# Video-XL:
# Extra-Long Vision Language Model for Hour-Scale Video Understanding

Yan Shu[1,2*] Zheng Liu[2,6*†] Peitian Zhang[2,3] Minghao Qin[2,4]
Junjie Zhou[2,5] Zhengyang Liang[2] Tiejun Huang[2,7] Bo Zhao[1,2†]

[1] School of AI, Shanghai Jiao Tong University [2] BAAI [3] RUC [4] CAS [5] BUPT [6] PolyU [7] PKU

https://github.com/VectorSpaceLab/Video-XL

## Abstract

*Long video understanding poses a significant challenge for current Multi-modal Large Language Models (MLLMs). Notably, the MLLMs are constrained by their limited context lengths and the substantial costs while processing long videos. Although several existing methods attempt to reduce visual tokens, their strategies encounter severe bottleneck, restricting MLLMs' ability to perceive fine-grained visual details. In this work, we propose* ***Video-XL****, a novel approach that leverages MLLMs' inherent key-value (KV) sparsification capacity to condense the visual input. Specifically, we introduce a new special token, the Visual Summarization Token (**VST**), for each interval of the video, which summarizes the visual information within the interval as its associated KV. The VST module is trained by instruction fine-tuning, where two optimizing strategies are offered. 1.* ***Curriculum learning****, where VST learns to make small (easy) and large compression (hard) progressively. 2.* ***Composite data curation****, which integrates single-image, multi-image, and synthetic data to overcome the scarcity of long-video instruction data. The compression quality is further improved by* ***dynamic compression****, which customizes compression granularity based on the information density of different video intervals. Video-XL's effectiveness is verified from three aspects. First, it achieves a superior long-video understanding capability, outperforming state-of-the-art models of comparable sizes across multiple popular benchmarks. Second, it effectively preserves video information, with minimal compression loss even at 16× compression ratio. Third, it realizes outstanding cost-effectiveness, enabling high-quality processing of thousands of frames on a single A100 GPU.*

## 1. Introduction

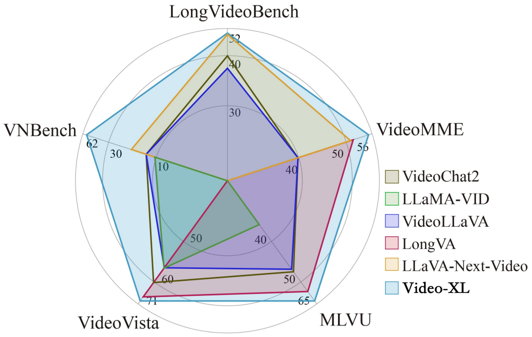


Figure 1. Video-XL achieves state-of-the-art results across several video benchmarks, surpassing other models of comparable sizes.

Multi-modal Large Language Models (MLLMs) have attracted widespread attention from the AI community. By augmenting large language models (LLMs) [34, 35, 42] with vision encoders, MLLMs are enabled to perform various vision-language modeling tasks, e.g., image captioning and visual question answering [15, 23, 28]. Recently, there has been growing interest in applying MLLMs for video understanding given MLLMs' proficiency in comprehending and reasoning over visual information [9, 16, 18, 38, 52].

However, it remains a significant challenge for the existing MLLMs to process long videos due to their limited context lengths and the huge costs involved. Particularly, a long video consists of a long sequence of frames, where each frame usually consumes a large number of visual tokens for MLLMs to perceive (e.g., 144 tokens per frame). As a result, the input may easily exceed the limits of MLLMs' context lengths. Even if the context lengths can be extended sufficiently, it will still take considerable computation and memory costs to process long videos, making it challenging in real-world scenarios. Recently, many studies have attempted to reduce the token count from the visual encoder for each frame [10, 18, 32, 41, 45]. While these approaches

[*] Equal contribution.

[†] Correspondence to <bo.zhao@sjtu.edu.cn> and <zhengliu1026@gmail.com>

enable handling longer input, they often lead to a substantial loss of visual information, which creates a severe bottleneck for MLLMs' fine-grained perception of long videos.

To address the existing challenges, we introduce a novel approach for long video understanding, called **Video-XL**. Unlike the existing methods which rely on the reduction of tokens generated from visual encoder, we leverage the LLMs' inherent KV sparsification capability to generate compact representations for long videos. Specifically, it's observed that LLMs tend to form sparse attention patterns when dealing with long inputs [25, 29]. This phenomenon suggests that the LLMs' inputs are secretly compressed, which allows them to perceive useful information from long contexts. Based on this inspiration, we design the following compression mechanism.

- **VST Compression**. We employ a new special token **VST** (**V**isual **S**ummarization **T**oken) to generate compact representations for long videos. The VSTs are assigned to different intervals of the video, which summarizes the visual information within the intervals (i.e., the original KVs from its preceding visual tokens) into their associated KVs. The VSTs' compressed KVs are maintained for future encoding, while the KVs from other visual tokens are offloaded. Thus, it enables a substantial cost reduction for long video processing. Knowing that different parts of a video exhibit variant information density, we propose **dynamic compression strategy**. Particularly, we form small intervals for the information-dense parts of the video; therefore, it enables fine-grained compression for the corresponding parts. On the contrary, we form large intervals for those information-sparse parts of the video, which can do with coarse-grained compression. With this operation, the visual information loss can be minimized given a fixed budget.

- **Training**. The VST module is trained by instruction fine-tuning. Given a video understanding task, the MLLM is required to generate VST compressed KVs for an input video; then, it is asked to predict the ground-truth answer based on the compression result. The training of VST is non-trivial given the challenges on the problem's complexity and the limitation of data. Therefore, we introduce the following strategies to enable the effective model training.

First, we propose to train VST by **curriculum learning**. Once training process is started, we perform a random sampling of small compression ratios for VST (e.g., $2\times$, $4\times$). With training process going on, we gradually sample larger compression ratios for VST (e.g., $8\times$, $16\times$). As it's easier to perform small compressions, the VST module may well establish its preliminary capability after the initial stage. Upon this foundation, the VST module can progressively learn larger compressions with higher proficiency.

Second, we employ a **composite data curation** method to create training data. Currently, long-video instruction data is still scarce; therefore, we exploit two extra resources to overcome this shortage. Considering that video understanding is built on top of the comprehension of images, we leverage *single-image* and *multi-image* with captioning and QA datasets for augmentation. The images are formatted as sequences of frames, which is made consistent with the video instruction data, and thus facilitates knowledge transfer. In addition, video understanding calls for precise retrieval of useful information for the given instruction. As a result, we create a *synthetic dataset*, called VIsual Clue Ordering (VICO), to strengthen this fundamental capability.

We implement Video-XL based on Qwen-2-7B, whose effectiveness can be verified from three perspectives. First, Video-XL outperforms state-of-the-art models of comparable sizes across popular long-video benchmarks, including VideoMME [7], MLVU [56], LongVideoBench [46], etc., as shown in Figure 1. Second, Video-XL realizes high-fidelity compression of long videos, as it well maintains its performance throughout various compression ratios ($2\times$, $4\times$, $8\times$, $16\times$). Third, it also produces outstanding cost-effectiveness. Notably, it effectively handles 2048 frames with a single A100 GPU, while achieving 95% accuracy in the Needle-in-a-Haystack [53] evaluation.

## 2. Related Work

**Multimodal Large Language Models.** Building on the success of Large Language Models (LLMs), Multimodal Large Language Models (MLLMs) incorporate a visual encoder to extract visual features. A connector is then used to align these features to the same dimension as LLM tokens, enabling the LLM to handle visual information. Recent advancements in MLLMs [14, 57] have significantly improved performance in image understanding tasks. As pioneers, Flamingo [1] proposes to connect pre-trained vision-only and language-only models. The lightweight querying transformer is introduced in Blip2 [15] to bridge the gap between the image encoder and LLMs. LLaVA [23] proposes the visual instruction tuning using machine-generated instruction-following data.

**Video MLLMs.** With the excellent foundation of image MLLMs, many works [13, 16, 17, 26, 31, 32, 52] try to transfer the success of image understanding to the video understanding. However, the main difficulty with (long) video understanding is the sheer number of tokens, which often exceeds the context length of current LLMs. To handle this, MovieChat [41] and MA-LMM [10] use memory modules with long-term memory banks for accurate long video predictions. LLaMA-VID [18] reduces each frame to two tokens with context attention, while LongVLM [45] and Video-CCAM [5] focus on token merging and cross-attention modules for long context modeling. But these methods suffer from serious information loss, obstructing fine-grained comprehension. Unlike these methods,

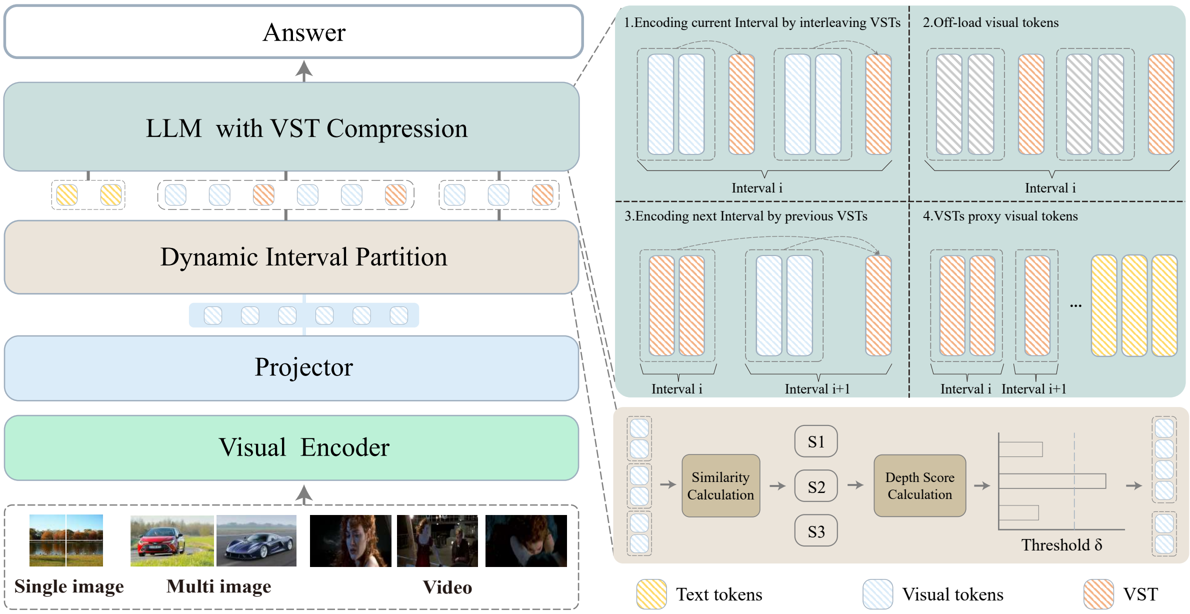


Figure 2. Overview of Video-XL. The input data (single-image, multi-image, or video) is encoded and projected as visual tokens. The visual tokens are dynamically split into intervals based on semantic consistency (measured by depth score). The visual information in each interval is compressed as VSTs' KVs, which enables MLLMs to perceive and understand longer inputs than its context window.

LWM [24] incrementally extends context using RingAttention [22], while LongVA [53] expands LLM context length directly. Other approaches enhance training methods [49] or improve LLM architecture [43]. However, substantial computational and memory costs for processing thousands of tokens in long videos remain unresolved.

# 3. Method

## 3.1. Overview

The architecture of Video-XL inherits the minimalism design of LLaVA series, which comprises a visual encoder, a visual-language projector, and an LLM backbone. First, the input image is encoded by a visual encoder, where we use CLIP-ViT-L [37] to carry out this operation. Second, the visual encoder's output embeddings are projected as visual tokens, where we leverage a two-layer MLP component with GELU activation. Third, the visual tokens, along with the text prompts, are fed into the LLM for conditioned text generation. Video-XL is featured for the introduction of VST module, which generates compressed KVs for lightweight and thus extended processing of long videos. In this section, we'll explain details about the compression mechanism (Section 3.2) and its training process (Section 3.3).

## 3.2. VST Compression

Unlike previous methods which reduce token count before LLM, we leverage the LLM itself to generate compact representations of videos. Given visual tokens $X$, we propose to compress the KVs of $X$ into the KVs of $C$, where $|C| \ll |X|$. This could substantially save the memory cost and thus allow the model to accommodate longer visual inputs within the constraints of the LLM's context length.

**Compression mechanism.** When encoding a token $x_i$ within the input $X$, the LLM needs to query for the entire KVs from $X_{<i}$. Consequently, it will consume significant GPU memory due to the storage of massive visual tokens, and it will be expensive to compute due to the quadratic complexity of self-attention. To avoid the huge cost from direct computation, we partition $X$ ($\{x_1, ..., x_n\}$) into shorter intervals $\{X_1, \ldots, X_i\}$ of sizes $\{w_1, ..., w_i\}$:

$$[x_1, \ldots, x_n] \xrightarrow{\text{Partition}} [X_1, \ldots, X_i], \tag{1}$$

where $\sum w_i = n$ and $|X_i| = w_i$. The length of each interval is within the constraint of LLM's context window. For each interval, we introduce a new special token, called Visual Summarization Token (**VST**): $\langle\text{vs}\rangle$, which prompts the LLM to compress the visual information into VST's KV, i.e. keys and values at every layer. We then determine a compression ratio $\alpha_i$ for each interval $X_i$. Based on this ratio, we uniformly interleave $k_i$ VSTs into the interval (denoted as $V_i = \{\langle\text{vs}\rangle_1^i, \ldots, \langle\text{vs}\rangle_{k_i}^i\}$), where $k_i = w_i/\alpha_i$. In other words, one VST is appended to every $\alpha_i$ visual token:

$$X_i \xrightarrow{\text{Interleave } V_i} X_i' = [x_1^i, \ldots, x_{\alpha_i}^i, \langle\text{vs}\rangle_1^i, \ \ldots \ , x_{w_i}^i, \langle\text{vs}\rangle_{k_i}^i]. \tag{2}$$

The LLM encodes each of these intervals one by one. Once the encoding of $X_i$ completes, the VSTs' KVs ($V_i$) are preserved as the compression of visual information, while the visual tokens' KVs ($X_i$) are off-loaded. When encoding

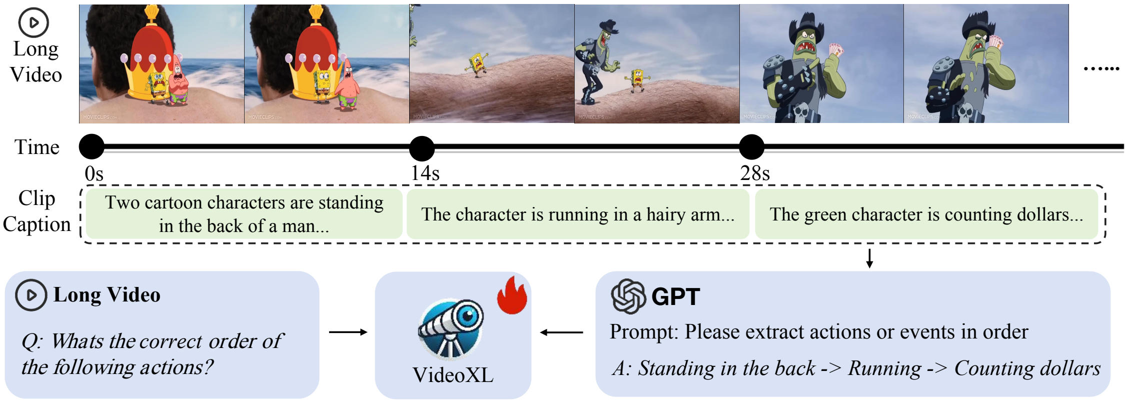

Long Video
Time
0s
14s
28s
Clip Caption
Two cartoon characters are standing in the back of a man...
The character is running in a hairy arm...
The green character is counting dollars...
Long Video
Q: Whats the correct order of the following actions?
VideoXL
GPT
Prompt: Please extract actions or events in order
A: Standing in the back -> Running -> Counting dollars

Figure 3. The pipeline of VICO generation. First, we generate short-clip captions for each few-second segments of the video. Then, we use GPT to extract key actions and events following the temporal order. Finally, the long video and QA pair are presented for model training.

the next interval $X_{i+1}$, the LLM will directly condition on the accumulated KVs from all preceding VSTs ($V_{\leq i}$) as a proxy to the original visual tokens $X_{\leq i}$.

**Dynamic compression strategy.** It's trivial to divide a long video into equal-sized intervals. However, the straightforward method is suboptimal considering that the information density is variant for different parts of the video. Particularly, some parts of the video exhibit high information density (*a.k.a.* information-dense), as they involve fast-changing visual semantics. While some other parts are of low information density (*a.k.a.* information-sparse), as they contain slow-paced visual semantics. The information-dense portions require fine-grained compression; in contrast, the information-sparse portions can do with coarse-grained compression. Because of this property, we design a dynamic compression method which customizes the compression granularity for each interval based on its information density. Inspired by VideoLLaMB [44], we employ CLIP to estimate the changing of visual semantic. Specifically, we make use of each frame's [cls] embedding to represent its global semantic. Thus, we can calculate the similarity scores $s_i$ for two neighboring frames ($i$-th and $i$+1-th). Based on this value, we can estimate the consistency of visual semantic using the depth score (defined in [44]):

$$d_i = max(s_1 \ \dots \ s_{i-1}) + max(s_{i+1} \ \dots \ s_n) - 2 \times s_i. \quad (3)$$

Intuitively, large depth scores mean sharp changes of visual semantic, which indicates potential semantic inconsistency. In our implementation, we introduce a threshold $\delta$, where the peak scores satisfying $d_i > \delta$ are chosen as the boundaries of video intervals. This enables the information-dense parts of the video to form small intervals for fine-grained compression, while the information-sparse portions to yield big intervals for coarse-grained compression.

## 3.3. Training

**Objective function.** Video-XL is trained by instruction tuning, where the model learns to optimize the generation likelihood of ground-truth response conditioned on the VST's compressed KVs and the task's instruction. Formally, the generation probability of the next token is formulated as:

$$\Pr(t_{i+1} \mid \underbrace{\langle \text{vs} \rangle_1^1, \dots, \langle \text{vs} \rangle_{k_j}^j}_{\text{compressed KVs}}, \underbrace{s_1, \dots, s_{\text{M}}}_{\text{instruction}}, \underbrace{t_1, \dots, t_i}_{\text{ground-truth}}; \boldsymbol{\Theta}),$$

where $\boldsymbol{\Theta}$ denotes the learnable parameters of the MLLM and VST module. We perform standard auto-regression to train the model, which minimizes the prediction loss for each of the tokens in ground-truth response.

**Curriculum learning.** The VST module is expected to support a wide range of compression ratios so as to flexibly handle videos of different lengths. By comparison, it's challenging to perform substantial compressions of long videos; however, it can be much easier to make small compressions. Because of this property, we propose to train Video-XL through curriculum learning. When the training process is started, we randomly sample small compression ratios, e.g., from (2, 4). Based on the sampled ratio, we apply VST compression to the input video and train the model via instruction tuning. In this stage, the VST module can acquire a preliminary capability in summarizing the visual information, which establishes a solid foundation to handle larger compressions. After the initial stage, we gradually improve the candidate compression ratios to 8, 12, and 16, thereby extending VST's capability in making larger compression.

**Composite Data curation**. Long-video instruction tuning data is very scarce in reality, which hinders the effective training of Video-XL. To mitigate this problem, we propose the composite curation of training data, where extra data resources are introduced to enhance the training effect. First, considering that understanding visual information in an image is foundational to video comprehension, we employ image captioning and QA data for augmentation. To facilitate knowledge transfer, we define a unified pipeline to transform all data into a uniform format. Specifically, we regard an arbitrary input data instance, whether it's a

| Model | Size | MLVU Dev | | MLVU Test | | VideoMME | | VNBench | VideoVista | LongVideo. | VideoChat. | MVBench |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | M-avg | G-avg | M-avg | G-avg | W/o sub | W sub | | | | | |
| **Proprietary Models** | | | | | | | | | | | | |
| GPT-4V [35] | - | 49.2 | 5.35 | 43.3 | 4.67 | 59.5 | 63.3 | 48.9 | - | 59.1 | **4.06** | **43.5** |
| GPT-4o [36] | - | **64.6** | **5.80** | **54.9** | **5.87** | 71.9 | 71.2 | 64.4 | **78.3** | **66.7** | - | - |
| Gemini-1.5-Pro [40] | - | - | - | - | - | **75.0** | **81.3** | **66.7** | - | 64.0 | - | - |
| **Open-source MLLMs** | | | | | | | | | | | | |
| VideoChat2 [17] | 7B | 47.9 | 3.99 | 35.1 | 3.99 | 39.5 | 43.8 | 12.4 | 61.6 | 39.3 | 2.98 | 62.3 |
| LLaMA-VID [18] | 7B | 33.2 | 4.22 | 17.2 | 3.43 | - | - | 10.8 | 56.9 | - | 2.89 | 41.4 |
| VideoLLaVA [20] | 7B | 47.3 | 3.84 | 30.7 | 3.68 | 39.9 | 41.6 | 12.4 | 56.6 | 39.1 | 2.84 | 43.0 |
| ST-LLM [26] | 7B | - | - | - | - | 37.9 | 42.3 | 22.7 | 49.3 | - | 3.15 | 54.9 |
| Shargpt4Video [3] | 7B | 46.4 | 3.77 | 33.8 | 3.63 | 39.9 | 43.6 | - | 53.6 | 39.7 | - | 51.2 |
| LLaVA-Next-Video [54] | 34B | - | - | - | - | 52.0 | 54.9 | 20.1 | 56.7 | 50.5 | **3.26** | - |
| PLLaVA [48] | 7B | - | - | - | - | - | - | - | 60.4 | 40.2 | 3.12 | 46.6 |
| LongVA† [53] | 7B | 56.3 | 4.33 | 41.1 | 3.91 | 52.6 | 54.3 | 41.5 | 67.4 | 47.8 | - | - |
| VideoLLaMA2† [4] | 8x7B | - | - | - | - | 47.9 | 49.7 | 24.9 | 60.5 | 36.0 | **3.26** | 53.9 |
| Video-CCAM† [5] | 9B | 58.5 | 3.98 | **42.9** | 3.57 | 50.3 | 52.4 | 35.6 | 69.0 | 43.1 | - | **64.6** |
| Long-LLaVA [43] | 13B | - | - | - | - | 51.9 | - | 52.1 | - | - | - | - |
| **Video-XL** | 7B | **64.9** | **4.50** | **45.5** | **4.21** | **55.5** | **61.0** | **61.6** | **70.6** | **50.7** | 3.17 | 55.3 |

Table 1. Experimental results on mainstream video benchmarks. "LongVideo." and "VideoChat." refer to LongVideoBench and VideoChat-GPT Bench, respectively. † indicates that the results on VNBench and LongVideoBench were reproduced using their official weights.

single-image, a multi-image, or a video, as a super image. We then divide the super image into multiple patches, each one in a resolution of 336 × 336. For each patch, we make use of CLIP to encode it as $M$ visual tokens ($M = 144$ in our implementation). In our work, the following datasets are collected: Bunny [11], Sharegpt-4o [12] (57k), and MMDU [30] (20k). These datasets are combined with our video data, which contains NExT-QA [47] (32k), Sharegpt-4o [12] (2K), CinePile [39] (10k), VCG [33] (25k) and in-house video captions with GPT-4V (11k).

Second, understanding long videos also relies on precise and comprehensive utilization of proper information from the input. Thus, we additionally curate another synthetic dataset, called VIsual Clue Order (VICO), to strengthen this fundamental capability. VICO contains 20k QA pairs, each one is associated with a video of 3 minutes on average. The videos are sourced from CinePile [39], which covers diverse genres, like movies, documentaries, games, sports, etc. As shown in Figure 3, each long video is segmented into 14-second clips. For each clip, we use the VILA-1.5 [21] to generate detailed descriptions. Based on these captions, we leverage GPT-4 to extract the key events and arrange them in a temporal order. VICO requires models to identify and reason about key information in a long video, thereby enhancing their long video comprehension capabilities.

# 4. Experiment

## 4.1. Implementation

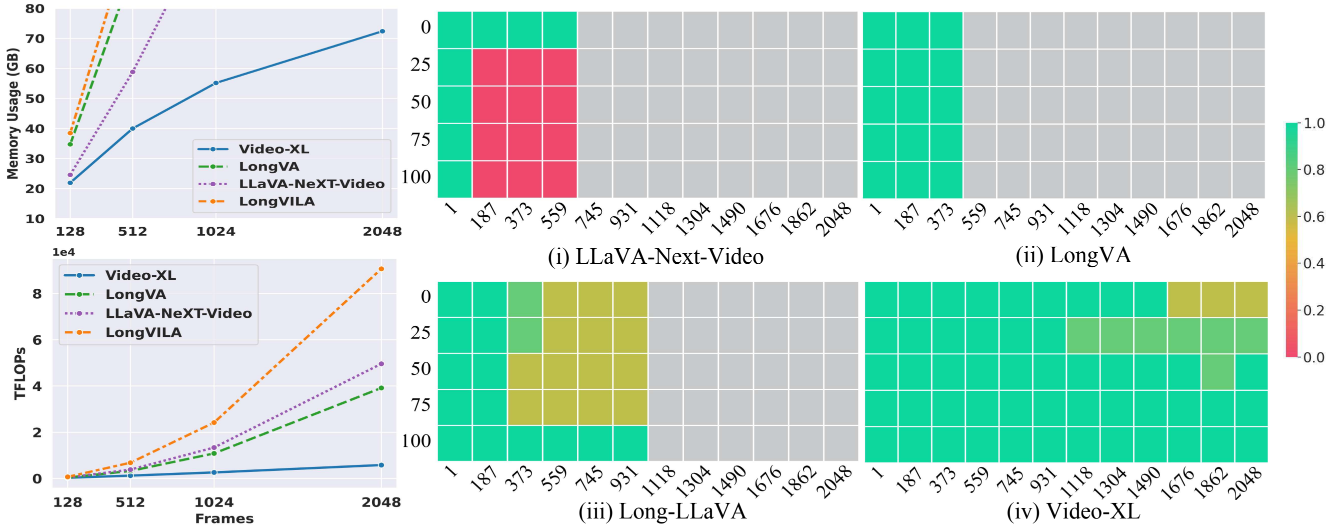
Video-XL is trained on Qwen-2-7B [50]. We employ a two-stage strategy to train Video-XL. During pre-training, we use the Laion-2M dataset [11] to optimize the projector, where visual embeddings from a CLIP-ViT-L [37] based vision encoder are aligned with the text embeddings of LLM. During fine-tuning, we apply visual instruction tuning to optimize the parameters of vision encoder, projector and LLM. The batch sizes for pre-training and finetuning are 8 and 1, while the learning rate is 5e-5 for pre-training and 1e-5 for fine-tuning, with linear decay and no warmup. All experiments are conducted on one 8×A800-80GB machine.

## 4.2. Benchmarks

We empirically evaluate the effectiveness of Video-XL based on several popular long video understanding benchmarks. 1. MLVU [56], a comprehensive benchmark which is made up of both multiple choice and generation tasks. 2. Video-MME [7], another extensive benchmark covering videos of diverse genres and lengths (short, medium, and long). 3. VNBench [55], a synthetic benchmark focused on assessing models' ability to handle long-video tasks, such as retrieval, ordering, and counting. 4. LongVideoBench [46], a benchmark designed for tasks that require precise retrieval and reasoning over detailed multi-modal information within extended inputs. 5. VideoVista [19], which aims to evaluate a model's long-context reasoning ability over videos of varying durations. In addition to the above long video evaluation, we also make extension for two short video question answering benchmarks: the VideoChatGPT Benchmark [32] and MVBench [17].

## 4.3. Main Results

We present the performance of Video-XL on popular long-video benchmarks in Table 1. Our results show that Video-XL consistently achieves strong performances across these experiments. Notably, it outperforms the existing methods on both Dev and Test tasks of MLVU. It even surpasses GPT-4o on the Dev tasks despite having only 7B parameters. For Video-MME, Video-XL achieves accuracies of



(i) LLaVA-Next-Video

(ii) LongVA

(iii) Long-LLaVA

(iv) Video-XL

Figure 4. (Left) Comparison of the memory usage and the forward FLOPs of different models. (Right) Results on the Needle-in-a-haystack evaluation within a single A100 80GB GPU. The x-axis represents the total number of frames in the video haystack. The y-axis shows the position where the needle image is located. Gray grids mean "OOM'.

55.5% and 61.0% for the 'without' and 'with-subtitle' settings, respectively, which yields competitive results compared to the state-of-the-art models on this benchmark. For VNBench, Video-XL sets the top performance among the open-source models, leading the previous best model by nearly 10% in accuracy. Once again, it surpasses GPT-4V and achieves a comparable performance to GPT-4o in this evaluation. While for VideoVista, Video-XL ranks the first place among all open-source MLLMs, trailing only behind GPT-4o and Gemini-1.5 [40]. Video-XL also brings forth the highest performance among all open-source models with no more than the 7B parameters on the Dev task of LongVideoBench. Last but not least, although designed primarily for long video understanding tasks, Video-XL excels in short video tasks as well, yielding competitive results on both VideoChatGPT and MVBench benchmarks.

## 4.4. Extra-Long Evaluation

To explore Video-XL's ability to process extra-long video inputs, we further conduct the Needle-In-The-Haystack evaluation [53] based on an A100-80GB GPU. We consider two types of baselines in our evaluation: 1) LLaVA-NexT-Video and LongLLaVA, which rely on position extrapolation methods to make extension for longer inputs, and 2) LongVA, which fine-tunes the MLLM to handle longer inputs. As shown in Figure 4, Video-XL exhibits notable advantages over the baselines. First, Video-XL is able to cover much longer video inputs. However, neither LLaVA-NexT-Video nor LongLLaVA can support more than 1000 frames due to the constraint of computation cost, while LongVA is only fine-tuned to support less than 400 frames. Second, Video-XL well maintains a superior performance. It preserves 100% accuracy within 128 frames, which is the maximum length of its fine-tuning data; meanwhile, it achieves nearly 95% accuracy when dealing with longer inputs. In contrast, LLaVA-NexT-Video and LongLLaVA suffer from inferior retrieval performance, while LongVA can only handle inputs within its fine-tuned length.

| Model | MLVU | VideoMME | MME | MMB |
|---|---|---|---|---|
| Pooling | 33.7 | 41.0 | 1405.5 | 62.3 |
| Q-Former | 35.1 | 42.1 | 1410.2 | 61.9 |
| LLaMA-VID | 35.5 | 45.7 | 1421.2 | 64.3 |
| LLaMA-Adapter | 35.3 | 42.2 | 1418.3 | 65.5 |
| C-Abstractor | 37.1 | 46.3 | 1440.2 | 65.1 |
| Video-XL | 41.4 | 52.0 | 1510.2 | 70.9 |
| Upper-bound | 41.8 | 52.6 | 1533.7 | 71.6 |

Table 2. Comparison of compression techniques. All methods are implemented in the same setting and with 16× compression.

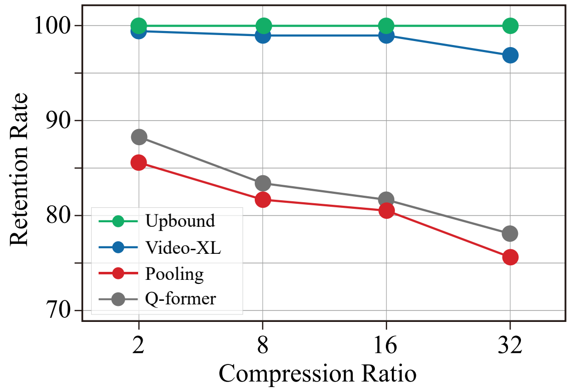


Figure 5. MLVU performance with variant compression ratios. The retention rate is calculated as the ratio to the upper-bound.

| Train | Test | MLVU | VideoMME | MME | MMB |
|---|---|---|---|---|---|
| ✗ | ✗ | 39.8 | 50.9 | 1460.6 | 70.9 |
| ✗ | ✓ | 39.6 | 50.8 | 1455.0 | 70.8 |
| ✓ | ✗ | 41.5 | 52.0 | 1515.5 | 71.2 |
| ✓ | ✓ | **41.6** | **52.3** | **1520.0** | **71.3** |

Table 3. Evaluation of dynamic compression strategy.

## 4.5. Inference Efficiency

We further evaluate the inference efficiency of Video-XL in comparison to three baselines: LongVA, LLaVA-NeXT-Video, and LongVILA. As shown in Figure 4 (left), Video-XL significantly reduces memory usage thanks to its compression of visual information. The substantial reduction in memory consumption allows it to process over 2048 frames with a single A100-80GB GPU. In addition, Video-XL also results in much smaller TFLOPs than the baseline methods, as it eliminates the need for direct self-attention over long input sequences.

## 4.6. Ablation Studies

We conducted extensive ablation studies to explore Video-XL's effectiveness regarding its compression mechanism, training method, and data curation.

**Compression mechanism**. First, we compare Video-XL with previous common pre-compression methods, including average pooling, Q-Former [15], LLaMA-VID [18], LLaMA-Adapter [8], and C-Abstractor [2]. For a fair comparison, these methods are implemented based on their official codes, but switched to the same architecture and training data as our method. With a uniform compression ratio of 16×, we report the results on two long video benchmarks, MLVU-test and VideoMME, as well as two popular VQA benchmarks, MME [6] and MMB [27]. As shown in Table 2, Video-XL significantly outperforms previous methods across all benchmarks, particularly on long video benchmarks, which require fine-grained detail understanding and long-term relational reasoning. Additionally, Video-XL achieves high-fidelity compression with minimal performance loss, even at compression ratios as high as 16×. Moreover, we further explore the performance under various compression ratios (2×, 8×,16×, 32×) in Figure 5. Note that 32× is directly tested without fine-tuning. In these experiments, Video-XL maintains a close performance as the upper-bound, outperforming the baselines by a large margin. Meanwhile, it also effectively preserves its performance for the unseen compression ratio (32×), suggesting the generality of the proposed method.

| Settings | MLVU | VideoMME | MME | MMB |
|---|---|---|---|---|
| w/o random compre. | 40.5 | 51.0 | 1500.4 | 70.3 |
| w/o curriculum learn. | 41.1 | 51.6 | 1512.4 | 71.0 |
| Ours | **41.6** | **52.3** | **1520.0** | **71.3** |

Table 4. Evaluation of curriculum learning.

| Video | Single Image | Multi Image | TR | NQA | AO | Avg |
|---|---|---|---|---|---|---|
| 100k | - | - | 73.4 | 64.5 | 53.6 | 63.8 |
| 100k | 350k | - | 77.5 | 66.9 | 54.0 | 66.1 |
| 100k | 700k | - | 80.6 | 70.0 | 54.1 | 68.2 |
| 100k | 1M | - | 81.3 | 69.8 | 53.8 | 68.3 |
| 100k | 700k | 20k | **82.0** | **70.3** | 55.3 | **69.5** |
| 100k | 700k | 40k | 82.1 | 70.1 | **55.4** | 69.2 |

Table 5. Analysis of training effect from different data.

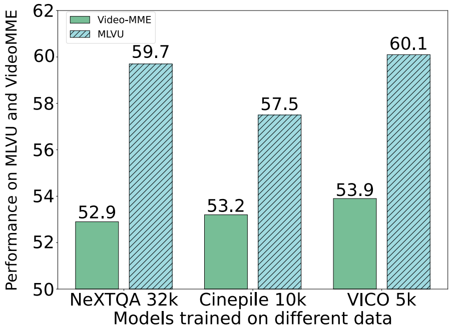
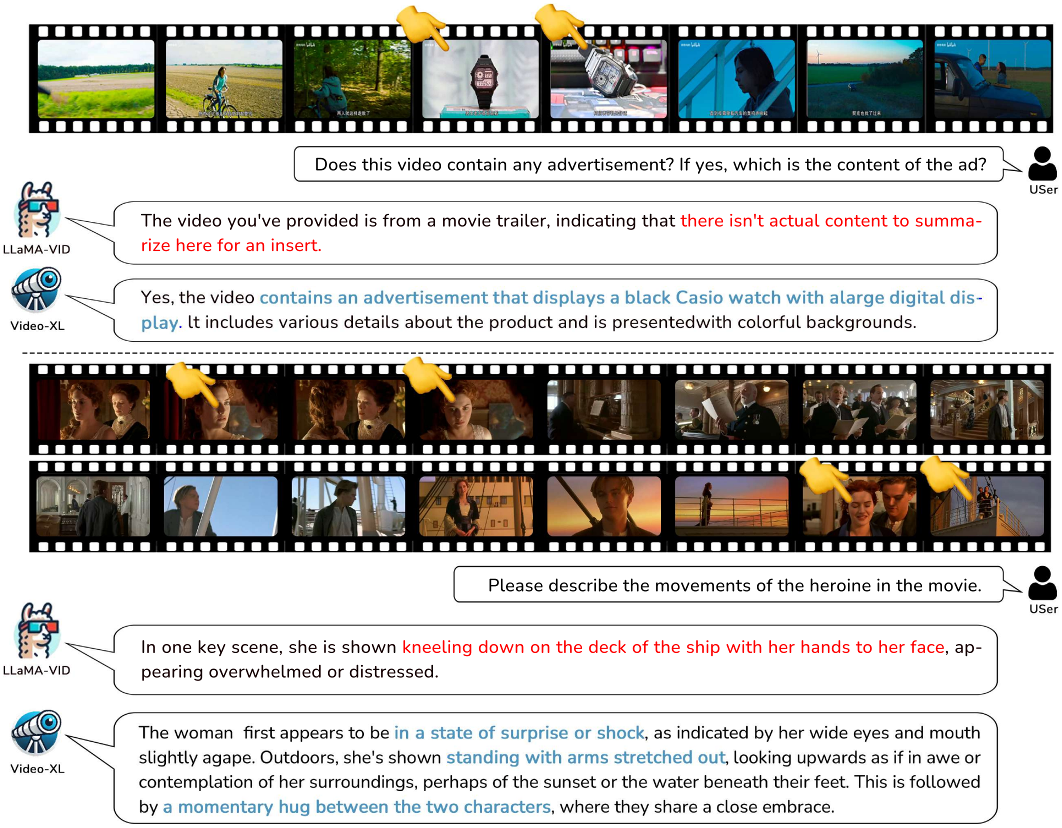


Figure 6. Analysis of training effect from VICO.

**Dynamic compression strategy**. Second, we analyze the effect of dynamic compression strategy. In this experiment, we compare the settings where dynamic compression is disabled, or individually enabled for training and testing. If dynamic compression is disabled, we perform fixed compression based on an interval of 1440 tokens. From the experiment results in Table 3, we can validate the effectiveness of our method, as it leads to a substantial improvement over the dynamic-compression disabled baselines. Meanwhile, we can also observe that it's necessary to conduct dynamic compression during training, as no improvements are obtained if it's only enabled for testing stage. More discussions on this issue are provided in the supplementary.

**Curriculum learning**. To assess the effectiveness of curriculum learning, we re-train the model with two settings: 1. using randomized compression ratios within 16×, i.e., w/o curriculum, 2. using a fixed compression ratio 16×, i.e. w/o random (16× is the compression ratio used for testing). As shown in Table 4, the our methods substantially improves upon the two baselines, indicating the necessity to learn progressively from small compression ratios.

**Composite data curation**. To explore the effect from different data sources (video, single-image, multi-image), we make fine-grained analysis based on three types of tasks from MLVU: 1. Topic Reasoning (TR): for holistic understanding capability, 2. Needle QA (NQA): for single-detail understanding capability, 3. Action Order (AO): for multi-detail understanding capability. As shown in Table 5, the increasing of image data effectively enhances the model's holistic (TR) and single-detail (NQA) capability, however, it contributes little to multi-detail (AO) capability (from 1st row to 3rd row). Meanwhile, once sufficient image data is



Does this video contain any advertisement? If yes, which is the content of the ad?

The video you've provided is from a movie trailer, indicating that there isn't actual content to summarize here for an insert.

Yes, the video contains an advertisement that displays a black Casio watch with alarge digital display. lt includes various details about the product and is presentedwith colorful backgrounds.

Please describe the movements of the heroine in the movie.

In one key scene, she is shown kneeling down on the deck of the ship with her hands to her face, appearing overwhelmed or distressed.

The woman first appears to be in a state of surprise or shock, as indicated by her wide eyes and mouth slightly agape. Outdoors, she's shown standing with arms stretched out, looking upwards as if in awe or contemplation of her surroundings, perhaps of the sunset or the water beneath their feet. This is followed by a momentary hug between the two characters, where they share a close embrace.

Figure 7. Quantitative evaluation of Video-XL in two tasks.

presented, the additional benefit becomes marginal (as reflected from the 3rd row to the 4th row). Finally, the introduction of multi-image data significantly improves the model's multi-detail capability, as it enables the model to learn fine-grained relationships within long inputs. The above observations indicate that different data sources are complementary to each other, which jointly contribute to the superior performance of Video-XL.

To further analyze the effect of VICO dataset, we re-train the model using three video instruction-tuning datasets: (a) NeXTQA 32k, (b) CinePile 10k, and (c) VICO 5k. The corresponding results on Video-MME and MLVU are shown in Figure 6. Although VICO is the smallest of all datasets, it substantially outperforms the other two datasets which contain more training samples (5k vs. 32k and 10k), demonstrating its value to establish the long-video understanding capability for MLLMs. We also discuss the effect from scaling up VICO in our supplementary material.

### 4.7. Qualitative Evaluation

We leverage qualitative evaluation for an intuitive analysis of Video-XL. In this experiment, we make comparison with LLaMA-VID [18] based on extra-long videos (over 30 minutes). As shown in Figure 7, Video-XL accurately locates the inserted advertisement and presents its details; in contrast, LLaMA-VID struggles to comprehend the video and make judgment on whether an advertisement is inserted. Additionally, Video-XL effectively summarizes the plots about the heroine in the long video, whereas LLaMA-VID only returns a short and inaccurate description. We include more qualitative analysis in our supplementary material.

## 5. Conclusion

In this paper, we introduce Video-XL, which enables the processing of long videos on top of the compressed representations generated by our visual summarization token (VST). To better retain the visual information, we conduct dynamic compression based on the information density of the video. Additionally, to optimize the training effect, we design a curriculum learning method, allowing for progressive learning of different compression ratios. We also propose composite data curation, which jointly utilizes multiple data sources to improve the model's performance. The effectiveness of Video-XL is empirically verified, as it achieves superior performance across popular long-video benchmarks and delivers competitive compression quality and cost-effectiveness in our experiments.

# References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 2

[2] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, pages 13817–13827, 2024. 7

[3] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *NeurIPS*, 37:19472–19495, 2024. 5

[4] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 5

[5] Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024. 2, 5

[6] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 7

[7] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 2, 5

[8] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 7

[9] Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. *arXiv preprint arXiv:2411.14794*, 2024. 1

[10] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. *arXiv preprint arXiv:2404.05726*, 2024. 1, 2

[11] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024. 5

[12] https://sharegpt4o.github.io/. sharegpt4o. 5

[13] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, pages 13700–13710, 2024. 2

[14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[15] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ICML*, 2023. 1, 2, 7

[16] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 1, 2

[17] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, pages 22195–22206, 2024. 2, 5

[18] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 1, 2, 5, 7, 8

[19] Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. Videovista: A versatile benchmark for video understanding and reasoning. *arXiv preprint arXiv:2406.11303*, 2024. 5

[20] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 5

[21] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, pages 26689–26699, 2024. 5

[22] Hao Liu and Pieter Abbeel. Blockwise parallel transformers for large context models. *NeurIPS*, 36, 2024. 3

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2

[24] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*, 2024. 3

[25] Liu Liu, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yufei Ding, and Yuan Xie. Dynamic sparse attention for scalable transformer acceleration. *IEEE Transactions on Computers*, 71(12):3165–3178, 2022. 2

[26] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. *arXiv:2404.00308*, 2024. 2, 5

[27] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023. 7

[28] Yexin Liu, Zhengyang Liang, Yueze Wang, Muyang He, Jian Li, and Bo Zhao. Seeing clearly, answering incorrectly: A multimodal robustness benchmark for evaluating mllms on leading questions. *arXiv preprint arXiv:2406.10638*, 2024. 1

[29] Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, et al. Deja vu: Contextual spar-

sity for efficient llms at inference time. In *ICML*, pages 22137–22176. PMLR, 2023. 2

[30] Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, et al. Mmdu: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. *arXiv preprint arXiv:2406.11833*, 2024. 5

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

[32] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 1, 2, 5

[33] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arxiv*, 2024. 5

[34] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022. 1

[35] OpenAI. Gpt-4 technical report, 2023. 1, 5

[36] OpenAI. Gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024. 5

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 3, 5

[38] Jiayuan Rao, Haoning Wu, Hao Jiang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards universal soccer video understanding. *arXiv preprint arXiv:2412.01820*, 2024. 1

[39] Ruchit Rawal, Khalid Saifullah, Ronen Basri, David Jacobs, Gowthami Somepalli, and Tom Goldstein. Cinepile: A long video question answering dataset and benchmark. *arXiv preprint arXiv:2405.08813*, 2024. 5

[40] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 5, 6

[41] 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 preprint arXiv:2307.16449*, 2023. 1, 2

[42] 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 preprint arXiv:2302.13971*, 2023. 1

[43] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024. 3, 5

[44] Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long-context video understanding with recurrent memory bridges. *arXiv preprint arXiv:2409.01071*, 2024. 4

[45] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*, 2024. 1, 2

[46] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *NeurIPS*, 37:28828–28857, 2024. 2, 5

[47] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, pages 9777–9786, 2021. 5

[48] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024. 5

[49] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 3

[50] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 5

[51] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024. 1

[52] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 1, 2

[53] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 2, 3, 5, 6, 1

[54] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 5

[55] Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic framework for benchmarking video mllms. *arXiv preprint arXiv:2406.09367*, 2024. 5

[56] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 2, 5

[57] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2