour-long video.

4.3 Component-wise Analysis

In this subsection, we conduct ablation studies with input resolution 224 and 2 tokens for each image by default.

Generated Token Types. As illustrated in Figure 2, each image is represented with a context token and a content token in LLMs. To validate the effectiveness of each part, we conduct experiments with different types of tokens in Table 6. Without the context token, the compressed content token, which encodes each image with 1 token, cannot adjust to input instructions, leading to subpar performance. Compared with a single content token, the instruction-guided context token results in significant gains across all datasets with only 1 token. With both tokens for each image, the model achieves the best performance across all image- and video-based benchmarks. It shows that both instruction cues in the context token and the image content itself in the content token are important.

Table 6: Comparison with different token types. We report results with 1 *context* token (if exists) and 1 *content* token on image-based and video-based benchmarks (MSVD).

context	*content*	**GQA**	**POPE**	**SQA^{I}**	**VQA^{T}**	**MSVD**
✗	✓	53.3	80.9	66.1	46.5	65.9
✓	✗	54.3	82.4	67.7	48.3	69.3
✓	✓	**55.5**	**83.1**	**68.8**	**49.0**	**69.7**

Table 7: Comparison with different token numbers. We report results with various numbers of *context* token and *content* token on image-based benchmarks for efficiency.

context	*content*	**GQA**	**POPE**	**SQA^{I}**	**VQA^{T}**
0	256	61.9	85.5	67.5	53.0
1	256	**63.0**	**86.6**	67.7	**53.8**
1	64	60.8	85.1	68.7	52.3
1	16	58.2	83.1	67.4	50.8
1	4	56.2	83.5	68.7	49.1
1	1	55.5	83.1	**68.8**	49.0

Generated Token Numbers. In Table 7, we conduct experiments with different numbers of tokens for further investigation. With an image size 224×224, we set up experiments with n content tokens, where $n = (224/14)^2 = 256$ for uncompressed settings in the first two rows. The results clearly show that the context token consistently improves performance across different benchmarks with only 1 extra token. When we compress the content token to 1/4 with $n = 64$, the performance drops about 1% to 2% but increases 1% in SQA^{I}. Considering the extra efficient setting for hour-long videos, we compress the content token to 1/256 with $n = 1$ by default. Compared to the original setting without context token, we can reduce the computational cost to 1/128 with about 2%-6% performance drop, which is generally acceptable. The linear increase in performance presents significant potential for token compression. For instance, we can dynamically compress the content token to different numbers according to resource budget and content importance. Interestingly, the model achieves peak performance in SQA^{I} with only 2 tokens. This could be attributed to the fact that problems in ScienceQA [37] focus more on visual-based reasoning rather than image details. As demonstrated in Tables 1 and 2, with only 2 tokens for each image, LLaMA-VID still outperforms all previous work in different video-based benchmarks. This makes it feasible to enable LLMs for hour-long video processing.

Text Decoder. As depicted in Figure 2, the text decoder plays an essential role in producing instruction-guided context cues. Here, we further perform comparisons with different text decoders in Table 8. We mainly instantiate the text decoder with two types of modules, namely BERT [14] and QFormer [13]. For BERT, we randomly initialize it as a cross-modality decoder and only retain the first two layers. As for QFormer, we utilize the pre-trained modules and fix them

Table 8: Comparison with different text decoders. We report results with 1 *context* token (if exists) and 1 *content* token on image-based benchmarks for efficiency.

text	**GQA**	**POPE**	**SQA^{I}**	**VQA^{T}**
–	53.3	80.9	66.1	46.5
BERT	54.1	80.8	67.9	48.1
QFormer	**55.5**	**83.1**	**68.8**	**49.0**

Table 9: Average module latency of video frame on a A100 GPU. *Encoder*, *Content*, *Decoder*, and *Context* denote the latency for text-free visual encoder and content token, text-related text decoder and context token generation, respectively.

text	**Text-free**		**Text-related**		**Total** (ms)
	Encoder	Content	Decoder	Context	
BERT	4.89	0.01	0.25	0.02	5.17
QFormer	4.87	0.01	1.08	0.02	5.98

for modality alignment. Even with a simple 2-layer BERT, as shown in Table 8, the generated context token achieves significant gains in most of benchmarks. This proves the effectiveness of the paradigm for context token generation. With a pre-trained text decoder like QFormer, the model can be further enhanced and attains peak performance in all datasets with 2.2% to 2.7% significant gain.

Latency Analysis. To further analyse the module latency, we keep the original setting in Table 8 and report average module latency over 100 videos in Table 9. Here, we divide the modules for generation into two types, namely text-free and text-related. In particular, we generate visual tokens one time for each frame with text-free modules. And text-related modules are required to produce different context tokens according to user inputs. It is clear that the text-free tokens account for the most computation, while the text-related modules only cost 5% to 18.3% for each user instruction. That means the total cost will not increase greatly given variant user questions because of the efficient text-related modules.

Response in Context Attention. To more vividly explore the context attention, we visualize the high response areas with the top 20 scores in Figure 6. Specifically, we draw the normalized heatmap for the first four queries in $\mathbf{Q}_t$ before applying the Softmax function, which is used to formulate context token in Equation 1. As shown in Figure 6, the text-guided query $\mathbf{Q}_t$ effectively focuses on important areas relevant to the input questions. For example, in the second row, when inquiring whether the image depicts a fishing village, the query $\mathbf{Q}_t$ focuses more on buildings along the river and a seagull. These are all typical characteristics to distinguish a fishing village in common sense. Other examples also confirm that the designed context attention successfully achieves its goal of formulating the context token under instruction guidance.

Fig. 6: High response areas with top scores to input question in Equation 1. We present the response of first four queries in $\mathbf{Q}_t$. Images are sampled from VQA V2 *test-dev* set.

5 Discussion and Conclusion

We have introduced LLaMA-VID, a simple yet effective token generation approach for VLMs. The central concept behind LLaMA-VID is to represent an image with the context token and the content token. In particular, the context token is generated according to input instructions, and the content token is produced based on the image content. Depending on the budget, the content token can be compressed to one token or expressed without compression. It allows us to represent a single image with preserved details and efficiently encode each video frame with only two tokens. Moreover, we have constructed an instruction dataset for hour-long video understanding. Our experiments on several video- and image-based benchmarks prove the superiority of our method. We hope that LLaMA-VID can serve as a strong benchmark for efficient visual representation.

There still exists certain limitations in the current method. Although LLaMA-VID succeeds to represent hour-long videos in LLMs, it focuses more on temporal consistency and may lose spatial details in each frame. In the future work, we plan to maintain dense spatial information for each frame outside the LLM and conduct the visual token fusion within LLMs to avoid such detail loss.

Acknowledgement. This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

References

1. Sharegpt. https://sharegpt.com/ (2023) 7
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. In: NeurIPS (2022) 4
3. Anthropic: Claude 2. https://www.anthropic.com/index/claude-2 (2023) 7, 8
4. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv:2308.12966 (2023) 10
5. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV (2021) 7, 8
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: NeurIPS (2020) 3
7. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015) 7, 8
8. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011) 8
9. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv:2306.15195 (2023) 10
10. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv:1504.00325 (2015) 3
11. Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., Jia, J.: Longlora: Efficient fine-tuning of long-context large language models. arXiv:2309.12307 (2023) 7
12. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/ (2023) 3, 6
13. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. arXiv:2305.06500 (2023) 1, 2, 4, 5, 6, 8, 10, 12
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018) 2, 3, 5, 12
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 2, 5
16. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale. In: CVPR (2023) 5, 8
17. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv:2306.13394 (2023) 8
18. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV (2017) 8, 10

19. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017) 3, 7, 8
20. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: CVPR (2018) 8
21. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding. In: ECCV (2020) 7, 8
22. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019) 7, 8
23. IDEFICS: Introducing idefics: An open reproduction of state-of-the-art visual language model. https://huggingface.co/blog/idefics (2023) 10
24. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021) 3
25. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP (2014) 7
26. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV (2017) 7
27. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. arXiv:2308.00692 (2023) 3
28. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv:2307.16125 (2023) 8
29. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv:2301.12597 (2023) 1, 2, 4, 10
30. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: Chat-centric video understanding. arXiv:2305.06355 (2023) 1, 4, 9
31. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv:2305.10355 (2023) 8
32. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv:2310.03744 (2023) 6, 7, 8, 10
33. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeruIPS (2023) 1, 2, 4, 6, 7
34. Liu, R., Li, C., Ge, Y., Shan, Y., Li, T.H., Li, G.: One for all: Video conversation is feasible without video instruction tuning. arXiv:2309.15785 (2023) 9
35. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019) 3
36. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv:2307.06281 (2023) 8
37. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: NeurIPS (2022) 3, 8, 12
38. Luo, R., Zhao, Z., Yang, M., Dong, J., Qiu, M., Lu, P., Wang, T., Wei, Z.: Valley: Video assistant with large language model enhanced ability. arXiv:2306.07207 (2023) 1

39. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv:2306.05424 (2023) 1, 4, 7, 8, 9
40. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. NeurIPS (2024) 8, 9
41. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016) 7
42. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016) 7
43. Mishra, A., Shekhar, S., Singh, A.K., Chakraborty, A.: Ocr-vqa: Visual question answering by reading text in images. In: ICDAR (2019) 7
44. OpenAI: Chatgpt. https://openai.com/blog/chatgpt/ (2023) 1, 3
45. OpenAI: Gpt-4 technical report. arXiv:2303.08774 (2023) 1, 3, 7
46. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. In: NeurIPS (2022) 3
47. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 3, 5
48. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-okvqa: A benchmark for visual question answering using world knowledge. In: ECCV (2022) 7
49. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018) 7
50. Sidorov, O., Hu, R., Rohrbach, M., Singh, A.: Textcaps: a dataset for image captioning with reading comprehension. In: ECCV (2020) 7
51. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: CVPR (2019) 8
52. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., et al.: Moviechat: From dense token to sparse memory for long video understanding. In: CVPR (2024) 1, 4
53. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca (2023) 3, 6
54. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. arXiv:2302.13971 (2023) 1, 3, 6
55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 3
56. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv:2212.03191 (2022) 9
57. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv:2109.01652 (2021) 3
58. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv:2303.04671 (2023) 3
59. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR (2016) 3, 8

60. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models. In: NeurIPS (2022) 9
61. Yang, R., Song, L., Li, Y., Zhao, S., Ge, Y., Li, X., Shan, Y.: Gpt4tools: Teaching large language model to use tools via self-instruction. arXiv:2305.18752 (2023) 3
62. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv:2304.14178 (2023) 9
63. Zhang, C., Lu, T., Islam, M.M., Wang, Z., Yu, S., Bansal, M., Bertasius, G.: A simple llm framework for long-range video question-answering. arXiv:2312.17235 (2023) 4
64. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv:2306.02858 (2023) 1, 4, 9
65. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv:2303.16199 (2023) 9
66. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv:2205.01068 (2022) 1, 3
67. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv:2304.10592 (2023) 4