Clapper: Compact Learning and Video Representation in VLMs

Anonymous ACL submission

Abstract

Current vision-language models (VLMs) have demonstrated remarkable capabilities across diverse video understanding applications. Designing VLMs for video inputs requires effectively modeling the temporal dimension (i.e. capturing dependencies across frames) and balancing the processing of short and long videos. Specifically, short videos demand preservation of fine-grained details, whereas long videos require strategic compression of visual information to handle extensive temporal contexts efficiently. However, our empirical analysis reveals a critical limitation: most existing VLMs suffer severe performance degradation in long video understanding tasks when compressing visual tokens below a quarter of their original visual tokens. To enable more effective modeling of both short and long video inputs, we propose **Clapper**, a method that utilizes a slow-fast strategy for video representation and introduces a novel module named TimePerceiver for efficient temporal-spatial encoding within existing VLM backbones. By using our method, we achieves 13x compression of visual tokens per frame (averaging 61 tokens/frame) without compromising QA accuracy. In our experiments, Clapper achieves 62.0% on VideoMME, 69.8% on MLVU, and 67.4% on TempCompass, all with fewer than 6,000 visual tokens per video. The code will be publicly available on the homepage.

1 Introduction

Vision Language Models (VLMs) have achieved significant progress in understanding single images, high-resolution images (Wei et al., 2025; Ye et al., 2023; Chen et al., 2024a), and multiple images (Li et al., 2023; Alayrac et al., 2022; Li et al., 2024a) over the past few years. However, representation and understanding of videos in VLMs remain relatively underexplored. As an extension of images, videos comprise sequences of frames that introduce an additional temporal dimension. Unlike multiple-image inputs, videos continuously extend over time, typically recorded at 30 frames per second (fps) or 24 fps, offering rich and dynamic information. This temporal richness also imposes substantially higher computational demands. For instance, even when the video is sampled at a reduced rate of 1 fps, if we use 196 visual tokens per frame (Li et al., 2024a; Zhang et al., 2024c), we would need to encode 110,760 tokens in VLMs for a 10-minute video.

Many VLM works have been proposed to model the temporal characteristic of video data for video understanding. Some methods (Lin et al., 2023; Wang et al., 2024c) introduce dedicated video encoders specifically designed for video representation, while others (Xu et al., 2024a,b; Maaz et al., 2024; Li et al., 2024a) reuse the image encoders and apply spatial or temporal pooling to video frame features without adding extra training parameters. Additionally, some research works aim to reduce the number of tokens required for video inputs. For example, some methods (Li et al., 2024c; Chen et al., 2024b) compress video clips at extreme ratios, such as reducing 4 or 8 frames to 1–32 tokens. However, they may result in significant loss of detailed information, which undermines their performance in fine-grained video QA tasks such as VideoMME (Fu et al., 2024). Aiming to achieve strong performance across a variety of Video QA benchmarks with different video lengths and question types, models must make trade-offs between video token compression rates and the preservation of various detailed information. While high compression rates enable the model to accept longer inputs, which benefits long videos, the information loss introduced by such high compression often degrades QA performance on short videos.

Another critical issue we find is the lack of a fair evaluation setting of video understanding tasks. Many results in current video QA benchmarks fail to report the number of tokens used by the video,

which can lead to unfair comparisons. This is because, for a given number of input frames, the more vision tokens used, the better the QA performance typically becomes. Even for the same model, increasing the number of input frames within certain limits can lead to significant performance improvements.

In this work, we aim to study effective modeling and fair evaluation that can be applied to both short and long video inputs. For effective modeling, we propose a more effective and efficient approach by employing an image-based vision encoder combined with a TimePerceiver module to represent video content, allowing video compact learning within the VLM framework. For fair evaluation, our goal is to provide the video QA benchmark results within a fixed vision token upper bound. These results not only advocate for fairer comparisons, but also offer more relevant insights for practical applications, where a balance between computational cost and accuracy must be achieved.

The contributions of this work are as follows:

- We propose **Clapper**, a competitive VLM for video understanding, which is capable of achieving strong performance using less than one third of the tokens required by previous state-of-the-art models.
- We introduce a novel video representation method that leverages a slow-fast strategy and the TimePerceiver module, enabling efficient and compact learning for VLMs.
- We explored the performance of current VLMs under a unified upper bound on video tokens, providing greater practical value for the application of video VLMs in real-world scenarios.

2 Related Works

2.1 Video Representation

For video inputs in VLM, frames are typically sampled at a fixed frame rate, such as 1 fps, or a fixed number of frames are force-sampled and fed into the model. The VLMs are then revised to embed the video frames, producing an embedding that represents the video segment. For instance, InternVideo2 (Wang et al., 2024c) accepts a fixed input of 8 frames and outputs an embedding of size $C \times L = 3200 \times 2048$, C is the channel dimension, and L is the token length.

Some other models process frames individually through the image encoder, such as CLIP (Radford et al., 2021) or SigLIP (Zhai et al., 2023), to obtain their image embeddings. Ultimately, these visual inputs are then either passed through an MLP and directly concatenated with the text query, or processed using a Q-former structure, where an LLM performs video understanding tasks (Li et al., 2023; Zhang et al., 2023; Li et al., 2024c). If the Q-former structure is used, we need to recalculate visual inputs for each query, which is inefficient for multi-turn dialogue. Direct concatenation of visual tokens is more simple, but it can result in excessively long sequences, which limit the performance and efficiency of the LLM. Consequently, extensive research has focused on compressing visual tokens, and will be discussed in the following section.

2.2 Token Compression

Visual token compression have rapid advancements in VLMs with image inputs. For example, high-resolution images are essential for fine-grained perception tasks such as OCR, object grounding, and detailed information-based QA. To address the demands of high-resolution image inputs, dynamic high-resolution techniques have been proposed, which divide the high-resolution image into multiple small patches (referred to as "tails"). However, the increase in the number of tokens due to these multiple tails results in a rapid rise in memory usage and latency, which limits the practical applicability of VLMs. In response to the above issue, several models have introduced techniques such as pixel shuffle (Chen et al., 2024d; Dai et al., 2024), spatial pooling (Li et al., 2024a; Zhang et al., 2024c), or bilinear interpolation (Li et al., 2024a), to reduce the number of tokens generated by these tails, typically achieving a reduction of approximately a quarter.

For video inputs, there are also some works that reduce the number of visual tokens of multiple frames from videos. PLLaVA (Xu et al., 2024a) explored the effects of average pooling at varying strides in both spatial and temporal dimensions. Their experiments on MVBench (Li et al., 2024b) and VCGBench (Maaz et al., 2024) revealed that while 50% spatial downsampling maintains performance levels, temporal pooling or more aggressive spatial compression leads to notable performance degradation. To mitigate resolution loss for critical frames, SlowFast (Xu et al., 2024b; Zhang et al., 2024f) adopts an asymmetric strategy where key frames undergo spatial downsampling by a factor

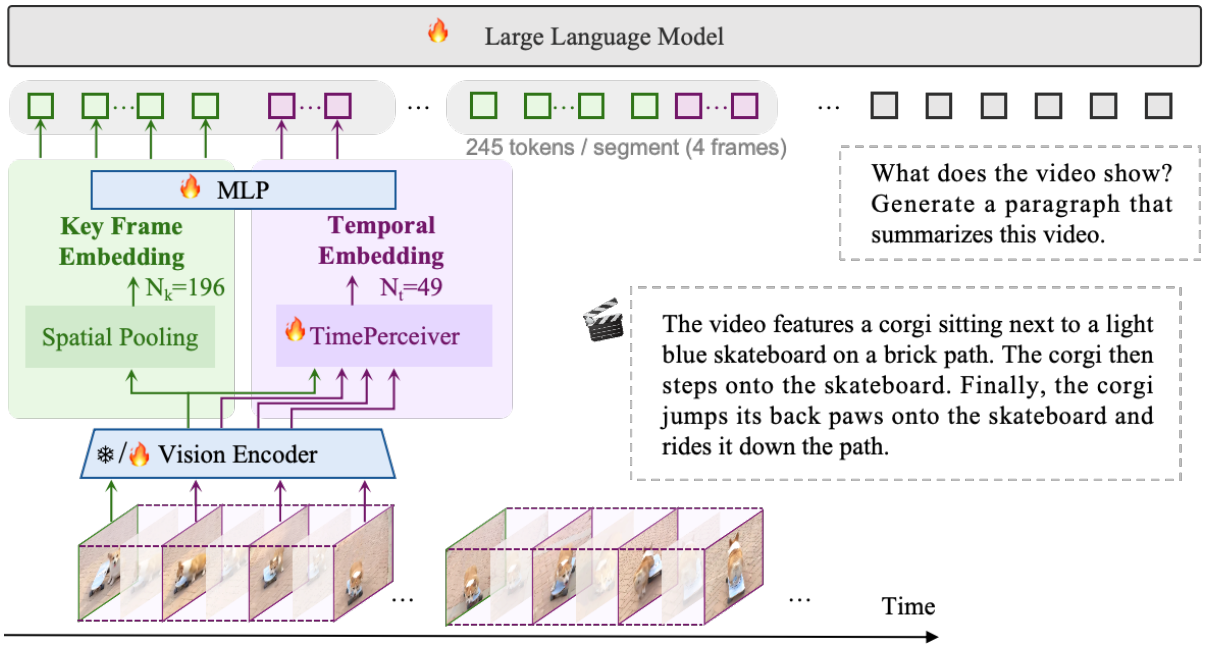

Figure 1: Architecture of Clapper. The model consists of a vision encoder, a TimePerceiver module, an MLP layer, and an LLM. Input videos are sampled and divided into segments, with each segment represented by a combination of high-resolution keyframe embedding and compressed temporal embedding.

of 2 (slow pathway) while other frames use downsampling by a factor of 4 (fast pathway). Meanwhile, alternative architectures have explored learnable compression mechanisms: MiniCPM (Yao et al., 2024) and InternVideo2-HD (Wang et al., 2024c) implement Perceiver-style cross-attention to project each frame's features into fixed-length tokens. However, experimental results indicate that these methods still suffer from non-negligible performance degradation and face challenges in further reducing token counts. LLaMA-VID (Li et al., 2024c) achieves an exceptional compression by using context-aware techniques, where text queries and visual embeddings interact through cross-modal attention to produce two adaptive tokens per frame. However, this approach may not be optimal for all tasks, such as video captioning.

3 Method

3.1 Architecture

We introduce Clapper, a VLM for video understanding that achieves compressed video representation and aligns it with an LLM backbone. As shown in Figure 1, unlike approaches that require extensive data resources to train a video foundation model from scratch or those that use image foundation models in a training-free manner, we adapt a pretrained image foundation model as the base.

Specifically, we employ SigLIP-448px/16 (Zhai et al., 2023) as the vision encoder, which outputs a 784-dimensional embedding for each input image. For input videos, during training, we sample the video at 1 fps, resulting in a sequence of frames $I \in \mathbb{R}^{(T\times448\times448\times3)}$ with the length T. We sequentially divide the video into short segments, each consisting of 4 consecutive frames. The first frame of each segment is designated as the key frame, and its embedding is obtained by applying a spatial pooling with a stride of 2 to the original 784-dimensional embedding, resulting in a 196-dimensional representation. This key frame is crucial for preserving detailed spatial information, such as the attributes of the scene and the main subjects. It ensures that important visual details including the layout of the scene, the appearance of objects, and the positions of characters, are retained in the video representation.

We then introduce a trainable component, TimePerceiver, to effectively learn the temporal information within each segment. TimePerceiver takes 4 frames as input and outputs a 49-dimensional embedding representation. Ultimately, the video segment is represented by combining high-resolution keyframe information with highly compressed temporal information. The keyframe captures the detailed spatial attributes, while TimePerceiver focuses on temporal dynamics, ensuring that both aspects are adequately rep-

resented in the final video embedding. Each video segment occupies a total of 245 tokens.

For the last video segment that may not have 4 frames, if it contains only one frame, it will have 196 tokens corresponding to the key frame. If it contains two or three frames, it still occupies 245 tokens. The visual tokens obtained from all processed video segments are concatenated in sequence and combined with the query to serve as the input to the LLM. In our experiments, we use Qwen2 (Yang et al., 2024) as our LLM backbone.

The architecture of TimePerceiver is depicted in Figure 2. This module accepts a sequence of T image features (in our approach, it typically receives 4 frames as input, though occasionally it may receive 2 to 3 frames) from the vision encoder, and it generates a fixed number of visual outputs (set to 49 in our method). For the input visual features $X_f \in \mathbb{R}^{(T\times L\times D)}$, we first apply spatial average pooling with a stride of 4, resulting in $X_s \in \mathbb{R}^{(T\times L/16\times D)}$. Subsequently, we perform average pooling along the temporal dimension to obtain $X \in \mathbb{R}^{(1\times L/16\times D)}$. L and D represents the length and dimension of the visual features. In our method, L is 784 and D is 1152. By using X as the input queries and cross-attending to the flattened visual features X_f, the model can focus on regions that change across the frames. Following the approach of Flamingo (Alayrac et al., 2022), the keys and values are computed from the concatenation of X and X_f. The number of output tokens from TimePerceiver is equal to the number of input queries. Our ablation studies demonstrate that employing such a TimePerceiver module yields superior performance compared to a standard Perceiver (Alayrac et al., 2022).

3.2 Training Recipe

The training of Clapper is divided into two stages. In the first stage, TimePerceiver is trained to better model temporal motion information using video-caption pairs. During this stage, video-caption pair data is used for training. We first selected 178k video-caption pairs from LLaVA-Video (Zhang et al., 2024f). These unedited videos, ranging from a few seconds to 3 minutes in length, offer detailed captions that describe various aspects of the content, capturing rich variations. Additionally, we used OpenVid-1M (Nan et al., 2024), a dataset for video generation. Videos in this dataset are trimmed to ensure scene consistency, and the captions are relatively concise.

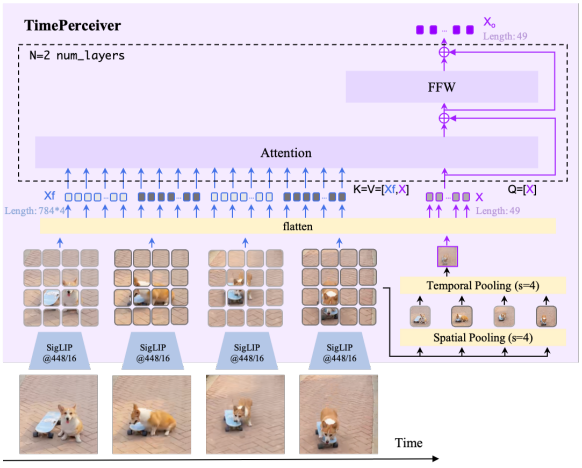

Figure 2: The `TimePerceiver` module processes 2-4 frames to generate a fixed number of temporal embedding outputs (49 in this paper).

In the second stage, we employ video instruction-tuning data to train the model's instruction-following capabilities. The training data includes LLaVA-Video-178K (Zhang et al., 2024f), ActivityNet-QA (Yu et al., 2019), NExT-QA (Xiao et al., 2021), PerceptionTest (Patraucean et al., 2024), and LLaVA-Hound-255K (Zhang et al., 2024d), which together provide a total of 1.6 million video-language samples.

4 Experiments

We conducted evaluations across all benchmarks using LMMs-Eval (Zhang et al., 2024a) to ensure standardization and reproducibility. To fairly compare with other leading VLMs that support video understanding, we primarily reused results from original papers. When results were not available, we integrated the models into LMMs-Eval and assessed them under consistent settings.

4.1 Implementation Details

We use LLaVA-Onevision-SI (Li et al., 2024a) with 7B parameters as our baseline model and finetune it for two stages. In the first stage, we freeze the weights of vision encoder and optimize the other parameters include the MLP layers, the TimePerceiver module and the language model. In the second stage, we finetune all parameters of the model on video instruction pairs. We train the model with a batch size of 128 for only 1 epoch in both two stages. The start learning rate is set to 2e-5. During training, videos are sampled at 1 fps, and for videos exceeding 96 seconds, we sample a fixed number of 96 frames at equal intervals.

4.2 Overall Results

We report the accuracy of Clapper on six popular and challenging multiple-choice video QA benchmarks. Compared to open-ended questions, multiple-choice questions are easier to evaluate and yield more stable results that are less influenced by the specific evaluation model. The benchmarks are introduced in the order of video duration, starting with TempCompass (Liu et al., 2024b), MVBench (Li et al., 2024b) and PerceptionTest (Patraucean et al., 2024), which focus on short videos. LongVideoBench (Wu et al., 2024) and MLVU (Zhou et al., 2024) focus on long video understanding. Finally, VideoMME (Fu et al., 2024) covers a broad range of video understanding tasks. Most results are from original papers or benchmark leaderboards.

Overall comparison. Table 1 shows the performance of Clapper compared to a wide range of VLMs that have leading performance on these benchmarks, including both proprietary models and open-source models within the 7B-9B parameter range. We explicitly display the number of frames used during evaluation and the average number of tokens per frame in the table. The open-source models are categorized based on their compression ratio into three groups: compression ratio $\leq$ 4x, 4x $<$ compression ratio $\leq$ 16x, and extreme compression ratio > 16x. The compression ratio (CR) is mathematically defined as:

$$CR = \frac{N_{\text{original}}}{N_{\text{compressed}}} \quad (1)$$

where N_{original} denotes the number of visual tokens generated by the vision encoder, and $N_{\text{compressed}}$ represents the number of compressed visual tokens actually used by the model. This metric quantifies the degree of token reduction achieved during the compression process, with higher values indicating more aggressive compression.

Our proposed Clapper achieves competitive results in the compression ratio > 4x category, obtaining the best performance on four benchmarks: TempCompass, PerceptionTest, MLVU and VideoMME. Specifically, Clapper achieves 67.4% on TempCompass, 66.5% on PerceptionTest, 69.8% on MLVU and 62.0% on VideoMME without subtitles. Compared to other models with similar performance, Clapper maintains a lower visual token overhead.

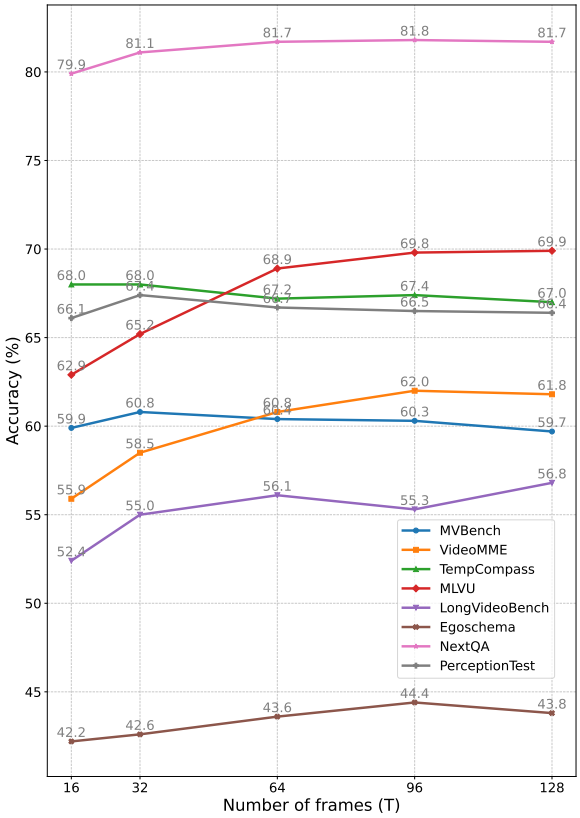

Figure 3: Performance of Clapper under different frames on the eight video QA benchmarks.

We also conducted a qualitative analysis of the model performance. As illustrated in Figure 4, we compare the video captioning results of our proposed Clapper method with those of advanced models, including LLaVA-Video (Zhang et al., 2024f), MiniCPM-V 2.6 (Yao et al., 2024), and InternVL2 (Chen et al., 2024d). The figure also presents the visual tokens utilized by each method. Notably, Clapper is able to capture key points and details that are absent in the outputs of other models, as highlighted in bold green text. This demonstrates Clapper's superior ability to extract and summarize dynamic and detailed information, even with the minimal use of visual tokens.

Performance under different frames. To explore how Clapper performs under varying visual token upper bounds, we adjusted the number of test frames for each video and observed the corresponding changes in performance across eight benchmarks. As illustrated in Figure 3, Clapper shows a steady improvement in MLVU performance as the number of input frames increases. On VideoMME, NextQA, LongVideoBench, and Egoschema, the overall trend also rises with in-

Model	Size	#Tokens./ Frame	#Frames	#Tokens	TempCompass M-Avg	MVBench Avg	PerceptionTest Val	LongVideoBench Val	MLVU M-Avg	VideoMME Overall
Avg. Duration					13s	16s	23s	473s	651s	1010s
Proprietary Models										
GPT4-V (OpenAI, 2023)	-	-	1fps	-	-	43.7	-	59.1	49.2	59.9/63.3
GPT4-o (OpenAI, 2024)	-	-	1fps	-	71.0	64.6	-	66.7	64.6	71.9/77.2
Gemini-1.5-Pro (Reid et al., 2024)	-	-	1fps	-	67.1	60.5	-	64.0	-	75.0/81.3
Compression Ratio ≤ 4x										
IXComposer-2.5 (Zhang et al., 2024b)	7B	400	[32,64]	26k	61.3	69.1	34.4	-	58.8	55.8/58.8
InternVL2 (Chen et al., 2024d)	8B	256	[8,16]	4k	65.6	65.8	-	54.6	64.0	54.0/56.9
InternVL2.5 (Chen et al., 2024c)	8B	256	[16,32,48,64]	16k	-	**72.0**	-	**60.0**	68.9	**64.2**/66.9
Kangaroo (Liu et al., 2024a)	8B	256	64	16k	62.5	61.1	-	54.8	61.0	56.0 / 57.6
LongVILA (Xue et al., 2024)	7B	196	256	50k	-	67.1	58.1	57.1	-	60.1/65.1
LLaVA-Video (Zhang et al., 2024f)	7B	196	64	13k	67.3	58.6	**67.9**	58.2	70.8	63.3/**69.7**
LLaVA-OneVision (Li et al., 2024a)	7B	196	32	6k	64.8	56.7	57.1	56.3	64.7	58.2/61.5
LLaVA-NeXT-Video (Zhang et al., 2024e)	7B	144	32	5k	50.6	53.1	48.8	49.1	-	- /46.5
LongVA (Zhang et al., 2024c)	7B	144	[32,128]	18k	56.1	-	-	-	56.3	52.6/54.3
LongLLaVA (Wang et al., 2024b)	9B	144	128	18k	-	49.1	-	-	-	43.7/ -
Qwen2-VL (Wang et al., 2024a)	7B	Dyn	1fps	-	**68.5**	67.0	62.3	-	-	63.3/69.0
4x < Compression Ratio ≤ 16x										
MiniCPM-V 2.6 (Yao et al., 2024)	8B	96	64	6k	66.3	-	-	54.9	-	60.9/63.7
VideoLLaMA2 (Cheng et al., 2024)	7B	72	16	1k	-	54.6	51.4	-	48.5	47.9/50.3
VideoChat2-HD (Li et al., 2024b)	7B	72	16	1k	51.1	62.3	-	-	47.9	45.3/55.7
InternVideo2-HD (Wang et al., 2024c)	8B	72	16	1k	-	67.2	63.4	-	-	49.4/ -
LongVU (Shen et al., 2024)	7B	64	1fps	-	-	66.9	-	-	65.4	- /60.6
Clapper (Ours)	7B	61	96	6k	67.4	60.3	66.5	55.6	69.8	62.0/64.6
Clapper (Ours)	7B	61	1fps	-	67.3	59.6	67.1	55.3	**72.0**	62.7/64.0
Extreme Compression Ratio										
LLaMA-VID (Li et al., 2024c)	7B	2	1fps	-	38.0	41.9	44.6	-	33.2	25.9/ -

Table 1: **Main results on multiple-choice video QA benchmarks.** M-Avg refers to the average score of multiple-choice QA tasks within each benchmark. The average tokens per frame and evaluation frames are shown for direct comparison. In "#Tokens./frame", Dyn denotes naive dynamic resolution. In "#Frames", brackets indicate that scores are tested across multiple frame settings and reported as the highest value. The best and second-best results in open-source models are **bolded** and underlined, respectively.

creasing frames, though with some fluctuations. In contrast, performance in TempCompass, PerceptionTest, and MVBench shows little correlation with the number of input frames. These trends are highly correlated with the average video duration of each benchmark. When the average video length is less than the number of test frames, their performance does not benefit significantly from additional frames. In contrast, when the average video length is several times longer than the number of test frames, the models show a more substantial performance gain with increased frame sampling.

Performance under the same visual token upper bounds. To understand the trade-off between model accuracy and computational cost for practical deployment, we analyzed model performance by adjusting the number of input video frames to ensure all models operate under the same visual token upper bounds. Specifically, we set two token limits: 2k and 6k tokens. The number of input frames for each model was calculated by dividing these token limits by each model's tokens-per-frame number (#Tokens/frame), as detailed in the #Frames column in Table 2.

We selected VideoMME to compare different models under the same Visual Token Upper Bounds by varying the number of evaluation frames because it includes videos of varying lengths, categorized into short, medium, and long for evaluation. This facilitates our experimental observations. The results are shown in Table 2. As can be seen, our proposed Clapper achieves the best performance within the 2k and 6k visual token limit, particularly excelling in short and medium-length videos. This highlights Clapper's ability to efficiently capture spatiotemporal information within limited token budgets. However, there is still significant room for improvement in understanding videos longer than 30 minutes within the 6k visual token limit. In the future, we aim to extend Clapper's strong performance to longer video durations.

4.3 Further Analysis

To reduce experimental overhead, all models in the analysis section were trained using one-tenth of the full dataset used in Stage 2, which amounts to approximately 160k samples. This reduced dataset size allows for more efficient experiments while still providing sufficient data to evaluate the effectiveness of different components and strategies in our model design.

Model	#Frames	VideoMME		
		Short	Medium	Long
Visual Token Under 2k				
InternVideo2-HD	32	53.7	43.0	40.1
LLaVA-Video	10	67.8	53.1	47.0
Clapper (Ours)	32	70.1	55.3	50.1
Visual Token Under 6k				
MiniCPM-V 2.6	64	71.3	59.4	51.8
LLaVA-Video	32	72.4	58.6	50.6
Clapper (Ours)	96	74.7	60.1	51.2

Table 2: Detailed results on VideoMME (wo) with two fixed visual token upper bounds. Video lengths: Short (0–2 minutes), Medium (4–15 minutes), and Long (30–60 minutes).

Method	CR	MVBench	TempCompass	VideoMME
Baseline	4x	55.4	63.1	59.1
Temporal Pooling	16x	56.3 (+0.9)	64.7 (+1.6)	56.9 (-2.2)
Spatial Pooling	16x	55.1 (-0.3)	64.8 (+1.7)	58.4 (-0.7)
Perceiver	13x	56.5 (+1.1)	64.2 (+1.1)	56.5 (-2.6)
TimePerceiver	13x	57.2 (+1.8)	65.5 (+2.4)	59.3 (+0.2)

Table 3: Comparison of video token compression strategies. CR stands for compression ratio. The performance differences show accuracy variations from the baseline.

Analysis of video token compression. To identify the optimal compression strategy that maintains the model performance while significantly reducing the token count, we conducted comprehensive experiments comparing different compression approaches for ablation. The Baseline method achieves a 4x compression of visual tokens by applying spatial pooling with a stride of 2 to all input frames. When using SigLIP@448px/16 as the vision encoder, the average number of tokens per frame in baseline is 196. To further reduce the token count from 4x to 16x compression, we compared four different compression strategies:

1. Temporal Pooling: Extend the baseline by incorporating temporal pooling with a stride of 4 along the temporal dimension.

2. Spatial Pooling: Build upon the baseline by applying an additional spatial pooling layer with a stride of 2.

3. Perceiver: Utilize a Perceiver architecture (Alayrac et al., 2022) with 49 randomly initialized learnable queries to aggregate information from every 4 frames.

4. TimePerceiver: Our method to integrate information from every 4 frames into 49 tokens.

In Perceiver and TimePerceiver, the first frame of every 4 frames is treated as a key frame and compressed using the baseline method. It is then concatenated with the generated 49 tokens to represent these 4 frames. Table 3 shows the results. We evaluated these strategies on three video QA datasets with input frames uniformly sampled to 64. Temporal and Spatial Pooling achieve a 16x compression ratio, using 3k tokens per video, while Perceiver and TimePerceiver achieve a 13x compression ratio, using 4k tokens per video.

In benchmarks less sensitive to input frame count, like MVBench and TempCompass, methods that primarily compress temporal information, such as Temporal Pooling and Perceiver, perform better. Specifically, Temporal Pooling achieves a +0.9% improvement on MVBench and a +1.6% improvement on TempCompass. Perceiver improves by +1.1% on both. However, on VideoMME, which is more frame-sensitive, both Temporal Pooling and Perceiver suffer significant performance drops, indicating substantial loss of temporal information. In contrast, Spatial Pooling achieves a +1.7% improvement on TempCompass but shows a -0.3% drop on MVBench and a -0.7% drop on VideoMME. This suggests that MVBench and VideoMME may contain more fine-grained understanding questions, while TempCompass has lower requirements for spatial resolution.

Our proposed TimePerceiver method performs well across all three benchmarks, demonstrating effective compression in both spatial and temporal dimensions. Compared to the baseline with a 4x compression ratio, TimePerceiver achieves a 13x compression ratio while maintaining stable or even improved overall performance.

Impact of training designs. Experiments in this section were designed to evaluate how each training stage impacts the model performance, particularly the integration of the TimePerceiver module. We compared three setups: (1) a baseline model without TimePerceiver, (2) add TimePerceiver with direct training in the fine-tuning stage, and (3) add TimePerceiver with the full two-stage training strategy, stage 1 warm-up followed by stage 2 fine-tuning.

Results in Table 4 demonstrate that the inclusion of the TimePerceiver module, even with direct training, provides modest improvements over the baseline, with gains of +0.5 on MVBench and +0.7 on TempCompass. However, this setup shows a

Instruction: Give a detailed description of the video. **7s video**

Clapper (Ours): The video captures a vibrant and festive scene of a traditional dragon dance performance. The setting is a crowded outdoor area, filled with people who are enthusiastically watching the event. **Many in the crowd are wearing traditional conical hats**, adding to the cultural ambiance. The background is illuminated by bright lights, enhancing the celebratory atmosphere. **A large, elaborately decorated dragon figure, adorned with red and yellow colors, moves through the crowd. The dragon's head is particularly detailed, with intricate designs and glowing eyes, captivating the onlookers.** The crowd is actively engaged, with **many individuals holding up their phones to capture the moment**. The air is filled with smoke and sparks from the fireworks or firecrackers being used as part of the performance. The overall mood is one of excitement and celebration, with the dragon dance being the focal point of the event.

8 frames; 490 visual tokens

LLaVA-Video: The video captures a vibrant scene of a crowd gathered in what appears to be an outdoor setting, **possibly during a festival or celebration**. The atmosphere is filled with excitement and energy, as evidenced by the **people's raised hands and the use of mobile phones to capture the moment**. The crowd is diverse, with individuals wearing various types of hats, **some of which are traditional conical hats**, suggesting a cultural event. The lighting is low, with the primary light sources being the fireworks and the ambient light from the surroundings. The fireworks display includes bursts of light and smoke, creating a dynamic and colorful spectacle. The smoke from the fireworks creates a hazy atmosphere, partially obscuring the view of the crowd and the background.

8 frames; 1568 visual tokens

MiniCPM-V 2.6: The video captures a vibrant and chaotic scene at what appears to be a cultural or festive event. A crowd of people, **many wearing hats with Chinese characters on them**, are gathered closely together amidst thick smoke and sparks flying in the air. The atmosphere is intense as bright lights illuminate parts of the scene, adding to the sense of excitement and energy. **Some individuals can be seen holding up their phones, possibly recording the event or taking photos.** In the background, there is a large structure that looks like it might be part of a traditional performance or display. As the sequence progresses, more sparks become visible, enhancing the dynamic nature of the scene. Despite the smoky conditions making visibility slightly difficult, the overall mood remains one of celebration and festivity.

8 frames; 768 visual tokens

InternVL2: The video depicts a lively and energetic scene set in what appears to be a crowded outdoor event, possibly a festival or concert. The atmosphere is vibrant, with a mix of bright lights and colorful smoke filling the air, creating a visually stimulating environment. The crowd is dense, with numerous people gathered closely together, all facing towards a central point of interest. In the foreground, **several individuals are holding up their smartphones, capturing the moment on camera.** Their faces are partially obscured by the smoke, but their intent is clear as they document the event. The smoke adds a dynamic element to the scene, creating a hazy, dreamlike quality that enhances the overall ambiance. The background is filled with various bright lights, including red and yellow hues, which illuminate the scene and add to the festive atmosphere. The lights are strategically placed, creating a visually appealing backdrop that contrasts with the darker tones of the smoke. The smoke itself is a mix of white and gray, swirling and dancing in the air, adding a sense of movement and energy to the scene.
As the video progresses, the smoke becomes more pronounced, enveloping the crowd and creating a sense of unity among the attendees. The lights continue to flicker and change, casting different colors and patterns across the scene. The crowd remains engaged, with some individuals raising their hands in the air, possibly in celebration or excitement. **In one notable moment, a person in the crowd is seen holding up a large, colorful object, possibly a flag or a piece of clothing**, which adds to the visual spectacle. The object is brightly colored, standing out against the darker tones of the smoke and the crowd. The video captures the essence of a lively event, with the crowd fully immersed in the experience. The combination of bright lights, colorful smoke, and the enthusiastic crowd creates a dynamic and engaging scene that is both visually stunning and emotionally resonant. The use of smartphones to document the event further emphasizes the importance of capturing these moments, highlighting the significance of the occasion for those in attendance.

8 frames; 2048 visual tokens

Figure 4: Comparison of video captioning results using Clapper and others. Key points are displayed in bold. Details uniquely captured by Clapper, which are absent in other models, are highlighted in bold green.

Stage1	Stage2	MVBench	TempCompass	VideoMME
Baseline		55.4	63.1	59.1
✗	✓	55.9 (+0.5)	63.8 (+0.7)	58.3 (-0.8)
✓	✓	57.2 (+1.8)	65.5 (+2.4)	59.3 (+0.2)

Table 4: Ablation study on different training designs.

slight degradation of -0.8 on VideoMME, suggesting that direct training may not fully leverage the module's potential for video understanding. In contrast, the two-stage training strategy yields more significant improvements across all benchmarks, achieving +1.8 on MVBench, +2.4 on TempCompass, and +0.2 on VideoMME. These findings indicate that the two-stage training approach allows the TimePerceiver module to be more effectively trained for temporal information, leading to better overall performance on video understanding tasks. The warmup stage likely helps the model initialize and stabilize its learning of temporal features, while the fine-tuning stage refines these features for optimal performance. This highlights the importance of a carefully designed training strategy for integrating complex modules like TimePerceiver into video understanding models.

5 Conclusion

In this work, we proposed Clapper, an efficient video language model that demonstrates competitive performance across various benchmarks. Through the introduction of our TimePerceiver module, we successfully increased the compression ratio from 4x to 13x while preserving essential temporal and spatial information. This advancement allows Clapper to balance computational efficiency and accuracy, enabling faster inference and lower memory requirements. In the future, we aim to explore methods for further optimization of token representations to push compression boundaries even further. Additionally, we plan to extend our architecture's temporal modeling capacity to handle longer visual contexts.

6 Limitations

Clapper was trained at 1fps with a maximum of 96 frames, without additional training for length extrapolation. As a result, it is constrained by length extrapolation and the context length of large language models. For videos longer than 5 minutes, the model's performance may degrade as the video length increases. Additionally, Clapper has not undergone Reinforcement Learning from Human Feedback alignment training, which may lead to the generation of hallucinated outputs.

References

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

Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2024a. Onechart: Purify the chart structural extraction via one auxiliary token. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 147–155.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024b. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418.

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

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2024d. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *CoRR*, abs/2404.16821.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *CoRR*, abs/2406.07476.

Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nvlm: Open frontier-class multimodal llms. *arXiv preprint*.

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

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, pages 22195–22206. IEEE.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024c. Llama-vid: An image is worth 2 tokens in large language models.

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

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

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. 2024b. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd*

Annual Meeting of the Association for Computational Linguistics (ACL 2024).

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. 2024. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024. Gpt-4o. https://openai.com/index/hello-gpt-4o/.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. 2024. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

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

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

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. 2024b. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture. *CoRR*, abs/2409.02889.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. 2024c. Internvideo2: Scaling video foundation models for multimodal video understanding. In *ECCV*.

Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2025. Vary: Scaling up the vision vocabulary for large vision-language model. In *Computer Vision – ECCV 2024*, pages 408–424, Cham. Springer Nature Switzerland.

Haoning Wu, DONGXU LI, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pages 28828–28857. Curran Associates, Inc.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786.

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

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. 2024b. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv:2407.15841*.

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

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

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

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. 2023. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via

question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, Singapore. Association for Computational Linguistics.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*.

Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, Songyang Zhang, Wenwei Zhang, Yining Li, Yang Gao, Peng Sun, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Hang Yan, Conghui He, Xingcheng Zhang, Kai Chen, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024b. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *CoRR*, abs/2407.03320.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024c. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

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

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

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

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