

# Quickviewer: An LMM for Efficient Video Understanding via Reinforced Compression of Video Cubes

Ji Qi<sup>◇</sup>, Yuan Yao<sup>†\*</sup>, Yushi Bai<sup>◇</sup>, Bin Xu<sup>◇\*</sup>, Juanzi Li<sup>◇</sup>, Zhiyuan Liu<sup>◇</sup>, Tat-Seng Chua<sup>†</sup>

<sup>◇</sup>Tsinghua University <sup>†</sup>National University of Singapore  
qijithu1@gmail.com, yaoyuanthu@gmail.com

<https://quickviewer.github.io>

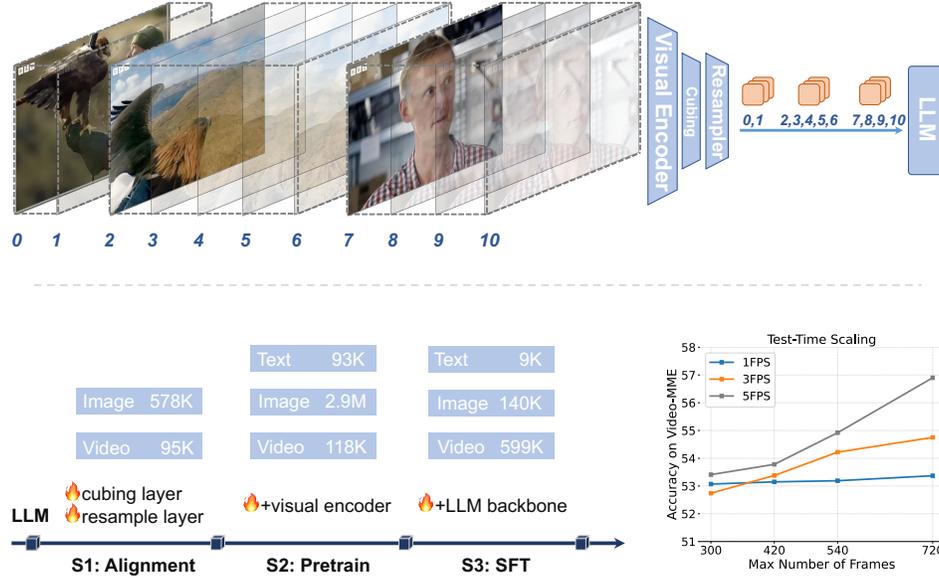


Figure 1: Quickviewer involves a cubing network that partitions a video into nonuniform cubes, followed by a 3D resampler to gather a fixed number of visual tokens per cube. This efficiency enables **Large Receptive Field** (420 frames) with **High Compression Rate** (64 $\times$ ) during all training stages, and scaling laws on extended frames in inference.

## Abstract

Large Multimodal Models (LMMs) uniformly perceive video frames, creating computational inefficiency for videos with inherently varying temporal information density. This paper presents **Quickviewer**, an LMM with a new perceiving paradigm that partitions a video of nonuniform density into varying cubes using Gumbel Softmax, followed by a unified resampling for each cube to achieve efficient video understanding. This simple and intuitive approach dynamically compresses video online based on its temporal density, significantly reducing spatiotemporal redundancy (overall 45 $\times$  compression rate), while enabling efficient training with large receptive fields. We train the model from a language backbone through three progressive stages, each incorporating lengthy videos on average of 420s/1fps thanks to the perceiving efficiency. With only 0.8M total video-text samples for training, our model outperforms the direct baseline employing a fixed partitioning strategy by a maximum of 8.72 in accuracy, demonstrating the effectiveness in performance. On Video-MME, Quickviewer achieves SOTA under modest sequence lengths using just up to 5% of tokens per frame required by baselines. With this paradigm, scaling up the number of input frames reveals a clear power law of the model capabilities. It is also empirically verified that the segments generated by the cubing network can help for analyzing continuous events in videos.

\*Corresponding author: yaoyuanthu@gmail.com

# 1 Introduction

Large Multimodal Models (LMMs) (Deepmind, 2024; OpenAI, 2024; Bai et al., 2025) have shown promising progress in video understanding, paving the way for general intelligence in physical world. These models build on Large Language Models (LLMs) and are trained in stages with large-scale image and video data, encoding video frames in the same manner as images before feeding them into the LLM for inference. At the core of these models is the efficient perception of input videos, which is crucial in tackling the persistent contradiction between the **temporal redundancy** of video streams (Buckler et al., 2018; Wenger, 1997) and the **computational efficiency** of LMMs with long context (Fu, 2024).

Extensive studies have been striving to develop LMMs for solving this fundamental issue. Building on devise of frame sampling methods, trailblazing efforts typically involve dedicated token merging strategies (Bai et al., 2025; Wang et al., 2025; Shen et al., 2024; Zohar et al., 2024; Li et al., 2024c; Zhang et al., 2025) and adapted parallel training infrastructures (Zhang et al., 2024a; Chen et al., 2024c; Shen et al., 2025). However, the arbitrary frames sampling and tokens merging introduces inevitable information loss, while marginal compression limits the number of frames in large-scale pre-training.

The velocity of content change in videos is inherently nonuniform, suggesting that the density varies across different temporal cubes. For example in Figure 1, the initial short period features rapidly changing scenes of a researcher attaching a camera to an eagle’s back, followed by an extended sequence of stable footage from the camera, and a largely static interview. Inspired by the way that humans adjust their perception speed based on content changes, this paper explores how LMMs can perform video understanding on the nonuniform cubes to achieve dynamic compression, and significantly reduce the spatiotemporal complexity and enhance the efficiency. For practical scenarios where videos typically originate from lengthy offline recordings or online video streams, we thereby aim for the model to **(1) learn from unlabeled data**, perform **(2) online cube partitioning**, and establish a **(3) unified perception paradigm** for images and videos.

We present Quicksviewer, an LMM that perceives nonuniform video cubes for efficient video understanding. Given a video passed from a visual encoder, a small cubing network first partitions it into nonuniform cubes based on the momentum of semantic feature differences between frames, a process that can be conducted online in streaming scenarios. Next, a unified resampling is employed to the cubes to gather a fixed number of tokens for adaptive compression. Finally, these visual tokens, along with absolute timestamps, are fed into the LLM for inference. We integrate the learning of the cubing network into the end-to-end training of the LMM using the Gumbel Softmax (Jang et al., 2016; Herrmann et al., 2020) method with an improved noise annealing mechanism. This reinforced approach not only enables efficient learning on videos without boundary labels but also insures sufficient sampling over the cubes distribution with continuous gradient during training. The nonuniform perception paradigm, which is solely driven by the properties of input video, together with the subsequent resampling enables an efficient video encoding with  $45\times$  compression rate, large temporal receptive field of 420 frames for pre-training, and a consistent representation for both images and videos.

We train our models starting from the LLM backbones through three progressive stages, each incorporating lengthy videos averaging 420s/1fps by benefiting from the efficient perception mechanism. The resulting model, which we coined as Quicksviewer, is an efficient LMM capable of understanding single/multi-images and long videos. We also find that our network is efficient in learning. With only 0.8M video-text samples in total for training, our model achieves SOTA performance on Video-MME (Fu et al., 2024) using just up to 5% of the tokens per frame required by baselines. In addition, to facilitate training on ultra-long videos (*e.g.*, over 1hour), we developed a training infra supporting dynamic changes of sequences lengths based on an existing effort (Chen et al., 2024c), to further facilitate potential explorations in future.

We evaluate Quicksviewer on various video understanding benchmarks, ranging the duration from 16 seconds to 1 hour. Results show that our model outperforms the direct baseline employing a fixed partitioning strategy by a maximum of 8.72 in accuracy, suggesting the utility of the nonuniform perception. We further analyze the cubes partitioning in videos, which demonstrates the emergence of "Visual Lag" phenomenon when the model perceives videos phase-by-phase. We also conduct

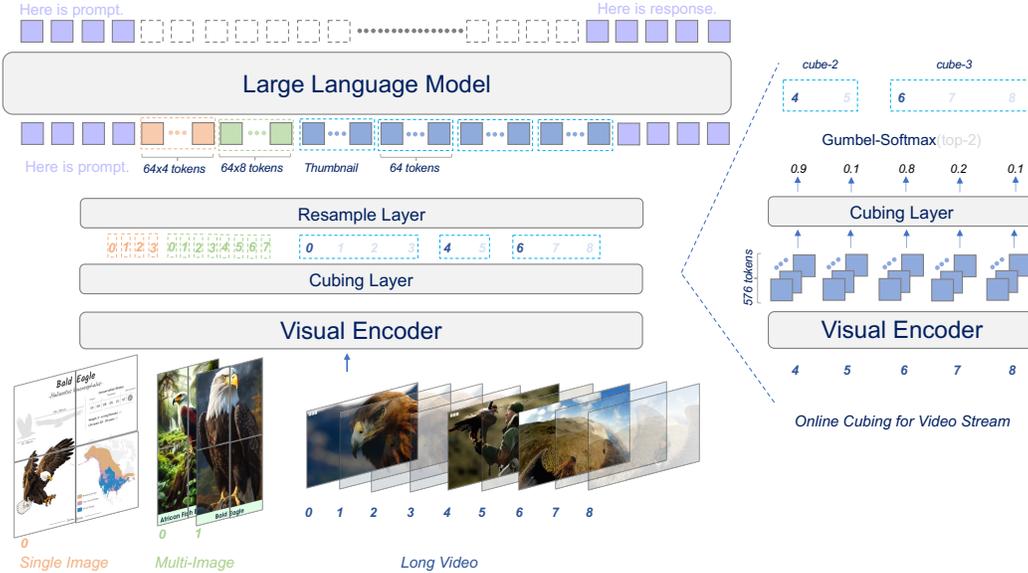


Figure 2: Left: The network architecture of Quicksviewer, that performs unified understanding of videos and images through visual tokens from cascaded modules. Right: The cubing network, that partitions an online video into nonuniform cubes based on Gumbel Softmax.

extensive ablation studies to show the effectiveness of presented components, including the cubing approach, 3D positional encoding, loss penalty, and the annealing strategy.

## 2 Approach

The overall architecture of the presented model is shown in Figure 2. We build an efficient LMM that can receive both images and videos as input, where the video is perceived based on nonuniform cubes partitioned through a small cubing network. The model consists of four basic components: (1) a visual encoder  $f_v(\cdot)$  that encode image slices or video frames into visual tokens, (2) a cubing network  $f_q(\cdot)$  that partitions the video frames into  $N_Q$  cubes, (3) a resampler  $f_r(\cdot)$  which compress each slice or cube into a fixed number of tokens, and (4) an LLM  $f_i(\cdot)$  which accept concatenation of visual tokens and user prompt for response generation. Note that we introduce **FPQ**, the average number of frames per cube, which regulates the perception granularity and enables adaptive number of cubes  $N_Q$  according to video duration.

### 2.1 Cubing using Gumbel Softmax

**Visual Encoding** Given an input video to our model, we first uniformly sample  $N_F$  frames with a fixed FPS to form  $[F_i]_{i=1, \dots, N_F}$ . And then, each frame is firstly encoded using the visual encoder  $f_v$  to obtain  $N_1$  visual tokens  $\mathbf{F}_j = [\mathbf{v}_i]_{i=1, \dots, N_1}$ , where  $\mathbf{v}_i \in \mathbb{R}^d$  is the token representation. The large number of visual tokens provides fine-grained semantic representations for each frame, while the concatenation of all tokens (*i.e.*,  $N_F \times N_1$ ) across frames results in an unacceptable sequence length.

**Cubing** It is natural to leverage the semantic difference of features to find keyframes. For compatible with online streaming scenarios and also considering the event-level long-term changes, we utilize the momentum accumulated from the representations of previous frames to track the semantic changes. Given the visual tokens of frames, the cubing network calculate the  $i^{th}$  momentum representation as:

$$\Delta_i = \alpha(\mathbf{F}_i - \mathbf{F}_{i-1}) + (1 - \alpha)\Delta_{i-1} \quad (1)$$

where  $\Delta_i = [\delta_i]_{j=1, \dots, N_1} \in \mathbb{R}^{N_1 \times d}$  captures the accumulated semantic changes discounted by factor  $\alpha \in [0, 1]$ . A 2-layers MLP with LayerNorm (Ba et al., 2016) is then applied to the mean of visual tokens momentums to quantify the significance of the frame:

$$\mathbf{z}_i = \text{MLP}_{\times 2}(\text{LayerNorm}(\frac{1}{N_1} \sum_{j=1}^{N_1} \delta_j)), \quad \mathbf{z}_i \in \mathbb{R}^2 \quad (2)$$

where the 2-dimensional vector  $\mathbf{z} = [w_0, w_1]$  forms a gate, and the sigmoid of their difference directly reflects the keyframe probability  $p = \frac{1}{1 + \exp(w_0 + w_1)}$  (Herrmann et al., 2020). We next apply the *top-k* operation on the first dimension of  $[\mathbf{z}_i]_{i=2, \dots, N_F}$  vectors to obtain the indices of  $N_Q - 1$  largest values  $[l_i]_{i=1, \dots, N_Q-1}$ , which are selected as keyframes. The keyframe  $F_{l_i}$ , along with its subsequent consecutive non-keyframes, forms the cube  $Q_{i+1} = [F_{l_i}, \dots, F_{l_{i+1}}]$ . Note that the first sampled frame consistently serves as the keyframe for deriving the first cube  $Q_1$ .

**Sampling for Training** During training, we expect the model to perform sufficient sampling exploration while ensuring gradient continuity. We achieve this using Gumbel Softmax with the Straight-Through trick (Gumbel, 1954; Jang et al., 2016):

$$\mathbf{z}_i = \text{softmax}(\mathbf{z}_i - \log(-\log(\epsilon)/\tau)), \quad \epsilon \sim U(0, 1) \quad (3)$$

where the log term approximates the sampling process and regulates the degree of exploration.

In experiments, we found that persistent exploration prevents the model from establishing a stable cubing paradigm for subsequent reasoning, leading to sustained loss oscillations. We propose to add an learning rate  $\eta$  before the Gumbel noise  $\eta \log(-\log(\epsilon)/\tau)$ , which is **annealed** from  $\eta_0 = 1.0$  to  $\eta_T = 0.001$  during training using a cosine scheduler.

## 2.2 Resampling with 3D Positional Encoding

Based on the partitions from cubing network, a unified 3D resampler is adopted to compress each cube of arbitrary length into a fixed number of  $N_2$  tokens.

**Resampling Video Cubes** We employ the same resampler architecture as (Yao et al., 2024) to compress each cube into a fixed number of dense tokens. We extend the original 2D positional encoding by incorporating a temporal dimension to form 3D position encoding. As a result, each video token is assigned three positional coordinates  $(x, y, z)$ , representing time, width, and height. We then unfold each cube into tokens sequence along the frame dimension. After adding 3D positional embeddings, a unified resampling is performed to obtain  $N_2 = 64$  visual tokens for each cube. For images, we first adopt the AnyRes (Liu et al., 2024) to divide high resolutional images into slices, and apply the same resampling to each slice to obtain visual tokens and finally concatenate the tokens.

**Resampling for Video Thumbnail** Using cubes quantized from the cubing network for response tokens generation introduces a fundamental problem: *How does the NTP training objective optimize the boundary prediction of the cubing network?* We introduce video thumbnail to resolve this problem meanwhile provide effective global representation. Specifically, we first (1) multiply the 0-1 discretized first dimension of vectors  $[\mathbf{z}_i]_{i=1, \dots, N_F}$  with their corresponding frame representations  $[\mathbf{F}_i]_{i=1, \dots, N_F}$ , then (2) average across the frame dimension to obtain  $N_1$  visual tokens. A further resampling is performed to get final thumbnail representation containing  $N_2$  tokens. This simple approach allows gradients to be directly propagated back to the cubing boundaries. The final representation of a video is a concatenation of the representations of the thumbnail and cubes.

## 2.3 LLM Inference with Auxiliary Loss

Following resampling of nonuniform video cubes, the tokens of each cube span varying temporal windows. We prepend each cube with an absolute timestamp as a float number in 0.01-second units, enabling explicit temporal awareness. We also enclose the video, thumbnail, and image tokens with their corresponding special tokens to enable explicit content differentiation.

During training, we observed that excessively large values of  $\mathbf{z}$  cause overly high gradients, impairing convergence. To address this, we introduce an auxiliary  $L_2$  norm loss with  $\beta = 0.001$  penalty weight on them to constrain its values within a reasonable range.

### 3 Training Process

We train our models with three progressive stages starting from LLM backbones, each stage incorporating lengthy videos on average of 420s by benefiting from the efficient perception approach.

**Stage-1: Multimodal Alignment** We utilize both interleaved and captioning image-text corpuses, and video-text captioning corpus to train our models, establishing fundamental alignment between visual encoder and LLM backbones with in-context learning capabilities. We sample a subset of 20K sequences from OBELICS (Laurençon et al., 2023), with each containing more than two interleaved pairs. We utilize LCS (Li et al., 2024a), a re-captioned dataset consisting of 558K detailed descriptions from the CC3M (Sharma et al., 2018). The video-text training data incorporates a sampled subset of 87K captioning pairs from FineVideo (Farré et al., 2024) and 8K captioning pairs from ANetCaptions (Krishna et al., 2017). We train parameters of the cubing network and resampler while keeping all other parameters frozen to establish a stable projection. The models are trained for 1 epoch with a  $lr$  that warms up to  $1e^{-4}$  over the first 2% of steps, then gradually decays to 0.

**Stage-2: Pre-training** We employ large-scale pretraining data, primarily consisting of image-text multi-task data, to pre-train models establishing general multimodal capabilities across broad visual scenarios. We utilize a subset of 2.99M samples from LLaVA-OneVision-SingleImage (Li et al., 2024a) as training corpus, which incorporates 2.9M image-text pairs and 93K textual instruction-tuning samples from Evo-Instruct (Chen et al., 2024a). For video-text corpus, we utilize a sampled subset of 75K video QAs from FineVideo (Farré et al., 2024) and 38K captioning pairs from ShareGPT4Video (Chen et al., 2024b). To mitigate catastrophic forgetting, we retain 5% of the previous image and video data in our training corpus. Alongside the cubing network and resampler, we also unfreeze the visual encoder to improve the visual representation. We train models for 1 epoch with a  $1e^{-5}$  initial learning rate, with the same warmup and decay schedule as stage-1.

**Stage-3: Supervised Fine-tuning** We primarily leverage extensive video-text paired corpus to train our models in this stage, enabling robust video understanding capabilities. We primarily utilize a subset of 476K video-text samples sourced from VideoChat2-IT (Li et al., 2024b), and a subset of 79K samples from ShareGPTVideo (Zhang et al., 2024b) as the video corpus. To enhance adaptation to long video scenarios, we further integrate 5K samples from MovieChat (Song et al., 2024) and 39K samples derived by (Chen et al., 2024c) from the Shot2Story dataset (Han et al., 2023). The image-text corpus incorporates a sampled subset of 100K multi-image, multi-task understanding samples from LLaVA-OneVision-MultiImages (Li et al., 2024a). We also preserve a subset of training data from the previous stage, consisting of 40K text-image pairs and 9K textual instruction-tuning samples. We train all parameters for 1 epoch with a learning rate that warms up to  $1e^{-5}$  over the 0.02 epoch, followed by gradual decay to 0 for the remaining duration.

During training, all videos sampled at 1FPS to extract full frames. For videos exceeding 420s, we uniformly extract 420 frames to maintain computational tractability. Images are processed using AnyRes with a resolution of  $384 \times 384$ . For all stages, the Gumbel noise learning rate  $\eta$  (initialized at 1.0) undergoes cosine annealing to 0.01 within: 0.8 epoch (Stage 1) or 0.6 epochs (Stages 2-3).

		Stage-1	Stage-2	Stage-3
Vision	<b>Resolution</b>	$384 \times 384$	$384 \times 384$	$384 \times 384$
	FPS,#Frames	1, Max 420	1, Max 420	1, Max 420
Data	<b>Image-Text</b> #samples	LCS, OBELICS 558K, 20K	LLaVAOV-SingleImage 2.99M	LLaVAOV-MultiImages 100K
	<b>Video-Text</b> #samples	FineVideo, ANetCaptions 87K, 8K	FineVideo, ShareGPT4Video, ANetCaptions 118K	Sec. 3. 3 599K
Model	<b>Trainable</b> #Parameters	Cubing, Resampler 75M	Cubing, Resampler, ViT 500M	Full Model 8B
Training	<b>Anneal:</b> $\eta_0, \eta_T$ , ratio	1.0, 0.01, 0.8	1.0, 0.01, 0.6	1.0, 0.01, 0.6
	<b>LR:</b> $\theta_c, \theta_r, \theta_v, \theta_l$	$1e^{-4}, 1e^{-4}, -, -$	$2e^{-5}, 2e^{-5}, 2e^{-5}, -$	$1e^{-5}$
	<b>Epoch</b>	1	1	1

Table 1: Detailed configuration for each training stage.

## 4 Experiment

Models	Size	#Tokens	#Train	MMBench-Video	MVBench	MLVU	Video-MME
Duration	/Frame	V-T	3 min	16 sec	3~120 min	1~60 min	
<i>Proprietary Models</i>							
GPT4-V (OpenAI, 2023)	-	-	-	1.53	43.7	-	60.7
GPT4-o (OpenAI, 2024)	-	-	-	1.63	64.6	66.2	77.2
<i>Open-Source Video LMMs</i>							
LLaMA-VID (Li et al., 2024d)	7B	2	0.4M	1.08	41.5	33.2	-
LongLLaVA (Wang et al., 2024)	9B	144	0.5M	-	49.1	-	43.7
Chat-UniVi (Jin et al., 2024)	7B	112	100K	1.06	42.9	-	45.9
ShareGPT4Video (Chen et al., 2024b)	8B	144	4.8M	1.05	51.2	46.4	43.6
LLaVA-NeXT-Video (Zhang et al., 2024c)	7B	144	100K	1.14	33.7	-	46.5
VideoLLaMA2 (Cheng et al., 2024)	7B	32	10.7M	1.08	54.6	48.5	46.6
LongVA Zhang et al. (2024a)	7B	144	-	-	-	56.3	54.3
VideoChat2 (Li et al., 2024b)	7B	64	2.8M	1.22	<b>60.4</b>	47.9	54.6
mPLUG-Owl3 (Ye et al., 2024)	8B	729	134K	<b>1.35</b>	54.5	-	53.5
Fixed-LLama3.1	8B	12.8	0.8M	0.71	45.2	50.2	45.0
Quicksviewer-LLama3.1	8B	12.8	0.8M	0.87	53.9	58.6	47.6
Quicksviewer	8B	12.8	0.8M	<u>1.24</u>	<u>55.6</u>	<b>61.5</b>	<b>56.9</b>

Table 2: Video benchmarking results between Quicksviewer and baselines under comparable total sequence length. Quicksviewer achieves multiple SOTA performance while using fewer tokens per frame (up to 5% of baseline) and substantially less video-text training samples.

### 4.1 Implementation Details

We use SigLIP (Zhai et al., 2023) (soo400m-path14-384) as our visual encoder inconsistent with previous works. We adopt Qwen2.5 (Yang et al., 2024) as the language backbone for our standard implementation (i.e., Quicksviewer), while utilizing Llama3.1 (Touvron et al., 2023) as the alternative LLM for another version (i.e., Quicksviewer-Llama3.1) for comprehensive exploration. We use AdamW (Loshchilov & Hutter, 2017) optimizer with a cosine scheduler for all training stages. The number of tokens generated from visual encoder and resampler are  $N_1 = 576$ , and  $N_2 = 64$ , respectively. The discounting factor of momentum is set to  $\alpha = 0.9$ . The penalty weight to the auxiliary loss is set to  $\beta = 0.001$ . We use FPQ=5 for all models. Our models is trained on 48 NVIDIA A100 GPUs.

### 4.2 Experiments on Video Understanding

We train a direct baseline, Fixed-Llama3.1, which utilizes uniform temporal partitioning with the same FPQ of input videos. For an unbiased comparison, we evaluate with baselines configured with comparable total sequence lengths, maintaining equivalent computational budgets.

**Benchmarks and Metrics** We evaluate the our models on widely used video understanding benchmarks Video-MME (Fu et al., 2024), MVBench (Li et al., 2024b), and MLVU (Zhou et al., 2024) to investigate the effectiveness. VideoMME is a general video understanding benchmark that collect videos (1min~1hour) from Youtube with manual annotations. MVBench covers 20 challenging tasks ranging from perception to cognition. MLVU (3mins~2hours) refers to an long video understanding benchmark for long-term inference.

**Quantitative Results** We adopt empirically optimized configuration: 5 FPS with a maximum of 720 frames for all benchmarks. As the main result shown in Table 2, our standard model achieves SOTA performance on Video-MME and MLVU, and the competitive performance on MVbench with significantly fewer tokens and training volumes. Specifically, our model achieves SOTA performance on Video-MME, albeit with slightly inferior results on long videos. This demonstrates that the encoding paradigm harness the scaling benefits by high frame rate. In comparison with the direct baseline with fixed cubing strategy, our model obtain large improvements, suggesting the effectiveness of the cuing strategy. Our approach achieves SOTA on MLVU while demonstrating competitive performance on MVBench, despite utilizing substantially less training data (only 28% of

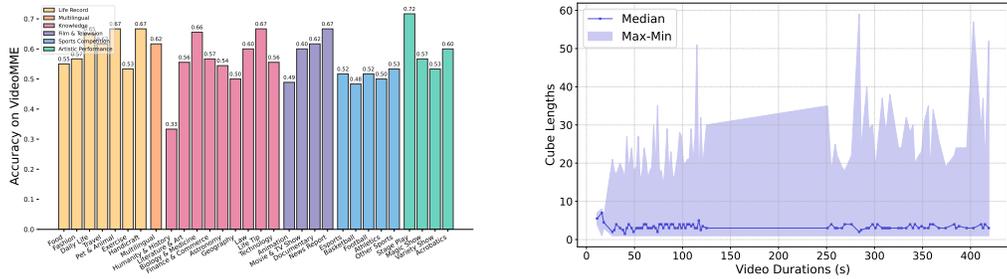


Figure 3: (a) Left: Performance of Quicksviewer on particular domains and categories of Video-MME. (b) Right: Distribution of cube lengths across Video-MME videos.

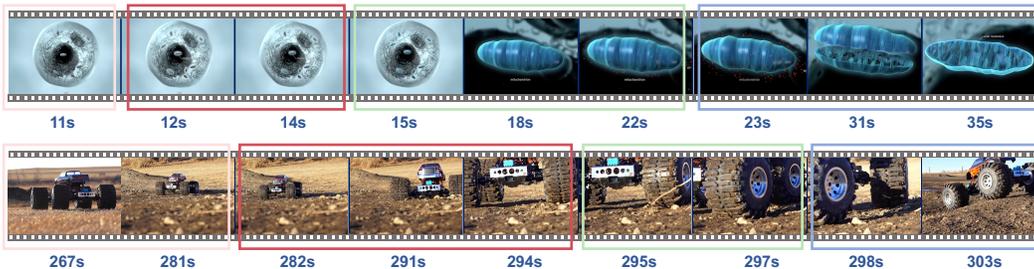


Figure 4: The "Visual Lag" phenomenon occurring during the model's cube-based segmental comprehension, where current cubes incorporate terminal frames from preceding event scenes to enable retrospective understanding.

VideoChat2's and 7.5% of VideoLLaMA2's requirements). This evidences our network's exceptional learning efficiency.

**Analysis** We further analyze the model performance across distinct domain categories in Video-MME, systematically examine both capability advantages and limitations. As illustrated in Figure 3 (a), bars sharing identical colors belong to the same domain. Primarily, we observe consistent model performance across all domains, with mean scores of 0.61, 0.62, 0.55, 0.60, 0.51, and 0.61 respectively, suggesting limited domain-specific variation in question difficulty. Secondly, the model demonstrates suboptimal performance (below 50%) in three categories: Humanity & History, Animation, and Basketball. This may indicate persistent challenges in fine-grained character recognition that require further improvement.

We further analyze the distribution of cube lengths on Video-MME, with results shown in Figure 3 (b). Based on the predefined FPQ, we found the median cube length approximates 5 frames. Notably, the model demonstrates a tendency to partition diverse length of cubes for longer videos, which aligns with the variable viewing speeds in human perception of lengthy videos.

### 4.3 Analysis of the Cubes Partitioning

To investigate how the trained model partitions cubes for understanding, we analyzed two representative video cases by examining cubes relative to content transitions. As shown in Figure 3, each box refers a cube spanning from its start to end timestamps (with similar intermediate frames omitted). We reveal a "Visual Lag" phenomenon during cube-based video perception of the model: terminal frames from preceding event scenes are incorporated into cubes containing subsequent event scenes. For example in the first video, the initial frames of cubes 2-4 respectively contain content from three event scenes: 1) cellular details, 2) mitochondrial positioning, and 3) ATP synthesis exhibiting in cubes 1-3 respectively. We posit this mechanism enables the model to retain partial memory of preceding scenes to facilitate current scene understanding.

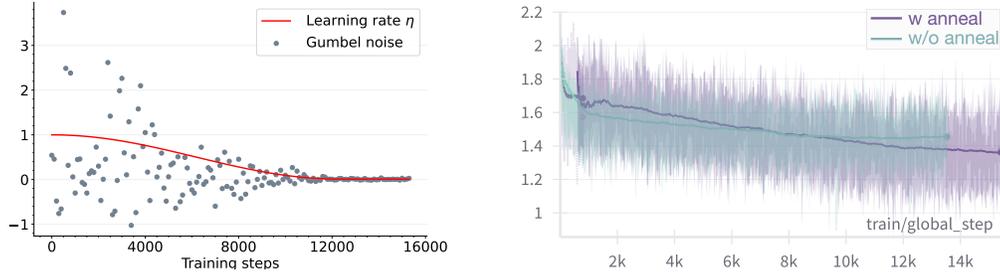


Figure 5: (a) Left: Gumbel noise progressively anneals to 0.001 following the decaying learning rate with cosine scheduler. (b) Right: Compared to non-annealed training (cyan curve), adding Gumbel noise annealing (purple curve) yields more stable and superior loss convergence.

#### 4.4 Analysis of the Annealing Strategy

Traditional Gumbel-Softmax training controls sampling randomness exclusively through temperature adjustment, making it unsuitable for training the cubing network as a component of an LMM. To resolve this issue, we propose annealing the Gumbel noise, which substantially improves both training stability and effectiveness. To further evaluate the performance of the proposed annealing mechanism, we examine the evolution of Gumbel noise values throughout a training epoch in with the annealed learning rate. For clear visualization, we uniformly sample training steps at 100-step intervals, as illustrated in Figure 5 (a). From the figure, we observe that in the early training stages, larger Gumbel noise effectively facilitates exploration for the cubing network. As training progresses, the Gumbel noise gradually converges to the predefined value of 0.01. This allows the model to leverage its learned segmentation mechanism for video understanding in later stages, stabilizing the training process and achieving optimal performance.

Figure 5 (b) compares the loss trajectories of models trained with and without the annealing mechanism. Initialized from the same checkpoint from Stage-2, we train parallel models using both approaches and monitor loss variations throughout one epoch to assess learning efficiency and stability. Our analysis reveals that the model benefits from the progressive annealing of Gumbel noise in the later stages. During this phase, the model effectively utilizes its learned cubing mechanism to accelerate loss minimization, achieving superior convergence efficiency, demonstrating the effectiveness and training stability.

#### 4.5 Ablation Studies

Ablation Components					Video-MME	
PE	Cubing Network	Penalty	Annealing	Trainable	Overall 1~60 min	Long 30~60 min
2D	ViT first 2-layers	0.1	-	$\theta_c, \theta_r, \theta_v$	33.92	35.56
2D	ViT full	0.1	-	$\theta_c, \theta_r, \theta_v$	41.22	38.67
3D	ViT full	0.1	-	$\theta_c, \theta_r, \theta_v$	44.37	40.67
3D	ViT full	0.001	-	$\theta_c, \theta_r, \theta_v$	44.66	40.44
3D	ViT full	0.001	annealing	$\theta_c, \theta_r, \theta_v$	45.44	43.44
3D	ViT full	0.001	annealing	All	<b>45.96</b>	<b>38.44</b>

Table 3: Ablation results of Stage-3 training initialized from a checkpoint pretrained only with image data (Stage 1-2). The optimal configuration: 3D positional encoding, Gumbel noise annealing with 0.001 penalty weight, and full trainable parameters, demonstrating superior performance.

We conduct comprehensive ablation studies to evaluate the efficacy of the components leveraged in Quickviewer. To establish a simple baseline, we first train a Llama3.1 (Touvron et al., 2023) model through Stages 1-2 using only image-text data introduced in Sec. 3, deliberately excluding video inputs. This image-only pretrained checkpoint then serves as the initialization point for systematically investigating various Stage-3 configurations with video-text data.

**Cubing network with ViT** To accelerate cube processing, we investigate the feasibility of using only the initial  $n$  layers of ViT for the cubing network. Our ablation study employs the first 2 ViT layers for cube feature generation while maintaining all other model components unchanged. As demonstrated in Table 7, this configuration results in significant performance degradation, indicating that shallow visual features are insufficient for effective cubes partitioning.

**3D positional Encoding** We systematically evaluate the impact of 3D positional encoding compared to the original 2D formulation. Implementing this modification while keeping all other parameters fixed in Stage-3 training, our experiments demonstrate a consistent accuracy improvement of +3.15% (Table 7), confirming the benefits of spatiotemporal position awareness for video understanding.

### Penalty weight to the auxiliary loss

**The annealing strategy** Having established the optimal penalty weight, we proceed to evaluate the efficacy of our proposed Gumbel noise annealing strategy. This approach systematically reduces exploration randomness during training, transitioning from aggressive parameter space exploration to fine-tuned optimization. Comparative results in Table 7 demonstrate consistent performance improvements over the fixed-noise baseline, validating the benefits of noise scheduling.

**The tuning parameters** We examine the impact of trainable parameters on video understanding by comparing two training regimes: (1) our baseline approach that only fine-tune the ViT and resampler parameters during Stage-3, versus (2) a full-parameter optimization strategy that additionally fine-tunes the LLM backbone. As evidenced in Table 7, comprehensive parameter training yields superior benchmark performance, suggesting that joint visual-linguistic optimization enhances multimodal alignment for video understanding tasks.

## 4.6 Qualitative Analysis

Our unified perception paradigm demonstrates efficient visual understanding capabilities, successfully processing: 1) lengthy videos, 2) high-resolution single images, and 3) multi-image contextual reasoning tasks. Through representative qualitative analysis, we validate the model’s performance across these diverse inputs. Specifically, we evaluate the video understanding capabilities of our model through documentary recording and sports competitions, while the image understanding proficiency across diverse domains including outdoor road scenes, physics/biological/historical knowledge systems, and multi-image geographical analysis.

In the documentary depicting a penguin chick’s perilous encounter and subsequent escape, our model demonstrates comprehensive video understanding by: (1) identifying the nature of the unexpected attack, (2) precisely locating its temporal occurrence, and (3) summarizing the eventual outcome – showcasing its advanced capabilities in long-form video understanding, including temporal action recognition, event narrative abstraction, and exact timestamp localization.

In a lacrosse match video documenting a scoring play, our model precisely identifies the initiating player when queried about the offensive sequence, subsequently describing the play development and correctly specifying both the scoring player’s identity and jersey number. This demonstrates the model’s dual capability of (1) recognizing individual athletes in sports footage and (2) logically summarizing dynamic game situations.

We further validate our model’s image understanding capabilities across extensive scenarios. As illustrated in Figure 6, these include: 1) traffic sign recognition in driving environments, 2) physics problem solving, 3) biological image interpretation and association, 4) historical scene identification, and 5) multi-image geographical reasoning. These examples demonstrate that while exhibiting strong video understanding, our model maintains robust image understanding capabilities. Benefiting from interleaved image-text training data, the model additionally acquires in-context learning capabilities for complex reasoning tasks.



**User:** What is the main incident that happened in this video?

**Quickviewer:** The main incident that happened in the video is a group of penguins standing on the snow, while a bird approaches them. The bird tries to attack one of the penguins, but it fails.

**User:** During what time period (HH:MM:SS - HH:MM:SS) did the attack occur? Were any penguins injured?

**Quickviewer:** The attack occurred between 0:47 and 0:53, and no penguins were injured during the incident.



**User:** Which White team player started the last offensive sequence?

**Quickviewer:** 13, Barger.

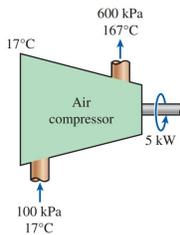
**User:** What happened next?

**Quickviewer:** He passed the ball to his teammate, who then passed it to another teammate. White 7 Miller shot and scored a goal. The teammates celebrated by hugging and cheering.



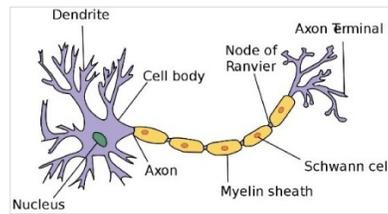
**User:** Where can I go by turning left?

**Quickviewer:** Las Vegas.



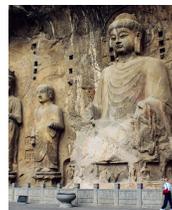
**User:** Describe the process shown in the diagram including the specific values.

**Quickviewer:** The thermodynamic process involves an air compressor of 5kW. The input values are 100 kPa and 17°C, while the output values are 167 kPa and 176°C.



**User:** Identify the biological structure and specify its computational counterpart in CS disciplines.

**Quickviewer:** The image depicts a neuron, a fundamental structure in nervous tissue. The corresponding concept in computer science would be the neuron in artificial neural networks.



**User:** Where is it located?

**Quickviewer:** This is the Longmen Grottoes, located in Luoyang, Henan, China.



**User:** What are the species of the birds, and which one inhabits the red area marked in the first map?

**Quickviewer:** The first bird is a South American ostrich, the second is an African ostrich, and the third is a cassowary. The African ostrich inhabits the red area marked in the first map.



Figure 6: Qualitative analysis shows that Quickviewer effectively understands lengthy documentary and sports videos, as well as informative single and multiple images.

## 5 Related Works

### 5.1 LMMs for Video Understanding

The rapid progress of Multimodal Large Language Models (MLLMs) has greatly enhanced video comprehension capabilities (Yao et al., 2024; Xue et al., 2024; Zhang et al., 2024a; Li et al., 2024a). In this section, we discuss MLLMs that excel in textual reasoning and long-video analysis, with a particular emphasis on 7B-scale architectures that are well-suited for real-world deployment, aligning with our goal of long video-text understanding. MiniCPMv2.6 is a highly adaptable multimodal model, demonstrating proficiency in single-frame, multi-frame, and full-video comprehension. Its outstanding performance in scene text recognition (STR) establishes it as a strong baseline for our research, which seeks to further enhance and refine this model. For long video comprehension, LongVA (Zhang et al., 2024a) and LongVila (Xue et al., 2024) introduce the "Needle-in-a-Haystack" evaluation framework, enabling models to process videos with up to 3,000 frames. However, their methodology incorporates semantically unrelated frames, introducing synthetic difficulties that deviate from naturalistic video-text comprehension. While these works illustrate the feasibility of handling long-context video processing, the artificially created challenges limit their applicability to real-world scenarios. To overcome this limitation, the Text Needle-in-Haystack task avoids inserting irrelevant frames and instead leverages only the original video frames to more accurately represent the alignment between videos and textual descriptions.

### 5.2 Video Understanding Datasets

Expanding on the foundation established by Visual Question Answering (VQA), Video Question Answering (VideoQA) extends this task to video-based queries, requiring models to exhibit strong spatiotemporal reasoning capabilities. A range of datasets (Tapaswi et al., 2016; Xu et al., 2017, 2016; Mun et al., 2017; Jang et al., 2017; Yu et al., 2019; Ye et al., 2017; Zeng et al., 2017) have been developed to support research in this domain. Specifically, MOVIE-QA (Tapaswi et al., 2016) and TVQA (Lei et al., 2018) extract scenes from films and TV series, while SUTD-TrafficQA (Xu et al., 2021) presents multiple-choice questions based on diverse traffic incidents. These datasets feature video clips from different environments, with all questions focused solely on the visual elements of the videos, disregarding the textual information naturally present in many scenes.

Several datasets, including NewsVideoQA (Jahagirdar et al., 2022), M4-ViteVQA (Zhao et al., 2022), and RoadTextVQA (Tom et al., 2023), require models to comprehend textual elements within videos to answer questions. NewsVideoQA primarily deals with news footage, where text plays a crucial role in conveying information, necessitating models that integrate both visual and textual cues for accurate question answering. In contrast, RoadTextVQA is centered around driving scenarios, where questions pertain to road signs and text present in driving videos, which can be challenging to recognize due to occlusions, blurring, and perspective distortions. M4-ViteVQA encompasses a diverse range of video content, spanning domains such as shopping, driving, sports, movies, and vlogs. From a modeling perspective, M4-ViteVQA incorporates visual perception modules like OCR and Faster RCNN (Girshick, 2015) in conjunction with a language model. Similarly, NewsVideoQA employs an OCR module and introduces a specialized loss function to optimize OCR-related tasks during training.

## 6 Conclusion

In this paper, we introduced Quicksviewer, an LMM designed for efficient video understanding through a nonuniform perception paradigm. By dynamically partitioning videos into nonuniform cubes and applying adaptive resampling, our approach achieves a  $45\times$  compression rate while maintaining a consistent representation for both images and videos. We demonstrated that integrating the cubing network into end-to-end training via Gumbel Softmax with an improved noise annealing mechanism, enables efficient learning without boundary labels. Furthermore, our model, trained on just 0.8M videos, attains SOTA performance on VideoMME with significantly fewer tokens per frame than baseline methods. To support training on ultra-long videos, we also developed an infra that allows dynamic sequence lengths. These contributions pave the way for efficient and scalable LMMs, facilitating future research in long video understanding.

## References

- Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Buckler, M., Bedoukian, P., Jayasuriya, S., and Sampson, A. Eva<sup>2</sup>: Exploiting temporal redundancy in live computer vision. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 533–546. IEEE, 2018.
- Chen, G. H., Chen, S., Zhang, R., Chen, J., Wu, X., Zhang, Z., Chen, Z., Li, J., Wan, X., and Wang, B. Allava: Harnessing gpt4v-synthesized data for lite vision-language models. *arXiv preprint arXiv:2402.11684*, 2024a.
- Chen, L., Wei, X., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Tang, Z., Yuan, L., et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024b.
- Chen, Y., Xue, F., Li, D., Hu, Q., Zhu, L., Li, X., Fang, Y., Tang, H., Yang, S., Liu, Z., et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024c.
- Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.
- Deepmind, G. Gemini 2.0. <https://deepmind.google/technologies/gemini/>, 2024.
- Farré, M., Marafioti, A., Tunstall, L., Von Werra, L., and Wolf, T. Finevideo. <https://huggingface.co/datasets/HuggingFaceFV/finevideo>, 2024.
- Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.
- Fu, Y. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*, 2024.
- Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- Gumbel, E. J. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.
- Han, M., Yang, L., Chang, X., and Wang, H. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv preprint arXiv:2312.10300*, 2023.
- Herrmann, C., Bowen, R. S., and Zabih, R. Channel selection using gumbel softmax. In *European conference on computer vision*, pp. 241–257. Springer, 2020.
- Jahagirdar, S., Mathew, M., Karatzas, D., and Jawahar, C. V. Watching the news: Towards videoqa models that can read. 2022.
- Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- Jang, Y., Song, Y., Yu, Y., Kim, Y., and Kim, G. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2758–2766, 2017.
- Jin, P., Takano, R., Zhang, W., Cao, X., and Yuan, L. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13700–13710, 2024.

- Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Carlos Niebles, J. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.
- Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S., Rush, A., Kiela, D., et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36:71683–71702, 2023.
- Lei, J., Yu, L., Bansal, M., and Berg, T. L. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018.
- Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.
- Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.
- Li, X., Wang, Y., Yu, J., Zeng, X., Zhu, Y., Huang, H., Gao, J., Li, K., He, Y., Wang, C., et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024c.
- Li, Y., Wang, C., and Jia, J. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024d.
- Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Lllavanext: Improved reasoning, ocr, and world knowledge, 2024.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Mun, J., Hongsuck Seo, P., Jung, I., and Han, B. Marioqa: Answering questions by watching gameplay videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2867–2875, 2017.
- OpenAI. Gpt-4v(ision) system card. <https://openai.com/research/gpt-4v-system-card>, 2023.
- OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024.
- Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernamed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.
- Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.
- Shen, Y., Fu, C., Dong, S., Wang, X., Chen, P., Zhang, M., Cao, H., Li, K., Zheng, X., Zhang, Y., et al. Long-vita: Scaling large multi-modal models to 1 million tokens with leading short-context accuracy. *arXiv preprint arXiv:2502.05177*, 2025.
- 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024.
- Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.
- Tom, G., Mathew, M., Garcia, S., Karatzas, D., and Jawahar, C. V. Reading between the lanes: Text videoqa on the road, 2023. URL <https://arxiv.org/abs/2307.03948>.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

- Wang, X., Song, D., Chen, S., Zhang, C., and Wang, B. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.
- Wang, Y., Li, X., Yan, Z., He, Y., Yu, J., Zeng, X., Wang, C., Ma, C., Huang, H., Gao, J., et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.
- Wenger, S. Video redundancy coding in h. 263+. In *1997 International Workshop on Audio-Visual Services over Packet Networks*. Citeseer, 1997.
- Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., and Zhuang, Y. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.
- Xu, J., Mei, T., Yao, T., and Rui, Y. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.
- Xu, L., Huang, H., and Liu, J. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9878–9888, 2021.
- Xue, F., Chen, Y., Li, D., Hu, Q., Zhu, L., Li, X., Fang, Y., Tang, H., Yang, S., Liu, Z., et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.
- Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.
- Ye, J., Xu, H., Liu, H., Hu, A., Yan, M., Qian, Q., Zhang, J., Huang, F., and Zhou, J. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*, 2024.
- Ye, Y., Zhao, Z., Li, Y., Chen, L., Xiao, J., and Zhuang, Y. Video question answering via attribute-augmented attention network learning. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 829–832, 2017.
- Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., and Tao, D. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9127–9134, 2019.
- Zeng, K.-H., Chen, T.-H., Chuang, C.-Y., Liao, Y.-H., Niebles, J. C., and Sun, M. Leveraging video descriptions to learn video question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.
- Zhang, B., Li, K., Cheng, Z., Hu, Z., Yuan, Y., Chen, G., Leng, S., Jiang, Y., Zhang, H., Li, X., et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.
- Zhang, P., Zhang, K., Li, B., Zeng, G., Yang, J., Zhang, Y., Wang, Z., Tan, H., Li, C., and Liu, Z. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.
- Zhang, R., Gui, L., Sun, Z., Feng, Y., Xu, K., Zhang, Y., Fu, D., Li, C., Hauptmann, A., Bisk, Y., et al. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*, 2024b.
- Zhang, Y., Li, B., Liu, H., Lee, Y., Gui, L., Fu, D., Feng, J., Liu, Z., and Li, C. Llava-next: A strong zero-shot video understanding model. 2024c.

- Zhao, M., Li, B., Wang, J., Li, W., Zhou, W., Zhang, L., Xuyang, S., Yu, Z., Yu, X., Li, G., et al. Towards video text visual question answering: Benchmark and baseline. *Advances in Neural Information Processing Systems*, 35:35549–35562, 2022.
- Zhou, J., Shu, Y., Zhao, B., Wu, B., Xiao, S., Yang, X., Xiong, Y., Zhang, B., Huang, T., and Liu, Z. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.
- Zohar, O., Wang, X., Dubois, Y., Mehta, N., Xiao, T., Hansen-Estruch, P., Yu, L., Wang, X., Juefei-Xu, F., Zhang, N., et al. Apollo: An exploration of video understanding in large multimodal models. *arXiv preprint arXiv:2412.10360*, 2024.

## Appendix

### A Training Data Details

Modality	Task	# Samples	Dataset
Image-Text	Interleaved Pairs	20K	OBELICS
	Single-Image Captioning	558K	LCS
Video-Text	Captioning	95K	FineVideoCaptions, ANetCaptions

Table 4: Training data statistics for the alignment stage.

Modality	Task	# Samples	Dataset
Text	Instruction	93K	Evo-Instruct
Image-Text	Interleaved Pairs	20K	OBELICS
	Single-Image Captioning	50K	LCS
	Single-Image Tasks	2.8M	LLaVAOneVision
Video-Text	Captioning	5K	FineVideoCaptions, AnetCaptions
	VQA	75K	FineVideoQAs
	Dense Captioning	38K	ShareGPT4Video

Table 5: Training data statistics for the pre-training stage.

Modality	Task	# Samples	Dataset
Text	Instruction	9K	Evo-Instruct
Image-Text	Single-Image Tasks	40K	LLaVA-OneVision-SingleImage
	Multi-Images Tasks	100K	LLaVA-OneVision-MultiImages
Video-Text	Captioning	52K	TextVR, MovieChat, YouCook2
	Dense Captioning	4K	ShareGPT4Video
	Classification	1K	Kinetics-710
	VQA	354K	NExT-QA, CLEVRER, EgoQA TGIF, ShareGPTVideo, FineVideoQAs
	Instruction	188K	VideoChatGPT, VideoChat, LongVILA

Table 6: Training data statistics for the supervised fine-tuning stage.

## B Additional Evaluations

Model	NExT-QA	ActivityNet-QA	Video-ChatGPT				
	acc	acc	Correctness	Detail	Context	Temporal	Consistency
LLaMA-VID (7B)	-	47.4/3.3	2.96	3.00	3.53	2.46	2.51
Chat-UniVi (7B)	-	46.1/3.3	2.89	2.91	3.46	2.89	2.81
Video-LLaVA (7B)	-	45.3/3.3	2.87	2.94	3.44	2.45	2.51
VideoChat2 (7B)	68.6	49.1/3.3	3.16	3.08	3.69	2.56	3.14
VideoLLaMA2 (7B)	75.6	50.2/3.3	3.30	33.18	3.78	2.66	3.12
LLaVA-NeXT-Video (7B)	78.2	53.5/3.2	3.39	3.29	3.92	2.60	3.12
Quicksviewer (8B)	77.5	47.6/2.7	3.10	3.11	3.09	2.48	3.04

Table 7: Evaluation results on more benchmarks.