# QUICKSVIEWER: AN LMM WITH EARLY VIDEO COMPRESSION USING ANNEALED GUMBEL-SOFTMAX

**Anonymous authors**
Paper under double-blind review

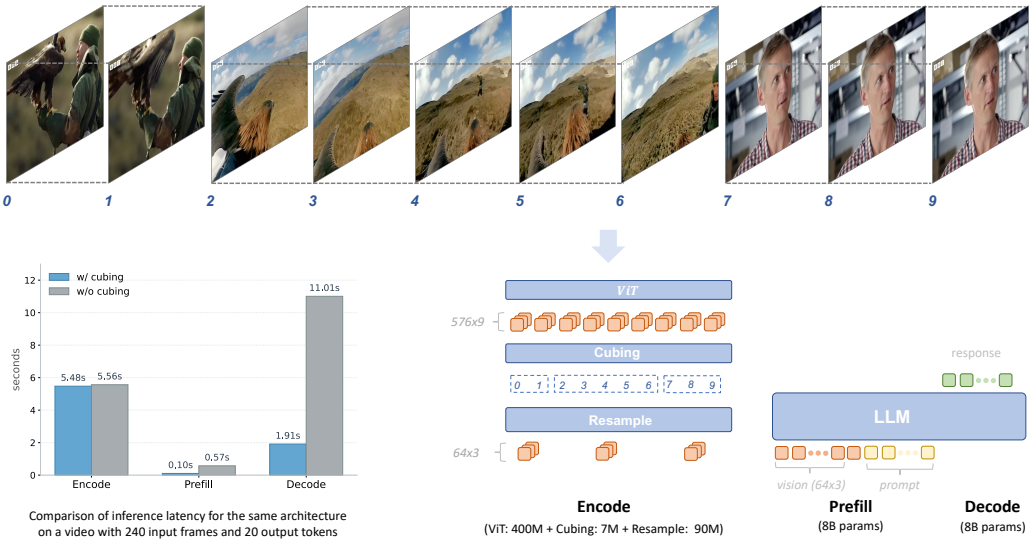



Figure 1: **Left:** With the same architecture and input/output lengths, early video compression via cubing substantially reduces LLM prefill and decoding latency. **Right:** A lightweight cubing network partitions inherently nonuniform video frames (events of 2, 5, and 3 frames) into variable-length cubes, followed by a 3D resampler with spatiotemporal awareness, enabling efficient LLM processing.

## ABSTRACT

Large Multimodal Models (LMMs) uniformly perceive video frames, creating computational inefficiency for videos with inherently varying temporal redundancy. This paper present **Quicksviewer**, an LMM with new perceiving paradigm that partitions a nonuniform redundant video into varying cubes using Gumbel Softmax, followed by a unified resampling for each cube to achieve efficient video understanding. We further propose a Gumbel-noise annealing strategy that enables stable end-to-end LMM training with the discrete cubing network. This simple and intuitive approach dynamically compress video online based on the momentum of frame differences, significantly reducing spatiotemporal complexity (overall 45$\times$ compression rate), while enabling efficient training with large receptive field. 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 samples, our model outperforms contemporary LMMs with the same backbone and comparable parameters amount. **An experiment in A/B pretraining with same backbone and training data shows that, w/ cubing achieves lower training loss and higher benchmark accuracy (+2.1) than the fixed partitioning baseline, strictly validating the benefits.** Moreover, the front-loaded compression in encoding substantially reduces latency in the computationally expensive prefill and decoding, making it suited to mainstream EPD disaggregated inference deployment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

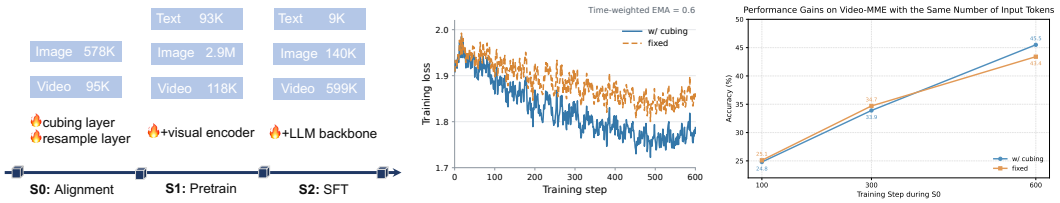



Figure 2: **Left:** Three-stage training. **Middle and right:** Controlled A/B pretraining with the same backbone (Qwen2.5 + SigLIP) and training data (150K captions). Cubing yields lower training loss and improves benchmark accuracy (+2.1) over fixed partitioning at the same input token budget.

## 1 INTRODUCTION

Video understanding with Large Multimodal Models (LMMs) (Deepmind, 2024; OpenAI, 2024; Bai et al., 2025) is fundamental to intelligence in the physical world and requires efficient video perception that addresses the tension between the **temporal redundancy** of frames (Buckler et al., 2018; Wenger, 1997) and the **computational efficiency** of long-context models (Fu, 2024). Extensive studies have been striving to develop LMMs for solving this fundamental issue. Building on devise of frame sampling methods, most efforts typically involve dedicated token merging strategies (Bai et al., 2025; Wang et al., 2025b; Shen et al., 2024; Zohar et al., 2024; Li et al., 2024c; Zhang et al., 2025a) and adapted parallel training infra (Zhang et al., 2024a; Chen et al., 2024d; 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.

**Redundancy in video frames is inherently nonuniform over time, suggesting that the long, high-redundancy intervals and short, low-redundancy intervals should be encoded with an identical token budget.** For example in Figure 1, the first segment (frames 0-1) corresponds to a scene with rapid content changes, followed by a long period of stable visual content frames 2-6), and an almost static segment in the last part (frames 7-9). However, it is nontrivial to determine intervals (*i.e.,* cubes) to guarantee consistent information content and intact video events, so as to ultimately enhance LMM understanding.

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 integrates 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 approach not only enables efficient learning on videos without boundary labels but also insures sufficient sampling over the cubes distribution with **continuous gradient** during stabilized training. Further, we introduce a boundary-weighted thumbnail as a surrogate for discrete boundary optimization to enable continuous gradient flow; appended after the cube tokens, it also serves as a compact memory analogous to Mamba's recurrent state (Gu & Dao, 2023), supporting long-video understanding and online accumulation during inference.

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. With only 0.8M video-text samples in total for training, our model achieves outperformed performance on Video-MME (Fu et al., 2024a) using just up to 5% of the tokens per frame required by baselines. We also conduct extensive ablation studies to show the effectiveness of presented components, including the cubing approach, 3D positional encoding, loss penalty, and the annealing strategy. Moreover, as shown in 1, the front-loaded compression in encoding stage substantially reduces latency in the computationally expensive prefill and decoding stages, making it well suited to current mainstream EPD disaggregated inference deployment.

## 2 Related Works

**LMMs for General Vision Understanding** Early LMMs align pretrained vision encoders with language models through image-text training and visual instruction tuning (Alayrac et al., 2022; Li et al., 2023a; Liu et al., 2023). Further developments improve visual grounding and multimodal reasoning (Bai et al., 2023; Wang et al., 2024c; Qi et al., 2024; Chen et al., 2024f), enabling more detailed perception and complex visual question answering. Recent extensions learn from video-text data to support video understanding (Hong et al., 2024; Chen et al., 2024e; Wang et al., 2024b; Bai et al., 2025; Li et al., 2024a; Yao et al., 2024; Li et al., 2023b; 2024b). These models commonly treat sampled video frames as images, encode them with a shared visual backbone, and aggregate their visual tokens for language-based reasoning.

**LMMs for Efficient Video Understanding** Efficient video LMMs reduce the cost of representing sampled frames at several stages. Before LLM processing, Video-ChatGPT and PLLaVA use pooling to produce compact visual representations (Maaz et al., 2023; Xu et al., 2024). VideoLLaMA2, Kangaroo, Apollo, and BIMBA combine spatial and temporal compression through convolution, clip-level resampling, or learned aggregation (Cheng et al., 2024; Liu et al., 2024b; Zohar et al., 2024; Islam et al., 2025). Adaptive methods such as LongVU and TimeMarker allocate visual tokens according to video length, content, or instructions (Shen et al., 2024; Chen et al., 2024c).

Other methods compress information within LLM or retain a bounded visual memory. VideoStreaming and Video-XL illustrate memory aggregation or compressed-context approaches (He et al., 2024; Qian et al., 2024; Shu et al., 2025; Ye et al., 2024b). Inference-time methods such as DyCoke, PruneVid, FrameFusion, and LLaVA-Scissor reduce tokens or cached states using temporal redundancy and token importance (Tao et al., 2024; Huang et al., 2024b; Fu et al., 2024b; Shao et al., 2025; Sun et al., 2025). Appendix C provides the detailed literature review.

| Model | Visual tokens / original frame | Compression | Max train frames | Eval frames | $D_{600}$ (%) | Video-MME w/o subs (%) |
|---|---|---|---|---|---|---|
| **Trained compression models** | | | | | | |
| VideoLLaMA2 7B | 72 | 8x | 16 | 16 | 97.33 | 47.9 |
| PLLaVA 7B | 144 | 4x | 16 | 16 | 97.33 | - |
| Kangaroo 8B | 128 | 8x | 160 | 64 | 89.33 | 56.0 |
| TimeMarker 8B | 36* | 16x* | 128 | 120* | 80.00* | 57.3 |
| LongVU 7B | $\leq$144 | Adaptive | - | 1 FP | - | - |
| Video-XL 7B | 144 | 4x pooling | 128 | 128 | 78.67 | 55.5 |
| BIMBA 7B | 144 | 4x | 128 | 128 | 78.67 | - |
| Apollo 7B | 32 | 22.78x | 600 | - | - | 61.3 |
| **Reference video models** | | | | | | |
| LongVA 7B | 144 | 4x | N/A | 128 | 78.67 | 52.6 |
| LongVILA 7B-256f | 256 | 4x | 256 | 256 | 57.33 | 60.1 |
| MiniCPM-V 2.6 | 64 | 16x square* | - | - | - | 60.9 |
| LLaVA-OneVision 7B | 196 | 3.72x | 32 | 32 | 94.67 | 58.2 |
| **Inference-time methods** | | | | | | |
| DyCoke 7B | 122.5 | 1.6x pooled | N/A | 32 | 94.67 | 59.5 |
| DyTo 7B | 36.8 | 15.65x | N/A | 100 | 83.33 | 41.2 |
| METoK 7B | Adaptive | Adaptive | N/A | 32 | 94.67 | 58.4 |
| **Quicksviewer 8B (Ours)** | 12.8 | 45x | 420 | 420 | 30 | 62.3 |

Table 1: Comparison of Quicksviewer with representative video LMMs and compression methods. Metrics are compiled from official implementations or reported results; see Appendix C for definitions and details. Quicksviewer achieves 45$\times$ compression and 62.3% accuracy on Video-MME.

## 3 Approach

We build an efficient LMM that can receive both images and videos. 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 $f_l(\cdot)$ 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 percepion granularity and enables adaptive number of cubes $N_Q$ with duration.

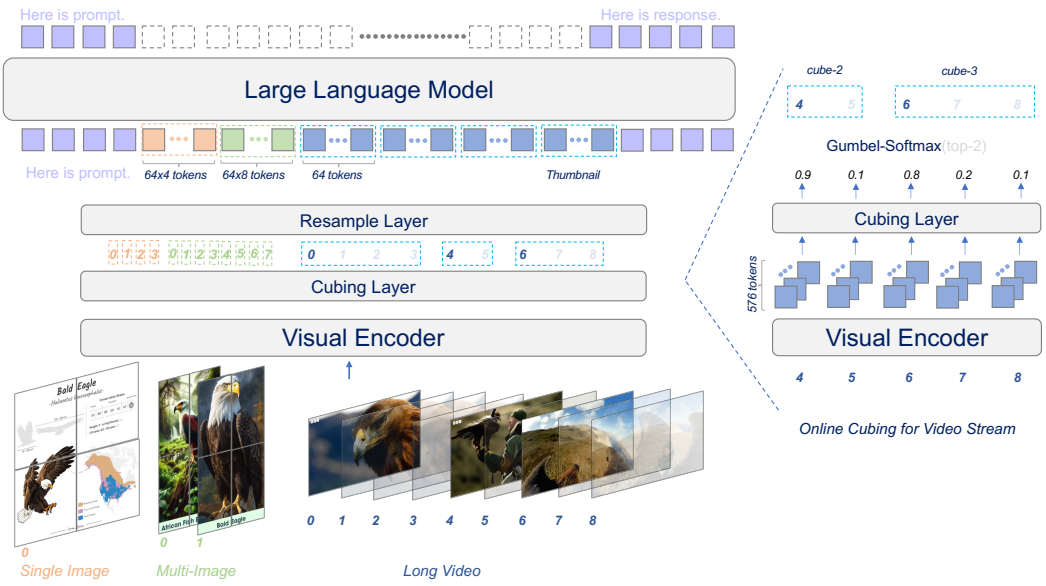



Figure 3: Left: The network architecture of Quicksviewer, that performs unified understanding of videos and images through visual tokens from cascaded modules. Right: The cubing network, that partitions an online video into nonuniform cubes based on Gumbel Softmax.

### 3.1 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 $[F_i]_{i=1,..,N_F}$. And then, each frame is firstly encoded using the visual encoder $f_v$ to obtain $N_1$ visual tokens $\mathbf{F}_j = [\mathbf{v}_i]_{i=1,..,N_1}$, 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:

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

where $\Delta_i = [\delta_j]_{j=1,..,N_1} \in \mathbb{R}^{N_1 \times d}$ 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:

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

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 $[\mathbf{z}_i]_{i=2,..,N_F}$ vectors to obtain the indices of $N_Q - 1$ largest values $[l_i]_{i=1,..,N_Q-1}$, which are selected as keyframes. The keyframe $F_{l_i}$, along with its subsequent consecutive non-keyframes, forms the cube $Q_{i+1} = [F_{l_i}, .., F_{l_{i+1}})$. 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):

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

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.

### 3.2 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, an unified resampling is performed to obtain $N_2 = 64$ visual tokens for each cube. For images, we first adopt the AnyRes (Liu et al., 2024a) 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 $[\mathbf{z}_i]_{i=1,..,N_F}$ with their corresponding frame representations $[\mathbf{F}_i]_{i=1,..,N_F}$, 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. Moreover, appended after the cube tokens, the thumbnail serves as a compact memory analogous to Mamba's recurrent state (Gu & Dao, 2023), supporting long-video understanding and online accumulation of visual information during inference.

### 3.3 LLM & Optimization

Given the visual tokens coming from videos or images, and the prompting tokens encoded from the LLM, the final sequence feed into the LLM is the concatenation of these tokens, separated by special indicator tokens and absolute timestamps as a float number in 0.01-second units for videos.

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.

## 4 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 captaining 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 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., 2024d) from the Shot2Story dataset (Han et al., 2023). The image-text corpus incorporates a sampled subset of 100K mult-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 lr that warms up to $1e^{-5}$ over the 0.02 epoch, followed by a cosine decay to 0.

During training, all videos sampled at 1FPS to extract full frames. For videos exceeding 420s, we uniformly extract 420 frames. The Gumbel noise learning rate $\eta$ (initialized at 1.0) undergoes cosine annealing to 0.001 within: 0.8 epoch (Stage 1) or 0.6 epochs (Stages 2-3).

| **Models** | **Size** | **#Tokens** | **#Train** | **MMBench-Video** | **MVBench** | **MLVU** | **Video-MME** |
|---|---|---|---|---|---|---|---|
| Duration | | /Frame | V-T | 3 min | 16 sec | 3∼120 min | 1∼60 min |
| *Proprietary Models* | | | | | | | |
| GPT4-V (OpenAI, 2023) | - | - | - | 1.53 | 43.7 | - | 60.7 |
| GPT4-o (OpenAI, 2024) | - | - | - | 1.63 | 64.6 | 66.2 | 77.2 |
| *Open-Source Video LMMs* | | | | | | | |
| LLaMA-VID (Li et al., 2024d) | 7B | 2 | 0.4M | 1.08 | 41.5 | 33.2 | - |
| LongLLaVA (Wang et al., 2024e) | 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 | 60.4 | 47.9 | 54.6 |
| mPLUG-Owl3 (Ye et al., 2024a) | 8B | 729 | 134K | <u>1.35</u> | 54.5 | - | 53.5 |
| Video-XL (Shu et al., 2025) | 7B | 9∼72 | 0.1M | - | 55.3 | 45.5 | 55.5 |
| LongVILA (Chen et al., 2024d) | 7B | 256 | 0.2M | - | 67.1 | - | 60.1 |
| LongVU (Shen et al., 2024) | 7B | 64 | 0.8M | - | 66.9 | 65.4 | <u>60.6</u> |
| Quicksviewer | 8B | 12.8 | 0.8M | **1.37** | 62.3 | **66.1** | **62.3** |

Table 2: Compared to baselines with similar model size, Quicksviewer achieves multiple SOTA performance while using fewer tokens per frame (up to 5% of baseline) and less training samples.

# 5 EXPERIMENT

## 5.1 IMPLEMENTATION DETAILS

We use SigLIP (Zhai et al., 2023) (soo400m-path14-384) as our visual encoder in consistent with previous works. We adopt Qwen2.5 (Yang et al., 2024a) as the language backbone. 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 stages.

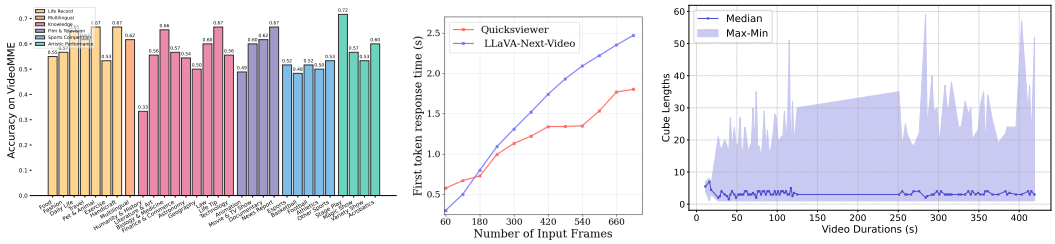



Figure 4: (a) Domain-specific results on VideoMME. (b) Inference time comparison with LLaVA-Next-Video. (c) Distribution of cube lengths on VideoMME.

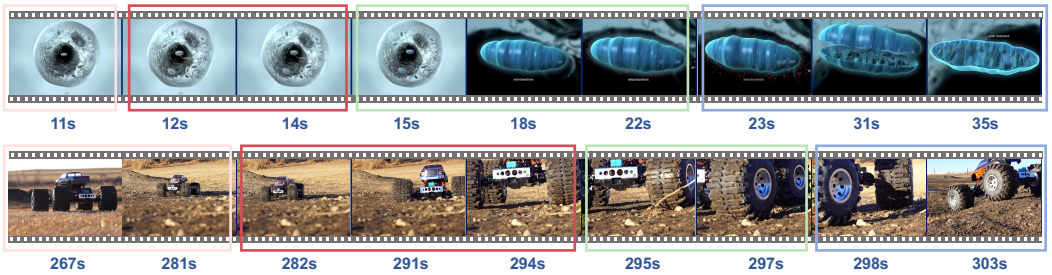



Figure 5: The **"Visual Lag"** phenomenon occurring during the model's cube-based segmental comprehension, where current cubes incorporate terminal frames from preceding event scenes to enable retrospective understanding.

### 5.2 Experiments on Video Understanding

We evaluate baselines with comparable sequence lengths and parameters for fair comparison. 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., 2024a), MVBench (Li et al., 2024b), and MLVU (Zhou et al., 2024) to investigate the effectiveness.

**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 the best performance on Video-MME and MLVU, and the competitive performance on MVbench with significantly fewer tokens and training volumes. Specifically, our model achieves optimal 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. Our approach achieves the best performance on MLVU while demonstrating competitive performance on MVBench, despite utilizing substantially less training data (only 28% of VideoChat2's and 7.5% of VideoLLaMA2's requirements). This evidences our network's 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 4 (a), bars sharing identical colors belong to the same domain. Primarily, we observe consistent model performance across all domains, 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 4 (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.

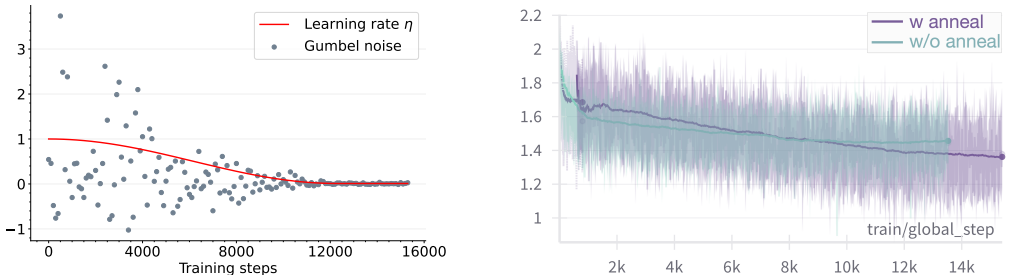



Figure 6: (a) Left: Gumbel noise progressively anneals to 0.001 following the decaying learning rate with cosine scheduler. (b) Right: Compared to non-annealed training (cyan curve), adding Gumbel noise annealing (purple curve) yields more stable and superior loss convergence.

| Ablation Components | | | | | Video-MME | |
|---|---|---|---|---|---|---|
| | | | | | Overall | Long |
| **PE** | **Cubing Network** | **Penalty** | **Annealing** | **Trainable** | 1∼60 min | 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 | **45.96** | **38.44** |

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

### 5.3 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 4, 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.

### 5.4 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 6 (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 6 (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 stabil-

ity. 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.

### 5.5 Ablation Studies

We conduct comprehensive ablation studies to evaluate the efficacy of the components leveraged in Quicksviewer. 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 introduce in Sec. 4, 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 3, 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 3), confirming the benefits of spatiotemporal position awareness for video understanding.

**Penalty weight to the auxiliary loss** The penalty weight applied to the $L_2$-norm of $\mathbf{z}$ values regulates the scale of logistic outputs, consequently influencing partition behavior. Through systematic ablation while maintaining other parameters fixed, we identify $\beta = 0.001$ as the optimal weighting that simultaneously: (1) maintains $\mathbf{z}$ values in an effective operational range, and (2) achieves superior model performance (Table 3).

**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 3 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 3, comprehensive parameter training yields superior benchmark performance, suggesting that joint visual-linguistic optimization enhances multimodal alignment for video understanding tasks.

## 6 Conclusion

In this paper, we introduce Quicksviewer, an LMM designed for efficient video understanding through a nonuniform perception paradigm. By dynamically partitioning videos into nonuniform cubes and applying 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. Wen train model on 0.8M videos, attains superior performance on VideoMME with significantly fewer number of vision tokens. Moreover, front-loading video compression into the encoding stage substantially reduces latency in the computationally expensive prefill and decoding stages, making Quicksviewer well suited to mainstream EPD disaggregated inference deployment.

## 7 AI USE STATEMENT

Generative AI tools were used to assist with literature research, manuscript drafting and editing, code generation and debugging, and experimental analysis. They were not used to create or edit any existing data, experimental settings, and results. All research decisions, outputs, and AI-assisted materials were verified by the authors, who take full responsibility for the final content.

## REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Mark Buckler, Philip Bedoukian, Suren Jayasuriya, and Adrian Sampson. Eva$^2$: Exploiting temporal redundancy in live computer vision. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 533–546. IEEE, 2018.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for lite vision-language models. *arXiv preprint arXiv:2402.11684*, 2024a.

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. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024b.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. TimeMarker: A Versatile Video-LLM for Long and Short Video Understanding with Superior Temporal Localization Ability, 2024c. URL https://arxiv.org/abs/2411.18211v1. Version v1; public by June 30, 2025.

Yukang Chen, Fuzhao Xue, 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*, 2024d.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024e.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024f.

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.

Google Deepmind. Gemini 2.0. https://deepmind.google/technologies/gemini/, 2024.

Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong 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*, 2024a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining Similarity and Importance for Video Token Reduction on Large Visual Language Models, 2024b. URL https://arxiv.org/abs/2501.01986v1. Version v1; public by June 30, 2025.

Yao Fu. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*, 2024.

Lishuai Gao, Yujie Zhong, Yingsen Zeng, Haoxian Tan, Dengjie Li, and Zheng Zhao. LinVT: Empower Your Image-level Large Language Model to Understand Videos, 2024. URL https://arxiv.org/abs/2412.05185v2. Version v2; public by June 30, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. URL https://arxiv.org/abs/2312.00752.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

Yanan Guo, Wenhui Dong, Jun Song, Shiding Zhu, Xuan Zhang, Hanqing Yang, Yingbo Wang, Yang Du, Xianing Chen, and Bo Zheng. FiLA-Video: Spatio-Temporal Compression for Fine-Grained Long Video Understanding, 2025. URL https://arxiv.org/abs/2504.20384v1. Version v1; public by June 30, 2025.

Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv preprint arXiv:2312.10300*, 2023.

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, 2024. URL https://arxiv.org/abs/2404.05726v2. Version v2; public by June 30, 2025.

Charles Herrmann, Richard Strong Bowen, and Ramin Zabih. Channel selection using gumbel softmax. In *European conference on computer vision*, pp. 241–257. Springer, 2020.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language Instructed Temporal-Localization Assistant, 2024a. URL https://arxiv.org/abs/2403.19046v1. Version v1; public by June 30, 2025.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual Token Pruning for Efficient Video Large Language Models, 2024b. URL https://arxiv.org/abs/2412.16117v1. Version v1; public by June 30, 2025.

Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. *arXiv preprint arXiv:2503.09590*, 2025.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Lingyu Kong, Hongzhi Zhang, Jingyuan Zhang, Jianzhao Huang, Kunze Li, Qi Wang, and Fuzheng Zhang. Clapper: Compact Learning and Video Representation in VLMs, 2025. URL https://arxiv.org/abs/2505.15529v1. Version v1; public by June 30, 2025.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36:71683–71702, 2023.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

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*, 2023b.

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

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024c.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024d.

Yixuan Li, Changli Tang, Jimin Zhuang, Yudong Yang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. Improving LLM Video Understanding with 16 Frames Per Second, 2025. URL https://arxiv.org/abs/2503.13956v2. Version v2; public by June 30, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024a.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024b.

Zhihang Liu, Chen-Wei Xie, Pandeng Li, Liming Zhao, Longxiang Tang, Yun Zheng, Chuanbin Liu, and Hongtao Xie. Hybrid-Level Instruction Injection for Video Token Compression in Multi-modal Large Language Models, 2025. URL https://arxiv.org/abs/2503.16036v1. Version v1; public by June 30, 2025.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*, 2024c.

Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: On-Demand Spatial-Temporal Understanding at Arbitrary Resolution, 2024d. URL https://arxiv.org/abs/2409.12961v4. Version v4; public by June 30, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. VideoGPT+: Integrating Image and Video Encoders for Enhanced Video Understanding, 2024. URL https://arxiv.org/abs/2406.09418v1. Version v1; public by June 30, 2025.

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.

OpenBMB. MiniCPM-V 2.6: Official model release and documentation, 2024. URL https://github.com/OpenBMB/MiniCPM-V/tree/48c0611a3f578abc70586ec0d13e2a9161b8371c. Historical repository snapshot, June 30, 2025; the 2024 MiniCPM paper alone does not describe the 2.6 video model.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: Train large vision-language models diving into details through chain of manipulations. *arXiv preprint arXiv:2402.04236*, 2024.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming Long Video Understanding with Large Language Models, 2024. URL https://arxiv.org/abs/2405.16009v1. Version v1; public by June 30, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic Token Merging for Fast Video Large Language Models, 2025. URL https://arxiv.org/abs/2505.21334v2. Version v2; public by June 30, 2025.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, 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.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Yunhang Shen, Chaoyou Fu, Shaoqi Dong, Xiong Wang, Peixian Chen, Mengdan Zhang, Haoyu Cao, Ke Li, Xiawu Zheng, Yan Zhang, et al. Long-vita: Scaling large multi-modal models to 1 million tokens with leading short-context accuray. *arXiv preprint arXiv:2502.05177*, 2025.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26160–26169, 2025.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, 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.

Boyuan Sun, Jiaxing Zhao, Xihan Wei, and Qibin Hou. LLaVA-Scissor: Token Compression with Semantic Connected Components for Video LLMs, 2025. URL https://arxiv.org/abs/2506.21862v1. Version v1; public by June 30, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic Compression of Tokens for Fast Video Large Language Models, 2024. URL https://arxiv.org/abs/2411.15024v3. Version v3; public by June 30, 2025.

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.

Han Wang, Yuxiang Nie, Yongjie Ye, Deng GuanYu, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-VLM: Simple Dynamic Visual Token Compression for VideoLLM, 2024a. URL https://arxiv.org/abs/2412.09530v1. Version v1; public by June 30, 2025.

Mengyue Wang, Shuo Chen, Kristian Kersting, Volker Tresp, and Yunpu Ma. METok: Multi-Stage Event-based Token Compression for Efficient Long Video Understanding, 2025a. URL https://arxiv.org/abs/2506.02850v1. Version v1; public by June 30, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024b.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024c.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. ReTaKe: Reducing Temporal and Knowledge Redundancy for Long Video Understanding, 2024d. URL https://arxiv.org/abs/2412.20504v5. Version v5; public by June 30, 2025.

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

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025b.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. LongVLM: Efficient Long Video Understanding via Large Language Models, 2024. URL https://arxiv.org/abs/2404.03384v3. Version v3; public by June 30, 2025.

Stephan Wenger. Video redundancy coding in h. 263+. In *1997 International Workshop on Audio-Visual Services over Packet Networks*. Citeseer, 1997.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Chenyu Yang, Xuan Dong, Xizhou Zhu, Weijie Su, Jiahao Wang, Hao Tian, Zhe Chen, Wenhai Wang, Lewei Lu, and Jifeng Dai. PVC: Progressive Visual Token Compression for Unified Image and Video Processing in Large Vision-Language Models, 2024b. URL https://arxiv.org/abs/2412.09613v1. Version v1; public by June 30, 2025.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*, 2024a.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. VoCo-LLaMA: Towards Vision Compression with Large Language Models, 2024b. URL https://arxiv.org/abs/2406.12275v2. Version v2; public by June 30, 2025.

Keunwoo Peter Yu, Achal Dave, Rares Ambrus, and Jean Mercat. Espresso: High Compression For Rich Extraction From Videos for Your Vision-Language Model, 2024. URL https://arxiv.org/abs/2412.04729v3. Version v3; public by June 30, 2025.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. TimeSuite: Improving MLLMs for Long Video Understanding via Grounded Tuning, 2024. URL https://arxiv.org/abs/2410.19702v2. Version v2; public by June 30, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

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*, 2024a.

Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*, 2024b.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token, 2025b. URL https://arxiv.org/abs/2501.03895v2. Version v2; public by June 30, 2025.

Y Zhang, B Li, H Liu, Y Lee, L Gui, D Fu, J Feng, Z Liu, and C Li. Llava-next: A strong zero-shot video understanding model. 2024c.

Yiming Zhang, Zhuokai Zhao, Zhaorun Chen, Zenghui Ding, Xianjun Yang, and Yining Sun. Beyond Training: Dynamic Token Merging for Zero-Shot Video Understanding, 2024d. URL https://arxiv.org/abs/2411.14401v2. Version v2; public by June 30, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024e.

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.

Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. *arXiv preprint arXiv:2412.10360*, 2024.

# A Appendix

# B Training Data Details

## B.1 Hyter-parameters

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

Table 4: Detailed configuration for each training stage.

## B.2 Training Data

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

Table 5: 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 6: 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<br>TGIF, ShareGPTVideo, FineVideoQAs |
| | Instruction | 188K | VideoChatGPT, VideoChat, LongVILA |

Table 7: Training data statistics for the supervised fine-tuning stage.

## C Detailed Literature Review

**LMMs for Efficient Video Understanding** Efficient video LMMs reduce the cost of representing sampled frames at several stages. Before language-model processing, spatial pooling and learned resamplers produce compact visual representations, as in Video-ChatGPT, PLLaVA, LLaMA-VID, VideoLLaMA2, and LITA (Maaz et al., 2023; Xu et al., 2024; Li et al., 2024d; Cheng et al., 2024; Huang et al., 2024a). More recent approaches combine spatial and temporal compression through convolution, clip-level resampling, or learned aggregation, including Kangaroo, Apollo, PVC, Clapper, and BIMBA (Liu et al., 2024b; Zohar et al., 2024; Yang et al., 2024b; Kong et al., 2025; Islam et al., 2025). Adaptive methods allocate visual tokens according to video length, content, or instructions: representative examples include LongVU, TimeMarker, Dynamic-VLM, VideoLLaMA3, HICom, LinVT, and FiLA-Video (Shen et al., 2024; Chen et al., 2024c; Wang et al., 2024a; Zhang et al., 2025a; Liu et al., 2025; Gao et al., 2024; Guo et al., 2025).

Other methods compress information within the language model or retain a bounded visual memory. MA-LMM, VideoStreaming, Video-XL, and VoCo-LLaMA illustrate memory aggregation or compressed-context approaches (He et al., 2024; Qian et al., 2024; Shu et al., 2025; Ye et al., 2024b). Inference-time methods such as DyCoke, PruneVid, FrameFusion, ReTaKe, HoliTom, and LLaVA-Scissor reduce tokens or cached states using temporal redundancy and token importance (Tao et al., 2024; Huang et al., 2024b; Fu et al., 2024b; Wang et al., 2024d; Shao et al., 2025; Sun et al., 2025). These methods differ in whether compression precedes the first language-model layer, occurs in deeper layers, or affects only the retained cache; the resulting token budgets therefore measure different costs. We compare these mechanisms separately and report training frames, evaluation sampling, compression denominators, and subtitle protocols alongside benchmark accuracy.

| Model | Visual tokens / original frame | Compression | Max train frames | Eval frames | $D_{600}$ (%) | Video-MME w/o subs (%) |
|---|---|---|---|---|---|---|
| **Trained compression models** | | | | | | |
| LLaMA-VID 7B | 2 | 128x | - | 1 FPS | conditional 0 | - |
| VideoLLaMA2 7B | 72 | 8x | 16 | 16 | 97.33 | 47.9 |
| VideoLLaMA3 7B | Dynamic | Dynamic | 180 | $\leq$180 | $\geq$70.00 | 66.2 |
| PLLaVA 7B | 144 | 4x | 16 | 16 | 97.33 | - |
| Chat-UniVi 7B | 7 at F=64 | 36.57x | 64 | 64 | 89.33 | - |
| LITA 7B | 3.56 at F=100 | 71.91x | 100 | 100 | 83.33 | - |
| LongVLM 7B | 3.05 at F=100 | 83.93x | 100 | 100 | 83.33 | - |
| TimeSuite / VideoChat-T | 3 | 65.33x* | 192 | 128 | 78.67 | 46.3 |
| Kangaroo 8B | 128 | 8x | 160 | 64 | 89.33 | 56.0 |
| TimeMarker 8B | 36* | 16x* | 128 | 120* | 80.00* | 57.3 |
| LongVU 7B | $\leq$144 | Adaptive | - | 1 FPS + pruning | - | - |
| Video-XL 7B | 144 | 4x pooling | 128 | 128 | 78.67 | 55.5 |
| LLaVA-Mini 8B | 1 + fused text | Qualified | $<$60 reported | 1 FPS | conditional 0 | - |
| VideoChat-Flash 7B 448px | 16 | 49x input | 512 | $\leq$512 | $\geq$14.67 | 65.3 |

| Model | Visual tokens / original frame | Compression | Max train frames | Eval frames | $D_{600}$ (%) | Video-MME w/o subs (%) |
|---|---|---|---|---|---|---|
| HICom 7B | 20.5 at F=128 | 35.56x | 32 | 128 | 78.67 | 60.3 |
| PVC / InternVL2 8B | 64 | 16x | 96 | - | - | 64.1 |
| BIMBA 7B | 144 | 4x | 128 | 128 | 78.67 | - |
| Apollo 7B | 32 | 22.78x | 600 | - | - | 61.3 |
| Clapper 7B | 61.25 | 12.8x | 96 | 96 | 84.00 | 62.0 |
| Dynamic-VLM 7B | 100 | 5.76x | 256 | 120 | 80.00 | 60.9 |
| Espresso-align 7B | 0.5 at F=128 | 2P | 128 | 128 | 78.67 | - |
| InternVideo2.5 7B | 16 | - | 512 | - | - | 65.1 |
| F-16 7B | $\lfloor\sqrt{P}/2\rfloor^2/16$ | formula | 1760 | $\leq$1760 | capacity0 | - |
| Oryx 7B | N/4 or N/16 | 4x or16x | 64 /256 | - | - | 58.3 |
| LinVT / Qwen2-VL7B | - | - | - | - | - | 66.4 |
| FiLA-Video 7B | 32N/F | 3x temporal | 96 | - | - | 62.3 |
| VideoGPT + 3.8B | 208 | 4x dual | 16 | 16 | 97.33 | - |
| **Reference video models** | | | | | | |
| LongVA 7B | 144 | 4x | N/A video | 128 | 78.67 | 52.6 |
| LongVILA 7B-256f | 256 | 4x | 256 | 256 | 57.33 | 60.1 |
| LongVILA 7B-1M | 196 | 3.72x | 2048 | - VMME | - | - |
| Long-VITA 14B-128K | 256 | 4x | 512 | 256 | 57.33 | 66.4 |
| NVILA 8B-Video | 32 | 32x | 256 | 256 | 57.33 | 64.2 |
| MiniCPM-V 2.6 | 64 | 16x square* | - | - | - | 60.9 |
| LLaVA-OneVision 7B | 196 | 3.72x | 32 | 32 | 94.67 | 58.2 |
| LLaVA-Video 7B | 169 | 4.31x | 64 | 64 | 89.33 | 63.3 |
| Qwen2.5-VL 7B | Dynamic | 4x ViT | - | $\leq$768 | capacity0 | 65.1 |
| **Inference-time methods (training maximum is N/A for the method)** | | | | | | |
| DyCoke + OV7B K=.5 | 122.5 | 1.6x pooled | N/A | 32 | 94.67 | 59.5 |
| ReTaKe + Qwen2-VL7B | Dynamic input/KV | Separate stages | N/A | $\leq$2048 | capacity0* | - |
| ReTaKe + LLaVA-Video7B | Dynamic input/KV | Separate stages | N/A | $\leq$1024 | capacity0* | - |
| DyTo + LLaVA7B | 36.8 | 15.65x | N/A | 100 candidates | 83.33 + adaptive | 41.2 |
| METok + OV7B | Adaptive | Layer-dependent | N/A | 32 | 94.67 + adaptive | 58.4 |

Table 8: Comparison of representative efficient video LMMs publicly available by June 30, 2025. Video-MME results use the no-subtitle setting. BIMBA follows the v1 128-frame setup (training cap assumed). Video-XL's count of 144 tokens/frame excludes additional visual summarization tokens (VSTs). – denotes unavailable or unresolved values, including unverified subtitle protocols; N/A denotes not applicable. $D_{600} = 100[1 - \min(600, F_{\text{eval}})/600]$ is the sampling-budget omission for 600 input frames, not an architectural limit or an observed adaptive frame-drop rate. `Capacity0` means the reported frame cap is at least 600; "+ adaptive" indicates additional content-dependent pruning. Formatting tokens are excluded. Asterisks mark conditional settings: TimeSuite's ratio uses the raw spatial patch grid; TimeMarker's values use a 600-second video; MiniCPM-V's ratio assumes a single square 448-pixel view; ReTaKe's capacity does not account for adaptive frame selection.

## D LIMITATIONS

This paper explores efficient multimodal large models for video understanding. Specifically, we train a video comprehension model from scratch using moderately-sized datasets based on a language model architecture. Benefiting from a dynamic compression-based efficient encoding scheme, the model processes video frames with high receptive fields during training and demonstrates competitive performance across multiple benchmarks during inference. While comprehensive ablation studies validate the model's effectiveness, two key limitations persist: (1) Training data quality - we observe that most pretraining video captions are summary-type texts, which hinders the model's ability to learn detailed video content understanding; (2) Spatial complexity - we find that using the final layer features of ViT as input to the cubing network for video partitioning yields the best results, but this may not be the optimal approach in terms of spatial complexity. Future work will explore more efficient spatial complexity approaches, such as utilizing features from earlier layers of ViT combined with a more efficient cubing network structure for video partitioning. We will also explore more efficient training data construction methods to enhance the model's learning of detailed video content understanding, such as utilizing video captioning models to generate more fine-grained captions for pretraining videos, and incorporating more video-text datasets with detailed annotations into the training data.

| 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 9: Evaluation results on more benchmarks.

# E Additional Evaluations

# F 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 inpus. 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 7, these includes: 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?

**Quicksviewer:** 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?

**Quicksviewer:** 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?

**Quicksviewer:** 13, Barger.

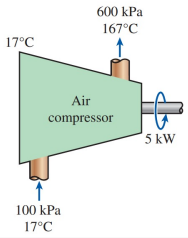
**User:** What happened next?

**Quicksviewer:** 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.



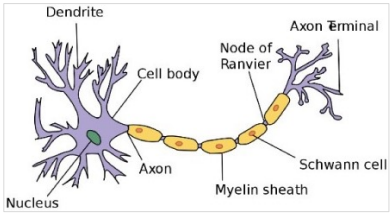

**User:** Where can I go by turning left?

**Quicksviewer:** Las Vegas.




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

**Quicksviewer:** 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.

**Quicksviewer:** 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?

**Quicksviewer:** 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?

**Quicksviewer:** 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 7: Qualitative analysis shows that Quicksviewer effectively understands lengthy documentary and sports videos, as well as informative single and multiple images.