MotionBench: Benchmarking and Improving Fine-grained Video Motion Understanding for Vision Language Models

Wenyi Hong[1*] Yean Cheng[2*] Zhuoyi Yang[1*] Weihan Wang[2] Lefan Wang[2]
Xiaotao Gu[2] Shiyu Huang[2] Yuxiao Dong[1†] Jie Tang[1†]
[1]Tsinghua University [2]Zhipu AI
wenyi.hong@outlook.com, cya17@tsinghua.org.cn,
zhuoyiyang2000@gmail.com, jietang@tsinghua.edu.cn

Abstract

In recent years, vision language models (VLMs) have made significant advancements in video understanding. However, a crucial capability — fine-grained motion comprehension — remains under-explored in current benchmarks. To address this gap, we propose MotionBench, a comprehensive evaluation benchmark designed to assess the fine-grained motion comprehension of video understanding models. MotionBench evaluates models' motion-level perception through six primary categories of motion-oriented question types and includes data collected from diverse sources, ensuring a broad representation of real-world video content. Experimental results reveal that existing VLMs perform poorly in understanding fine-grained motions. To enhance VLM's ability to perceive fine-grained motion within a limited sequence length of LLM, we conduct extensive experiments reviewing VLM architectures optimized for video feature compression and propose a novel and efficient Through-Encoder (TE) Fusion method. Experiments show that higher frame rate inputs and TE Fusion yield improvements in motion understanding, yet there is still substantial room for enhancement. Our benchmark aims to guide and motivate the development of more capable video understanding models, emphasizing the importance of fine-grained motion comprehension. Project page: https://motion-bench.github.io.

1. Introduction

With the rapid development of pre-training, an increasing number of studies focus on leveraging large vision language models (VLMs) for video understanding [15, 19, 27, 29, 34]. For instance, CogVLM2-Video [15], LLaVA-Video [51] and PLLaVA [44] continually train image-understanding models to achieve video-understanding models, and Qwen2-VL[37], LLaVA-OneVision [18] explore mixed training upon both images and videos. To effectively evaluate video understanding VLMs as well as guide further advancement, a series of video understanding benchmarks emerged, with focuses on general video understanding capability [8, 23, 24, 39] or specific capabilities such as long video understanding [39, 41, 54]. Video understanding questions can be categorized into three levels based on the

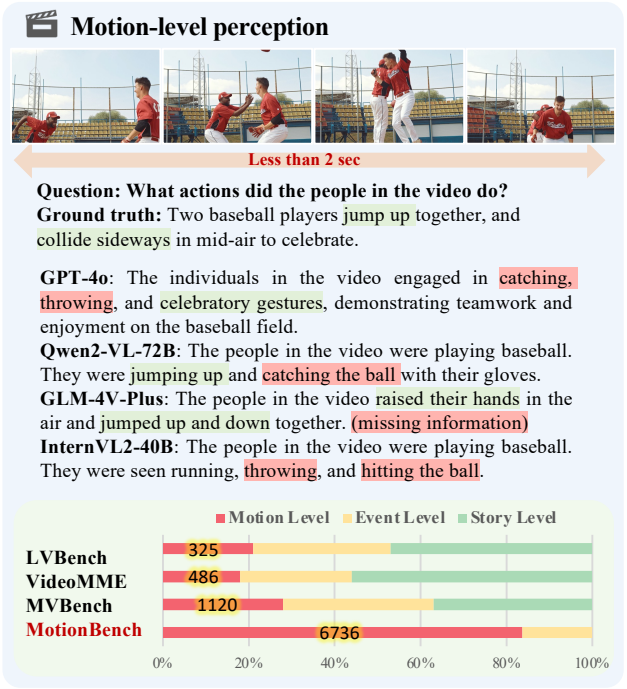


Figure 1. State-of-the-art video understanding models struggle with basic motion-level perception. Compared to existing benchmarks, our proposed MotionBench focuses on assessing the model's Motion level perception capability, which is critical in understanding videos with fast and instant interactions and motions.

*Equal contribution.
†Corresponding authors
Work was done when WH, ZY interned at Zhipu AI.

granularity of understanding: *motion-level* (capturing fine-grained motion), *event-level* (addresses distinct segments of activities [7]), and *story-level* (a holistic understanding of the storyline across the video [9]). Among them, motion-level understanding acts as a foundational ability and plays a pivotal role in applications such as anomaly detection, open-domain action analysis, detailed video captioning, *etc*. However, while some benchmarks shifted their focus toward *event-* and *story-level* understanding, most benchmarks lack a dedicated set for evaluating *motion-level* understanding. To quantitatively analyze the granularity distribution across benchmarks, we leverage GPT-4o[1] for question analysis. The results in Figure 1 indicate that the foundational motion-level comprehension is being overlooked, with the data volume and diversity for *motion-level* content being limited. Some datasets from earlier years focused on *low-level* action recognition within specific domains, but their content and categories are highly constrained.

Is this because *motion-level* understanding is too trivial to merit attention? To answer this question, we build MotionBench to thoroughly evaluate the *motion-level* capability of current video models. MotionBench comprises 8,052 questions covering six main categories of video motion, with diverse video collected from the web (Panda-70M [3], Pexels[2]), public datasets (MedVid [13], SportsSloMo [2], Ha-ViD [53]), and self-synthetic videos generated via Unity[3], capturing a broad distribution of real-world application. Surprisingly, most state-of-the-art models can only achieve accuracy lower than 60%, significantly below the threshold for practical applications, which highlights two primary technical challenges:

High Frame Rate vs. Computational Cost: The first challenge lies in the contradiction between the high frame rate required for fine-grained motion understanding and the high computational cost of long sequence lengths. Long sequence lengths substantially increase the computational and memory burden in both training and inference. Consequently, most current video understanding models can only handle a limited number of frames, falling short of the demands for fine-grained motion analysis. For example, Intern-VL2 [5], LLaVA-Next-Video [50] and CogVLM2-Video [15] can only accept 16 to 64 frames, thus can only sample frames at an extreme-low rate of 1 frame every 5 seconds (*i.e*., 0.2 fps) for a 5-minute video which is common in daily life. To address this, we **conduct the first comprehensive evaluation** over existing video feature compression architectures and identify their common shortcomings-shallow fusion. Based on these findings, **we propose a novel VLM architectural paradigm—Through-Encoder Fusion** (TE Fusion), which enhances video feature representation under a fixed decoder sequence length by applying deep fusion throughout the visual encoder. Experiments on benchmarks across various video lengths and contents demonstrate that TE Fusion achieves state-of-the-art performance, and shows particular advantages under high compression ratios.

Limited Fine-Grained Motion Understanding: The second challenge arises from the limited foundational capability to comprehend fine-grained motion in current video understanding models. While a higher frame rate brings some performance improvements (Tab. 4), models' *motion-level* understanding remains constrained, achieving accuracies of below 60% on MotionBench (Tab. 3). To address this, **we additionally release a dataset of 5,000 videos with manually annotated fine-grained motion descriptions**, which are annotated and double-checked together with the benchmark annotation process (refer to Fig. 4 for example). Each video includes dynamic information descriptions with annotation density reaching 12.63 words per second, providing researchers with resources for further development and training to enhance video models' *motion-level* comprehension capabilities.

Contribution. Our main contributions include:

- We introduce MotionBench, the largest *motion-level* video benchmark, featuring a wide range of video sources and question types, along with a carefully designed annotation pipeline that ensures diversity and accuracy.
- MotionBench reveals a critical deficiency in *motion-level* understanding among current video understanding models, which is largely overlooked by existing research.
- We propose TE Fusion, a novel compression architecture to enhance *motion-level* understanding under constrained LLM context length. Experimental results demonstrate that TE Fusion achieves state-of-the-art results on MotionBench and outperforms other compression methods across MotionBench, MVBench [23], LVBench [39], and VideoMME [8] in the ablation study, and shows a particular advantage in high compression ratio scenarios.

2. Related Work

2.1. Video Understanding Benchmarks

To effectively evaluate video understanding models and drive their advancement, a series of benchmarks are proposed. Traditional benchmarks like MSRVTT-QA [43] and ActivityNet-QA [48] primarily focus on basic action recognition and video question answering with short clips. While these benchmarks provide a foundation for assessing video understanding capabilities, they lack the granularity to evaluate subtle motion comprehension. Recently, more benchmarks emerged to assess video VLMs, as shown in Tab. 1. MVBench [23] emphasizes general video understanding, introducing 20 temporal-related tasks across

[1] gpt-4o-2024-08-06
[2] https://www.pexels.com
[3] https://unity.com/cn

Table 1. The comparison of existing video VLM benchmarks with MotionBench. MotionBench collects various video sources including web videos and synthetic videos, and provides a new evaluation perspective in motion level perception.

Benchmarks	#Videos	#QAs	Perception Level	Data source	Dataset Feature
MVBench [23]	4,000	4,000	general, motion<30%	existing datasets	general
TempCompass [28]	410	1,580	general, motion<20%	ShutterStock	temporal concept
VideoMME [8]	900	2,700	general, motion<20%	Youtube	general
AutoEval-Video [4]	327	327	event level	Youtube	open-ended QA
EgoSchema [31]	5,031	5031	event level	ego-centric video	ego-centric
LVBench [39]	103	1,549	event & story level	Youtube	long video
LongVideoBench [41]	3,763	6,678	event & story level	web channels	long videos
MovieChat-1K [35]	130	1,950	story level	movies	movie
Short Film Dataset [9]	1,078	4,885	story level	short films	story-level
MotionBench	5,385	8,052	motion level	web videos, movies, synthetic videos, datasets	motion perception

six domains. Video-MME [8] offers an evaluation framework featuring videos of varying durations—from 11 seconds to over an hour—while incorporating multimodal elements such as subtitles and audio. Some benchmarks focus on specific, challenging capabilities. For example, LVBench [39], LongVideoBench [41], and MLVU [54] target event- or story-level understanding across long temporal horizons. However, these benchmarks primarily focus on general video understanding, lacking a dedicated dataset or subset specifically designed for motion-level assessment. This limitation results in reduced volume and diversity in evaluating motion dynamics. Furthermore, most benchmarks rely on data from a single source, falling short of representing a comprehensive distribution of downstream applications.

To address these gaps, we propose MotionBench, a benchmark dedicated to fine-grained motion understanding. By leveraging data from seven distinct sources and encompassing six motion-oriented task categories, MotionBench offers a diverse range of video content and a specialized focus on motion-level perception, advancing the evaluation of video understanding models in this crucial area.

2.2. VLMs for video understanding

Recent advancements in Visual Language Models (VLMs) have demonstrated significant potential in video understanding, mostly extending pre-trained VLMs [25, 38] to handle video modality. Video VLMs typically comprise three core components: a visual encoder for visual feature extraction, a modality alignment module to integrate visual features into the language model's embedding space, and an LLM backbone for decoding multi-modal context. A straightforward architecture is LLaVA-Next-Video [50], CogVLM2-Video [15] and Intern-VL2 [6], where videos are treated as sequences of images, extending VLM's strong image understanding capabilities to videos. Qwen2-VL [36] further introduces 3D-RoPE to enable understanding of arbitrary-length videos. However, the high computational and memory demands of handling high-frame-rate, long-duration videos have prompted initial explorations into video compression in both pixel and feature spaces. For instance, InternVideo2 [40] and Video-LLaMA [49] adopt QFormer [20] for video feature extraction, PLLaVA [44] utilizes adaptive pooling, Kangaroo [26] employs a unified spatial-temporal patchification, and Qwen2-VL [36] fuses neighboring frames before visual encoder.

Despite these advancements, to our knowledge, no comprehensive and fair comparison exists among these compression methods and evaluating their performance as compression ratios increase. Moreover, current approaches are generally limited to shallow fusion that is confined to the compression operator itself, which restricts their performance, especially in high compression rate scenarios

3. MotionBench: Motion-Level Benchmarking

We introduce MotionBench, an evaluation benchmark designed to assess the motion-level perception capability of video VLMs. Fine-grained motion understanding is of paramount importance across a variety of daily scenarios, including human interaction, expression recognition, medical instruction, ambient object motion, sports replay, virtual reality, *etc*. Our approach begins with the collection of video clips from these diverse cases, which are then filtered and processed into the desired formats. We devise six primary categories of question types to evaluate the candidates' motion-level understanding, and we manually annotate the questions and answers within these categories, yielding the proposed MotionBench. Table 2 provides an overview of our data construction pipeline.

3.1. Data Curation

In this section, we elaborate on the video curation, filtering, and annotation process.

Video Collection. We obtain raw videos from publicly

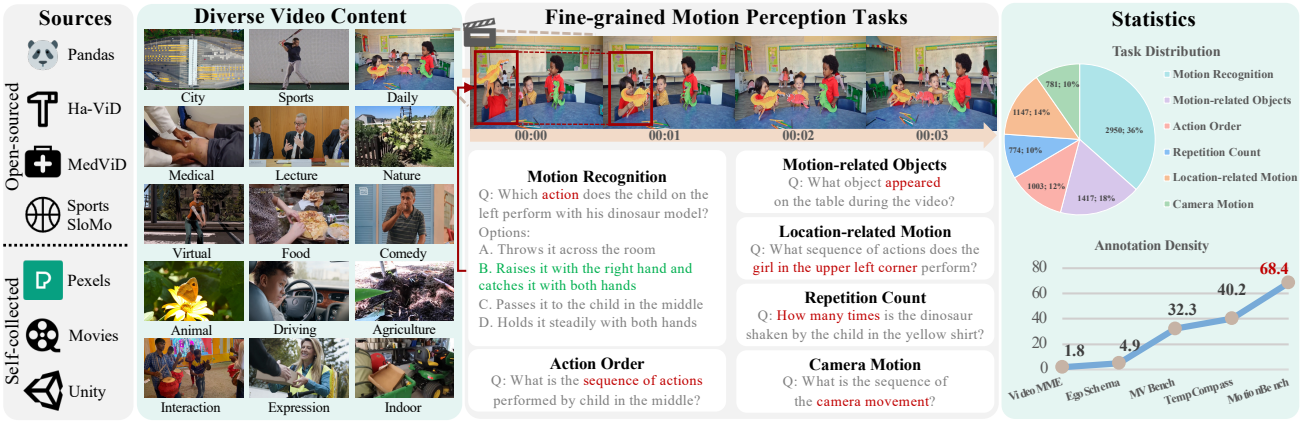


Figure 2. We propose MotionBench, a collection of manually curated multi-choice queries with video clips featuring dynamic changes from various scenes such as daily life and medical instructions. We devise six primary tasks to evaluate the capability of motion-level perception. Unlike previous story-level and event-level benchmarks, MotionBench is characterized by a significantly higher annotation density, allowing for the assessment of fine-grained motions.

Table 2. The MotionBench curation process. Categories [1-3] refer to "videos with intricate interactions", "videos from specific fields" and "virtual videos", detailed in Sec. 3.1. N. Vid/QA refers to the number of videos and queries in a category. min(H, W) is the minimum of the height and width of the video frames. *len* refers to the processed video duration. We automatically construct the queries in Virtual scenes, and manually annotate the other QA pairs in MotinBench.

Category	# Videos/QAs	Source	Collection	Post-process	Annotation
1	2,355/4,922	Pexels	Self-collected	Directly adopt	Caption & QA
		Pandas-70M [3]	Open-sourced	Segment with scene detection	Caption & QA
		Movie clips	Self-collected	Segment with scene detection	Caption & QA
2	2,430/2,530	MedVid [14]	Open-sourced	$\min(H, W) > 448$ & len$\in [3, 60]$sec	QA
		SportsSloMo [2]	Open-sourced	$\min(H, W) > 448$ & len$\in [3, 60]$sec	QA
		HA-ViD [52]	Open-sourced	$\min(H, W) > 448$ & len$\in [3, 60]$sec	QA
3	600/600	Virtual scenes	Self-collected	Remove renderings with occlusion	Automatic QA

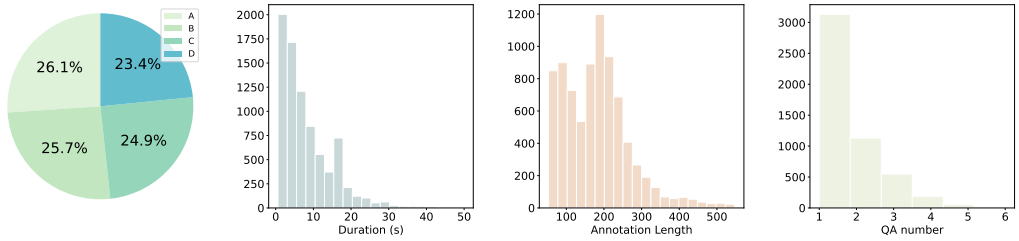
available datasets as well as from our self-collected corpus. Based on the video sources, the vividness of the scenes, and the complexity of the scenarios, we split the videos into three distinct categories. Each category is processed and annotated using tailored pipelines accordingly:

- **Videos with intricate interactions**: We acquire publicly-available videos from Panda-70M [3] and Pexels[4] and collect high-quality movie clips featuring various actions and motions, attributing to a total of 2355 videos. To ensure uniformity in clip duration, we follow the methodology in Panda-70M [3] to utilize a scene detection tool[5] to segment these videos into event-level clips.
- **Videos from specific fields**: We collect videos from MedVid [14], SportsSloMo [2] and Ha-ViD [52], representing specific use cases in medical, sports and industrial applications. These videos usually consist of one or two simple motions and demonstrate less complicated interactions. For this category, we filter out videos longer than 60 seconds or resolutions less than 448×448 pixels. An amount of 2430 videos are retrieved in this category.
- **Synthetic videos**: The above-mentioned videos are mostly from real-world scenes. For further evaluation in virtual reality applications, we render avatars with simple motions using the Unity rendering engine. Furthermore, graphic engines generate renderings that exclusively focus on motion changes, making them highly suitable for the assessment of motion perception. We randomly sample 20 motions from a publicly available website[6], and select 6 avatars and 5 scenes to render virtual avatars from a pool of 15 different viewpoints. Renderings with occlusion are manually filtered. Please refer to the supplementary for details in rendering.

Task Definition. To assess the capability in motion-level perception, we propose six categories of questions. Examples and the distribution of each category are illustrated in Fig. 2. A detailed description of each category is listed:

- **Motion Recognition (MR)**: Questions focus on what kind of motion emerged in the given video clips.

[4]https://www.pexels.com
[5]https://github.com/Breakthrough/PySceneDetect
[6]https://www.mixamo.com



(a) Option distribution (b) Video duration (c) Annotation length (d) QA per video

Figure 3. Basic statistics of MotionBench.

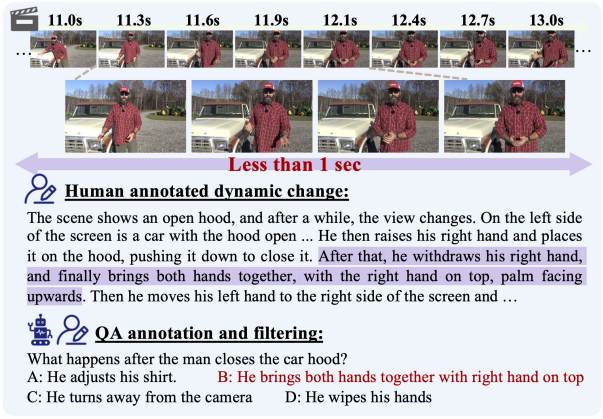


Figure 4. Example of dynamic information annotation

- **Location-related Motion (LM)**: Questions assessing the relative location changes before and after the motion takes place, and queries regarding a specific location.
- **Action Order (AO)**: Complex actions are composed of a sequence of motions. Questions in this category focus on the order of these motions.
- **Repetition Count (RC)**: Certain subtle motions occur rapidly but are repeated multiple times, such as nodding or jumping. This category of questions evaluates the model's ability to recognize and interpret such motions.
- **Motion-related Objects (MO)**: Queries designed to evaluate the model's ability to identify small objects involved in motion interactions.
- **Camera Motion (CM)**: Questions focus on the camera motion changes and trajectory, including the order and combinations of different motion types.

Question Answer Annotation. We employ different annotation pipelines for the above-mentioned video categories. For videos with intricate interactions, it is impractical to directly annotate the whole video clip, since the total complexity and quantity of the motions are too large. Therefore, we first manually annotate these videos with captions that focus on the dynamic changes within the video. Subsequently, we prompt GPT-4o [33] to generate 6 question-answer pairs for each video clip. For the prompt template and more details, please refer to the supplementary material. We find that the generated QA pairs are not only diverse in type but also presented in various sentence structures. We show an example of the dynamic information annotation pipeline in Fig. 4.

In addition, we also drop all the questions that can be answered solely based on common knowledge and a single frame. We use various **image** VLMs to predict answers using the first frame as input and discard questions that are answered correctly by all VLMs. Then, we manually filter out any questions with incorrect phrasing or ambiguous answers and categorize them. Finally, 4922 queries and answers are retained.

For videos from specific fields, we directly annotate the questions within the designed task types. A total of 2530 QA pairs are selected. For virtual videos, where we already possess the ground truth annotations for each query, we automatically construct the questions and corresponding options. Finally, 600 QA pairs are generated.

Evaluation protocal. we use regular expression matching to identify the first uppercase letter in the response as the prediction. We append "Only reply with the best option." to the end of each question for instruction following.

3.2. Dataset Statistics

MotionBench consists of 5385 videos and 8052 QAs, and each QA pair consists of a question, four options, an answer, and a category. The task distribution is displayed in Fig. 2.

Annotation Density. MotionBench is designed especially for evaluating the video VLM's motion-level perception capability. Such evaluation requires a larger annotation density per second. We define "Annotation Density" to represent such attribute, defined as follows:

$$\text{Annotation Density} = \frac{\text{Total length of questions}}{\text{Video duration}} \quad (1)$$

The results are demonstrated in Fig. 2. MotionBench features an Annotation Density of 68.4, which is two times more than existing benchmarks.

Basic Statistics. In Fig. 3, we illustrate the distribution of options, number of QAs per video, duration, and annotation

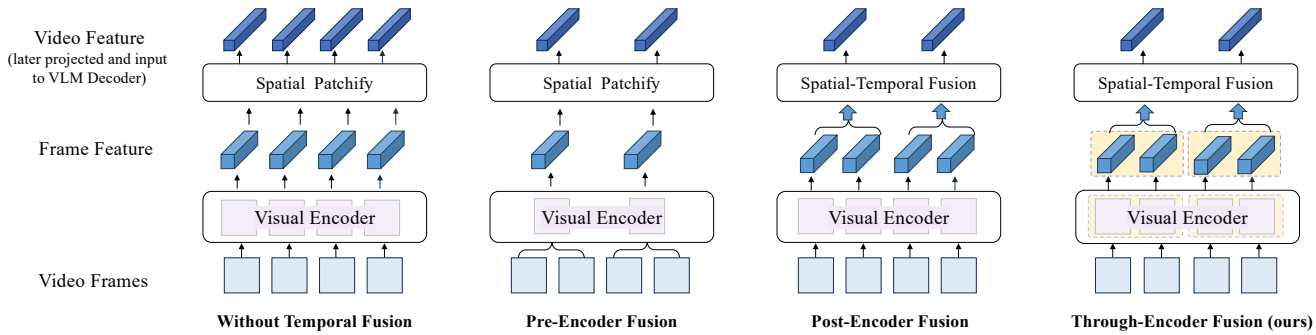


Figure 5. Summarization of prevalent paradigms for video compression and our proposed Through-Encoder Fusion (TE Fusion). Here we only illustrate the part before the VLM decoder where temporal compression performs.

length in the MotionBench. Regarding the distribution of answer options in MotionBench, it can be observed that the various options generally adhere to a random distribution. Due to our manual removal of erroneous and overly simplistic questions, it can be seen that the QA pairs in "Videos with intricate interactions" have been thoroughly filtered, resulting in the elimination of nearly half of the QA data. The video lengths in MotionBench are primarily concentrated around under 10 seconds, as motion events usually occur in very brief segments of the videos.

4. Model Design: Motion-Level Perception

Motion-level video perception demands high-frame-rate input, while the maximum input frame rate is significantly constrained by the sequence length limitations of VLMs, which are bounded by both infrastructure and computational budgets during training and inference. Therefore, it's necessary to design an efficient video understanding model structure with dense video representation. Recent studies, particularly in the domain of long video understanding, introduce various types of video feature compression methods [26, 37, 40, 44], but lack comprehensive and fair comparisons across all methods. Therefore, We comprehensively investigate commonly used architectures for video compression and categorize prevalent paradigms in Fig. 5.

- **Without Temporal Fusion**: A baseline widely used in [15, 50]. Each frame is independently processed by the visual encoder and projected into the decoder space.
- **Pre-Encoder Fusion**: This architecture conducts temporal fusion among neighboring k frames before the visual encoder, usually in pixel space. The temporal fusion operator varies across implementations. Typical examples include Qwen2-VL [37] where two adjacent frames are concatenated along the channel dimension for joint processing, and Kim et al. [17] which merges several nearby frames into a single image.
- **Post-Encoder Fusion**: In this architecture, each frame first *independently* goes through the visual encoder to generate frame-specific features, then performs feature fusion among neighboring frames with spatial-temporal fusion modules. Note that no temporal relationships are captured during visual encoding. This paradigm is the most widely adopted in video architecture with compression, with multiple variations in temporal fusion operators such as adaptive pooling [44], QFormer [20] [40], and unified spatial-temporal patchification [26].

All compression architectures rely on the assumption that redundancy exists between frames which contributes little to the video's comprehension and can therefore be removed. Achieving a higher compression ratio requires a more precise and thorough capture of this redundant information. However, current video temporal compression methods have a common limitation: the inter-frame relationships are considered only within the small compression operator, and each frame is treated independently before the operator. Consequently, it is difficult for this kind of shallow fusion to effectively capture higher-level redundancies. For instance, in a video of a running person, the individual's position, posture, and even the camera angle vary continuously. Only by applying sophisticated inter-frame fusion techniques can the model unify their representation throughout the video and capture this higher-level redundancy. Based on this observation, we propose a novel Through-Encoder Fusion paradigm that introduces deeper fusion across neighboring frames:

- **Through-Encoder Fusion (TE Fusion)**: During the visual encoding stage, adjacent frames are grouped in sets of k and apply group-level self-attention. This design gives the capacity to compute temporal dependencies through the whole visual encoder and conduct deep fusion. Following this, spatial-temporal compression is performed on each group of k frames.

Note that Through-Encoder Fusion represents a class of temporal compression methods that perform deep frame fusion before applying the compression operator. In this work, we experiment with the straightforward approach, leaving other variations for future exploration.

Table 3. Evaluation results of the existing video VLMs. Abbreviations: MR (Motion Recognition), LM (Location-related Motion), CM (Camera Motion), MO (Motion-related Objects), AO (Action Order), RC (Repetition Count). We randomly split MotionBench into "dev" and "test". We will release the ground truth answers in the "dev" set and set up an online platform for results submission in the "test" set.

Model	LLM	# Frames	Dev AVG (4020)	Test AVG (4034)	MR	LM	CM	MO	AO	RC
Random	-	-	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
*LLM: **Text** as Input*										
GPT-4o [33]	-	-	0.33	0.33	0.31	0.34	0.36	0.37	0.42	0.23
*Video VLMs : **Text + Multiple Frames** as Input*										
Gemini 1.5 Pro [34]	-	1fps	0.51	0.50	0.51	0.52	0.54	0.67	0.40	0.22
Qwen2VL-2B [36]	Qwen2 [37]	1fps	0.48	0.47	0.49	0.49	0.42	0.62	0.32	0.28
Qwen2VL-7B [36]	Qwen2 [37]	1fps	0.52	0.52	0.52	0.55	0.49	0.68	0.39	0.32
Qwen2VL-72B [36]	Qwen2 [37]	1fps	0.57	**0.58**	0.58	**0.61**	**0.63**	0.72	**0.47**	0.31
InternVL-40B [6]	NH-2-Yi-34B [32]	8	0.55	0.54	0.54	0.58	0.49	**0.76**	0.41	0.30
PLLaVA-34B [44]	Yi-34B [32]	16	0.52	0.51	0.55	0.51	0.47	0.66	0.38	0.31
CogVLM2-Video [15]	LLaMA3-8B [1]	24	0.41	0.44	0.43	0.39	0.38	0.64	0.37	0.33
GLM-4V-plus [15]	GLM4 [10]	30	0.54	0.55	0.57	0.57	0.54	0.69	0.40	0.37
LLaVA-NeXT [50]	Yi-34B [32]	32	0.48	0.40	0.53	0.45	0.36	0.66	0.39	0.23
MiniCPM-V2.6 [46]	Qwen2 [37]	64	0.52	0.53	0.56	0.49	0.45	0.72	0.39	0.33
Oryx-34B [29]	Yi-34B [32]	64	0.49	0.49	0.48	0.52	0.44	0.65	0.42	0.32
TE Fusion (ours)	GLM4-9B [10]	16	**0.58**	**0.58**	**0.64**	0.59	0.51	0.69	0.41	**0.39**

5. Experiments

5.1. Evaluation on MotionBench

We comprehensively evaluate the performance of existing video VLMs' capability in motion-level perception on MotionBench. We include multiple models with various model sizes and VLMs. The results are listed in Table 3. TE Fusion represents our proposed model, which uses TE Fusion on GLM-4V-9B backbone, with 16 input frames and a compress ratio of 4. Among existing VLMs, Qwen2VL-72B achieves the best overall performance on the dev and test set and scores highest in 3 out of 6 categories. Surprisingly, TE Fusion achieves state-of-the-art results with a 9B LLM backbone, verifying the effectiveness of our method.

Analysis. With text input alone, GPT-4 achieves an accuracy rate of 0.3 to 0.4, surpassing the random baseline of 0.25. This result indicates that LLMs possess a prior probability for certain actions, even when based only on text (note that questions answerable purely by common knowledge are filtered out during data curation). Building on LLMs, video VLMs improve accuracy by just 0.05 to 0.2, highlighting that current video VLMs still face challenges in reliably recognizing even short, simple motions. For the Repetition Count category, all models, except GLM-4V-9B with TE Fusion and GLM-4V-plus, scored near random. This is likely because fast motions are challenging to count at low frame rates or are easily overlooked by the models. Conversely, models generally achieved high scores in the Motion-related Objects category. This could be attributed to the pretraining video data, which is often constructed from image descriptions and emphasizes the objects in the video.

We further analyze the questions that all models fail to answer. The largest proportion involves fine-grained motion, suggesting that certain actions and their associated captions may be underrepresented in the training data. When examining questions by video duration, we find that even for short videos (0-4 sec), the proportion of all-model-failed questions remains 11% to 14%, highlighting models' difficulty in distinguishing certain motions even with limited content. As video duration increases, the failure rate rises significantly, reaching 18% for videos longer than 18 seconds. Further analysis from more perspectives and case studies are provided in the appendix.

5.2. Experiments on Video Feature Compression

To comprehensively and fairly evaluate all paradigms of video compression architecture, we implement representative methods from each paradigm based on the same image foundation model, GLM-4V-9B [15]: (1) Pre-encoder fusion: Qwen2-VL [37]; (2) Post-encoder fusion: QFormer [20], PLLaVA [44], Kangaroo [26]; (3) Through-encoder fusion: our proposed implementation; (4) Baseline without temporal fusion. All models take 224×224-pixel input and are trained for 10,000 iterations with a global batch size of 768 on the same collection of open-source datasets. Note that the training data is a subset of the data used in Sec. 5.1. The details of training and architecture are further provided in the Appendix. Besides MotionBench (dev), our motion-level video benchmark, we further evaluate all models on MVBench [23], LVBench [39], and Video-MME [8] as the representation of video benchmarks of varying duration and content.

Let N_{input} represent the number of frames fed into the visual encoder, and let each frame's uncompressed length at the VLM decoder be l tokens. With a given compression ratio k, the total compressed input length for the VLM

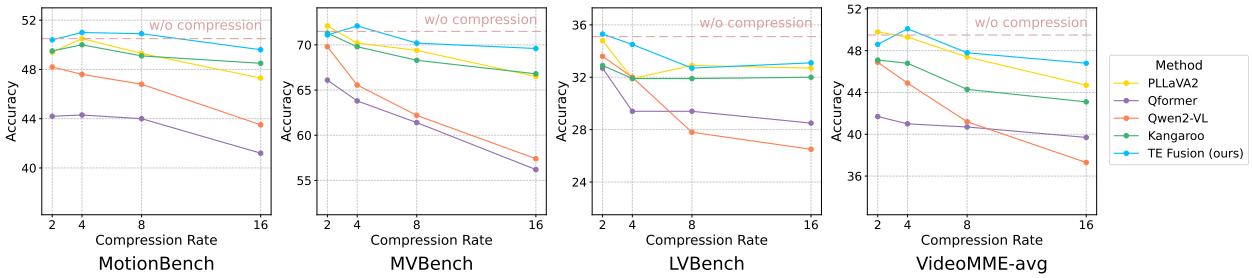


Figure 6. Model performance variation with respect to different compression ratios $k = 2, 4, 8, 16$, given a fixed VLM input frame count of $N_{\text{input}} = 16$. The pink dotted line represents the performance of the baseline model, which processes 16 frames without temporal compression. Note that each compression method is re-implemented on the GLM-4V-9B backbone to ensure a fair comparison.

Table 4. Benchmark results for different compression methods at various compression rates, all using the same sequence length in the VLM decoder. We set $\frac{N_{\text{input}}}{k} = 4$, with the baseline representing video models that process 4 frames without compression. Note that each compression method is re-implemented on the GLM-4V-9B backbone to ensure a fair comparison.

k	Method	MotionBench	MVBench	VideoMME		
				short	medium	long
1	baseline	47.6	64.5	51.4	41.0	38.3
2	QFormer	43.5	62.1	42.8	39.6	36.3
	Qwen2-VL	48.0	66.5	54.1	43.1	37.8
	PLLaVA	48.5	68.8	54.9	44.9	39.6
	Kangaroo	48.4	**69.2**	55.4	43.0	38.8
	TE Fusion (ours)	**49.1**	69.0	**55.2**	**46.3**	**40.0**
4	QFormer	44.3	63.8	45.2	41.0	36.8
	Qwen2-VL	47.6	65.6	51.8	43.4	39.4
	PLLaVA	50.5	70.2	58.9	46.4	41.3
	Kangaroo	50.0	69.8	55.3	45.6	39.5
	TE Fusion (ours)	**51.0**	**72.1**	**61.0**	**47.3**	**42.1**

decoder is $L_{\text{decoder}} = \frac{N_{\text{input}} \times l}{k}$. Our experiment centers on addressing two primary questions:

1. For a fixed sequence length at the VLM decoder (L_{decoder}), how does performance vary as the compression ratio increases?
2. For a fixed number of input frames (N_{input}), how does performance respond to changes in the compression ratio, and is there an optimal compression ratio?

For the first question, we conduct experiments with $\frac{N_{\text{input}}}{k} = 4$ and 8, varying the compression rate k at $2, 4, 6$, and 8. Results for $\frac{N_{\text{input}}}{k} = 4$ are shown in Tab. 4, with complete results included in the Appendix due to space constraints. Given the same L_{decoder}, most temporal compression methods demonstrate performance improvements across all benchmarks, with higher compression rates generally yielding better scores. Notably, PLLaVA, Kangaroo, and TE Fusion show relatively strong results, with our TE Fusion achieving the highest scores in 9 out of 10 metrics, improving upon the baseline by 11.8% on MVBench and 18.7% on VideoMME-short with $k = 4$. Qwen2-VL performs well with $k = 2$ but shows minimal improvement (or even a decline) with $k = 4$, likely due to the limited high-level compression capabilities of post-encoder fusion. QFormer, on the other hand, occasionally underperforms compared to the baseline, potentially due to the complexity of the additional module, which is challenging to optimize during the video compression training stage.

For the second question, we set the input frame count to $N_{\text{input}} = 16$ and test compression rates of $k = 2, 4, 6$, and 8 across all methods. The results, shown in Fig. 6 (with full numerical data in the appendix), reveal that while all methods experience some performance decline as the compression rate increases, our TE Fusion method exhibits almost no performance drop for $k \leq 4$. Even with a larger $k = 16$, the average performance reduction remains under 4% compared to the high-consumption baseline without compression. Additionally, the performance decline caused by temporal compression is less significant in shorter-duration videos (MotionBench, MVBench) compared to longer-duration videos (LVBench), suggesting that high-frame-rate input offers greater potential for effective, high-ratio temporal compression. Interestingly, We find that TE fusion achieves the highest score with compression-4 instead of compression-2 in 3 of 4 datasets. An explanation is that a higher compression rate increases attention length within the ViT component while decreasing it in the LLM component. This finding suggests that the computational allocation in previous video VLMs may be suboptimal and enlightens a new direction to improve model performance.

6. Conclusion

We present MotionBench, a new benchmark for assessing fine-grained motion understanding in video models. Our experiments show that current state-of-the-art models struggle with motion-level comprehension, emphasizing the need for specialized benchmarks. To tackle this, we propose the Through-Encoder (TE) Fusion method, which improves video feature representation by deeply integrating fusion within the visual encoder. TE Fusion achieves state-of-the-art results, especially under high compression, paving the way for advances in motion perception.

Acknowledgments

This work is supported by Natural Science Foundation of China (NSFC) 62425601 and 62495063, Daimler Greater China Ltd. and Tsinghua University Joint Institute for Sustainable Mobility and the New Cornerstone Science Foundation through the XPLORER PRIZE. We thank Xiaohan Zhang, Yuean Bi, Xiaoying Ling, Jiapeng Wang, Zikang Wang from Zhipu AI for managing the data annotation team, and Zhao Xue from Zhipu AI for data management.

References

[1] AI@Meta. Llama 3 model card. 2024. 7

[2] Jiaben Chen and Huaizu Jiang. Sportsslomo: A new benchmark and baselines for human-centric video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6475–6486, 2024. 2, 4, 14

[3] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, pages 13320–13331, 2024. 2, 4, 14

[4] Xiuyuan Chen, Yuan Lin, Yuchen Zhang, and Weiran Huang. Autoeval-video: An automatic benchmark for assessing large vision language models in open-ended video question answering. *arXiv preprint arXiv:2311.14906*, 2023. 3

[5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 2

[6] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024. 3, 7

[7] Yifan Du, Kun Zhou, Yuqi Huo, Yifan Li, Wayne Xin Zhao, Haoyu Lu, Zijia Zhao, Bingning Wang, Weipeng Chen, and Ji-Rong Wen. Towards event-oriented long video understanding. *arXiv preprint arXiv:2406.14129*, 2024. 2

[8] 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*, 2024. 1, 2, 3, 7

[9] Ridouane Ghermi, Xi Wang, Vicky Kalogeiton, and Ivan Laptev. Short film dataset (sfd): A benchmark for story-level video understanding. *arXiv preprint arXiv:2406.10221*, 2024. 2, 3

[10] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. 7

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 12

[12] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 12

[13] Deepak Gupta, Kush Attal, and Dina Demner-Fushman. A dataset for medical instructional video classification and question answering. *Scientific Data*, 10(1):158, 2023. 2, 14

[14] Deepak Kumar Gupta, Kush Attal, and Dina Demner-Fushman. A dataset for medical instructional video classification and question answering. *Scientific Data*, 10, 2022. 4

[15] 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. 1, 2, 3, 6, 7

[16] Y. Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 12

[17] Wonkyun Kim, Changin Choi, Wonseok Lee, and Wonjong Rhee. An image grid can be worth a video: Zero-shot video question answering using a vlm. *arXiv preprint arXiv:2403.18406*, 2024. 6

[18] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

[19] Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. Aria: An open multimodal native mixture-of-experts model. *arXiv preprint arXiv:2410.05993*, 2024. 1

[20] 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*, pages 19730–19742. PMLR, 2023. 3, 6, 7

[21] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Y. Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *ArXiv*, abs/2211.09552, 2022. 12

[22] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wen Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *ArXiv*, abs/2305.06355, 2023. 12

[23] 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*, pages 22195–22206, 2024. 1, 2, 3, 7

[24] Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. Videovista: A versatile benchmark for video understanding and reasoning. *arXiv preprint arXiv:2406.11303*, 2024. 1

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3

[26] 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*, 2024. 3, 6, 7

[27] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 1

[28] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024. 3

[29] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*, 2024. 1, 7

[30] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. 2024. 12

[31] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023. 3

[32] NousResearch. Yi-vl-34b, 2024. 7

[33] OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023. 5, 7

[34] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 1, 7

[35] 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*, pages 18221–18232, 2024. 3

[36] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 3, 7

[37] 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*, 2024. 1, 6, 7

[38] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 3

[39] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024. 1, 2, 3, 7

[40] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024. 3, 6

[41] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024. 1, 3

[42] Junbin Xiao, Xindi Shang, Angela Yao, and Tat seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 12

[43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 2

[44] 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. 1, 3, 6, 7

[45] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 12

[46] 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. 7

[47] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *ICLR*, 2020. 12

[48] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9127–9134, 2019. 2

[49] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

[50] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 2, 3, 6, 7

[51] 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*, 2024. 1

[52] Hao Zheng, Regina Lee, and Yuqian Lu. Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding, 2023. 4, 14

[53] Hao Zheng, Regina Lee, and Yuqian Lu. Ha-vid: a human assembly video dataset for comprehensive assembly knowledge understanding. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[54] 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. 1, 3

MotionBench: Benchmarking and Improving Fine-grained Video Motion Understanding for Vision Language Models

Supplementary Material

7. Training Details

Here we provide the detailed training hyperparameters for both TE Fusion in Tab. 3 and all ablated models in Tab. 4 and Fig. 6.

Table 5. Training settings TE Fusion in Tab. 3 and all ablated models in Tab. 4 and Fig. 6.

	Configurations
Total steps	10,000
Warmup steps	1,000
Global batch size	768
Learning rate	8e-6
Minimal learning rate	1e-6
Learning rate decay	cosine
Optimizer	Adam
Adam ϵ	1e-8
Adam $\beta 1$	0.9
Adam $\beta 2$	0.95
Precision	bf16

The training is conducted on several datasets, mainly including VideoChat [22], VideoChatGPT [30], NExT-QA [42], CLEVRER [47], Kinetics-710 [21], Sth-SthV2 [11], Ego4D [12], TGIF-QA [16], WebVidQA [45], In-house VideoQA Dataset. We also include an in-house video QA dataset for better temporal understanding.

8. Model Details

A detailed explanation of TE Fusion: Consider the temporal/spatial downsampling factor k_t / k_s. The input video is first patchified into $X_{\text{video}} \in \mathbb{R}^{T\times h\times w\times d}$. Every k_t frames are grouped along the spatial dimension, forming $X_{\text{in}} \in \mathbb{R}^{\frac{T}{k_t}\times(k_t hw)\times d}$. X_{in} is processed by a ViT-based visual encoder, where self-attention operates within each group of k_t frames. Note that the group-by-k_t attention is only one implementation of deep fusion across k_t frames, thus TE Fusion can be extended to a paradigm of architectures. The encoder outputs:

$$X_{\text{out}} = \text{stack}(\text{enc}(X_{\text{in}}^{1:k_t}), \text{enc}(X_{\text{in}}^{k_t+1:2k_t}), ..., \text{enc}(X_{\text{in}}^{T-k_t+1:T})),$$

where $X_{\text{in}}^{(i\cdot k_t+1):((i+1)\cdot k_t)} \in \mathbb{R}^{(k_t hw)\times d}$ represents the i-th group, and $X_{\text{out}} \in \mathbb{R}^{\frac{T}{k_t}\times(k_t hw)\times d}$.

Then, X_{out} is reshaped into $\mathbb{R}^{T\times h\times w\times d}$ and processed by a 3D convolutional operator with kernel size (k_t, k_s, k_s) and stride (k_t, k_s, k_s), producing downsampled features $X_{\text{feat}} \in \mathbb{R}^{\frac{T}{k_t}\times\frac{h}{k_s}\times\frac{w}{k_s}\times d_{\text{LLM}}}$, which are fed into the LLM. This grouped-by-k_t attention and k_t-downsampling allow deeper fusion of adjacent k_t frames, effectively capturing inter-frame dynamics and redundancy, therefore retaining critical features for subsequent processing.

Comparison among different compression architectures: Assume the temporal compression ratio be K, The specific feature of each ablated architecture is:

1. TE-Fusion (ours): Before the visual encoder, we concatenate every neighboring K frames into one sequence, and conduct self-attention across each K frames to fuse temporal feature. After the visual encoder, the tokens of K frames are concatenated along the hidden-size dimension, downsampled and projected to the output dimension.
2. Qwen2-VL: The neighboring K frames are concatenated along the channel dimension and patchified into one feature. Afterward, they go through the visual encoder as a whole. Since the fusion is conducted in the pixel space before any feature extraction or fusion, the optimized temporal compression ratio is usually low, with a vast information loss if a large K.
3. Kangaroo: This approach is the most similar one to TE Fusion, except that every frame is computed independently within the visual encoder and concatenated along the hidden size dimension to perform temporal downsample (with an MLP layer).
4. QFormer: After going through the visual encoder, the video feature is passed through a QFormer (learned from scratch). Every K frames' feature is combined into a sequence to fusion temporal information within the QFormer. From the experiment, we found that, though being light-weighted, the QFormer is hard to optimize and model temporal relationships during the video instruction-tuning stage, resulting in poor performance.
5. PLLaVA: This approach is similar to Kangaroo. Instead of fusion with the MLP layer, PLLaVA adopts a simple adaptive pooling. To avoid possible information loss, we conduct the pooling operation after the spatial downsample module.

To maintain a fair comparison, all model architectures are ablated with the same backbone, GLM-4V, with its model configuration as follows:

Table 6. The model configurations of all ablated architectures.

VLM decoder	
Layers	40
Hidden size	4096
Attention heads	32
num query groups	2
FFN hidden size	13696
Sequence len	4096
Position embedding	RoPE
Normalization	RMSNorm
visual encoder	
Input resolution	224
Patch size	14
Post spatial downsample	2×2
Layers	63
Hidden size	1792
Attention heads	16

9. QA Construction Process for Videos with Intricate Interactions

Here we illustrate the QA generation process corresponding to Fig. 4.

9.1. Step1: Video caption annotation

For videos with intricate interactions, it is impractical to directly annotate the whole video clip, since the total complexity and quantity of the motions are too large. Therefore, we first manually annotate these videos with captions that focus on the dynamic changes within the video (illustrated in Fig. 4). We hired 15 adult annotators with at least a bachelor's degree and conducted annotations over 20 working days. Each annotator's daily salary was approximately 250 RMB. All annotations underwent a secondary review.

9.2. Step2: Automatic QA generation

Then we use GPT-4o to generate 6 questions corresponding to each video description. The instruction to GPT-4o emphasizes diversity as well as accuracy, as shown below:

You are a professional question designer specializing in dynamic video details. Instead of a video, you will receive a detailed description of the first frame and all dynamic details throughout the video. Based on this description, design single-choice questions that focus on **the dynamic information** as if you're viewing the video directly, using the two-dimensional categorization system below (Content Dimension, Question Logic Dimension).

Question Design Guidelines

1. Each question should have 4 options.
2. For each question, combine one dimension from the Content Dimension and one from the Question Logic Dimension. It may draw from multiple highly related content dimensions.
3. Focus only on representative and prominent events or actions to keep options clear and unique without being overly detailed or tricky. **Select the most fitting dimension combination** for each video and avoid repeated combinations where possible.
4. Given possible ambiguities in some descriptions, **ensure the answer is unique and clear** to avoid deductions.
   - **Ambiguity Example 1: Temporal ambiguity.** If a description reads, "On the left, a woman in a khaki suit faces right, nodding her head while speaking. In the middle, a group faces the camera, and a man in a white shirt pulls a chair leftward to sit," the description is ambiguous and does not clarify the sequence of the woman's actions and the man's actions, making sequence ambiguous.
   - **Ambiguity Example 2: Content ambiguity.** If the description states, "The worker holds a long, thin tool," avoid options like "screwdriver," as the tool could be any slender object.
5. Choose only prominent events or actions, avoiding minor or indeterminate details. Ensure each answer is **unique and clear**.
   - **Minor Example:** If "slightly bent elbow" isn't mentioned, it does not necessarily mean it did not happen; if the video says "the mouth moved slightly a few times," it cannot be determined the interval and number of these movements, nor can it be determined whether the nose moved. Therefore, try to avoid using such minor actions for question creation or option design.
   - Avoid subjective options, like "Which detail reflects focus on work?" unless a behavior clearly reflects it. Similarly, avoid terms like "skilled movement" or "rhythmic."
   - Avoid overly similar distractors, e.g., "chin moving up and down" vs. "slight opening and closing."
6. Pretend you're viewing the video, avoiding terms like "based on the description" or expressions related to the description text, including questions, options, and explanations.
7. Aim for at least 4 questions to focus beyond appearance.
8. Keep questions to around six, focusing only on representative events or actions and ensuring options are clear, unique, and straightforward.
9. Questions should focus on dynamic actions only. The "first frame description" is supplementary and should not guide question design.
10. The video dynamic information description does not contain causal or other logical relationships, therefore,

do not involve logical relationships in the title.

Categorization System

Content Dimension Below is the **Content Dimension** in the video classification system:

1. **Human Dynamics:**
   1.1. Detailed actions of individuals
   1.2. Interaction among multiple people
   1.3. Emotional states and their changes
   1.4. Position and its changes (Location, Angle, etc.)
2. **Object Dynamics:**
   2.1. Movement trajectory
   2.2. State changes
3. **Animal Dynamics:**
   3.1. Detailed actions
   3.2. Position and its changes (Location, Angle, etc.)
4. **Camera Movement:**
   4.1. Camera movement
5. **Appearance Characteristics:**
   5.1. individuals
   5.2. objects
   5.3. environment

Question Logic Dimension Below is the **Question Logic Dimension** in the video classification system:

1. Whether a movement occurs
2. Movement count
3. Sequence between multiple movements
4. Appearance description and judgment

Response Format

Return only a Python list, where each element is a dictionary representing a question. Ensure it can be parsed by `json.loads()` without returning anything outside the list.

9.3. VLM Filtering

To avoid over simple QAs that do not utilize motion comprehension capability, we use various image VLMs to predict answers using the first frame as input and discard questions that are answered correctly by all VLMs. The VLMs include GPT-4o, Qwen2-VL, and GLM-4V-plus.

9.4. Manual Check

To ensure the correctness of all benchmark QAs, we further hire annotators to check all QAs generated by GPT-4o manually. A total of 10 annotators are hired to conduct manual checks for 5 days. The key points of inspection include: the reasonableness of the question, the correctness of the category, the relevance of the question to the video, the accuracy of the options, and the uniqueness of the correct answer. Each annotator's daily salary was approximately 250 RMB. All annotations underwent a secondary review.

10. Copyrights

MotionBench is a research preview intended for non-commercial use only. For existing open-sourced video sources [2, 3, 13, 52], we have carefully signed their provided license and will not re-distribute their videos without permission. For videos from Pexels, we will mandatorily ask the users to sign an agreement that the videos in MotionBench can only be used in non-commercial research and cannot be re-distributed. For self-collected movie clips, we will not directly distribute the raw videos, and will alternatively provide the download links and processing scripts.

11. The originality of MotionBench.

Unlike former works that primarily focus on fixed action labels, limited gestures, or predefined video/action fields (*e.g.*, egocentric scenes, sports, MotionBench aims to highlight motion-level understanding *across general domain with large-scale QAs*. Key innovations include:

1. Various QAs are uniformly distributed across diverse video sources, descriptions and motion object(s) for all categories. In contrast, MVBench evaluates limited video sources and scenes: subcategories FA, MD and OI each rely only on a single data source, and MD features only simulated movements of simple objects;
2. We process and annotate the raw videos from scratch with high annotation length (Fig.3a), without leveraging any existing annotations. MVBench heavily relies on previous datasets' annotations, which offers little incremental information and poses the danger of training data leakage;
3. A semi-automated pipeline for discovering and annotating motions from the video sources. MVBench uses a rule-based pipeline without human corrections.

12. More Experimental Results

Given the same sequence length in the VLM decoder, we benchmark results for different compression methods at various compression rates. We conduct experiments with $\frac{N_{\text{input}}}{k} = 4$ and 8, varying the compression rate k at $2, 4, 6,$ and 8. Tab. 7 provide the complete results.

Given the same VLM input frame count, we experiment different compression ratios on various architectures, with the numerical results illustrated in Tab. 8.

13. Case Study on Model Performance

We show more case studies regarding the performance of existing models on MotionBench.

Questions that confuses all models. As shown in Table 3, MotionBench is highly challenging for existing video

Table 7. Benchmark results for different compression methods at various compression rates, all using the same sequence length in the VLM decoder. We set $\frac{N_{\text{input}}}{k} = 4, 8$, with the baseline representing video models that process 4 frames without compression. Note that each compression method is re-implemented on the GLM-4V-9B backbone to ensure a fair comparison.

Equivalent Frames $\frac{N_{\text{input}}}{k}$	Compress Rate	Method	MotionBench (dev)	MVBench	LVBench	VideoMME		
						short	medium	long
4	1	baseline	47.6	64.5	30.9	51.4	41.0	38.3
	2	QFormer	43.5	62.1	31.0	42.8	39.6	36.3
		Qwen2-VL	48.0	66.5	31.5	54.1	43.1	37.8
		PLLaVA	48.5	68.8	**33.4**	54.9	44.9	39.6
		Kangaroo	48.4	**69.2**	31.6	55.4	43.0	38.8
		TE Fusion (ours)	**49.1**	69.0	32.3	**55.2**	**46.3**	**40.0**
	4	QFormer	44.3	63.8	29.4	45.2	41.0	36.8
		Qwen2-VL	47.6	65.6	32.0	51.8	43.4	39.4
		PLLaVA	50.5	70.2	34.3	58.9	46.4	41.3
		Kangaroo	50.0	69.8	31.9	55.3	45.6	39.5
		TE Fusion (ours)	**51.0**	**72.1**	**34.5**	**61.0**	**47.3**	**42.1**
8	1	baseline	48.9	70.5	32.9	56.4	44.2	39.7
	2	QFormer	44.2	66.1	32.7	48.0	39.8	37.2
		Qwen2-VL	48.2	69.8	33.6	57.3	44.1	39.4
		PLLaVA	49.4	**72.1**	34.8	**61.0**	46.4	39.8
		Kangaroo	49.5	71.3	32.9	58.3	45.2	37.7
		TE Fusion (ours)	**50.4**	71.1	**35.3**	58.7	**46.9**	**40.2**
	4	QFormer	44.4	66.0	31.6	45.7	40.0	37.2
		Qwen2-VL	48.7	69.3	33.1	55.2	43.3	38.1
		PLLaVA	49.4	71.5	**36.2**	60.3	47.3	41.1
		Kangaroo	49.9	**71.6**	33.5	59.0	45.8	38.2
		TE Fusion (ours)	**50.5**	**71.6**	36.0	**63.0**	**47.9**	**41.5**

Table 8. Model performance variation with respect to different compression ratios $k = 2, 4, 8, 16$, given a fixed VLM input frame count of $N_{\text{input}} = 16$. Note that each compression method is re-implemented on the GLM-4V-9B backbone to ensure a fair comparison.

Method	Compress Rate	MotionBench	MVBench	Video-MME			LVBench
				short	medium	long	
w/o compression	1	50.5	71.5	60.7	46.6	41.1	35.1
PLLaVA	2	49.4	72.1	61.0	46.4	42.0	34.8
	4	50.5	70.2	58.9	47.6	41.3	31.9
	8	49.3	69.4	56.7	45.2	40.4	32.9
	16	47.3	66.5	52.4	42.8	39.0	32.7
QFormer	2	44.2	66.1	48.0	39.8	37.2	32.7
	4	44.3	63.8	45.2	41.0	36.8	29.4
	8	44.0	61.4	45.3	40.6	36.3	29.4
	16	41.2	56.2	44.2	39.4	35.4	28.5
Qwen2-VL	2	48.2	69.8	57.3	44.1	39.4	33.6
	4	47.6	65.6	51.8	43.4	39.4	32.0
	8	46.8	62.2	47.2	39.9	36.4	27.8
	16	43.5	57.4	38.9	37.6	35.3	26.5
Kangaroo	2	49.5	71.3	58.3	45.2	37.7	32.9
	4	50.0	69.8	55.3	45.6	39.5	31.9
	8	49.1	68.3	51.9	42.3	38.7	31.9
	16	48.5	66.8	49.8	42.4	37.1	32.0
TE Fusion (ours)	2	50.4	71.1	58.7	46.9	40.2	35.3
	4	51.0	72.1	61.0	47.3	42.1	34.5
	8	50.9	70.2	56.6	45.8	41.1	32.7
	16	49.6	69.6	54.8	45.8	39.8	33.1

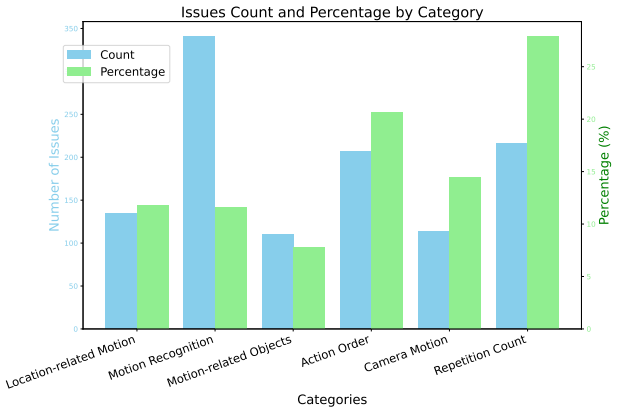


Figure 7. The absolute number and the proportion of questions that all models answered incorrectly relative to the total number of questions in each task type.

understanding models. Currently, even the best video understanding models can achieve only less than 60% accuracy. In MotionBench, there are some questions for which all models output incorrect answers. Figure 7 shows the absolute number and the proportion of questions that all models answered incorrectly relative to the total number of questions in each task type. Firstly, compared to the total number of questions in every task type, only a small fraction of questions were answered incorrectly by all models. Among the tasks, the highest proportion of questions that all models answered incorrectly is that in the "Fast action count" task type. This attributes to counting repetitive actions at the motion level is inherently a very challenging task, and current video understanding models still struggle to handle such issues correctly.

Case study. We show a case that all the models answered incorrectly. This is a case in which a male's hand is touching the car from the top and move to the lower left. However, most of the models believe that the video presents a hand "tapping on the car surface". Such prediction is correct from a single image perspective, while in the video, the hand stays on the car surface and moves from the top to the lower left. Hence the gesture "tapping" is not correct. This example demonstrates that single-frame predictions and perceptions can sometimes be misleading or even incorrect at the temporal level, which further underscores the value of creating a benchmark focused on motion-level temporal sequences.

14. Limitations and Broader impact

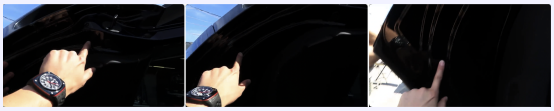
We propose MotionBench, a video understanding benchmark assessing the models' motion-level perception capability. However, there are several limitations to our approach that should be acknowledged. Firstly, although we have made efforts to include a diverse range of video content, our dataset may still have inherent biases in terms of geographical, cultural, and contextual variety. This could potentially limit the generalizability of research findings based on this dataset to different settings. Secondly, while we have performed extensive annotations, there may be occasional inaccuracies or inconsistencies due to human and automatic tool error.

Regarding the broader impact, motion-level perception is pivotal in video understanding. MotionBench provides a comprehensive benchmarking on video VLMs' motion-level perception. By making our dataset publicly available, we hope to further enhance the capabilities of video understanding models, thereby improving their applicability in real-world scenarios.

15. More Dataset Samples

For better demonstration, we show more samples from the MotionBench.

uid=y26CvHFcz7BboSXN_0



What action does the hand in the video perform?
A. Taps on the car surface (Gemini-1.5 pro, InternVL-40B, Oryx-34B, Qwen2-VL-72B)
B. Remains stationary (PLLaVA-34B)
C. Moves towards the lower left
D. Waves back and forth

Task type: Motion Recognition

What is the sequence of movements between the two males?
A. The male on the right raises his hand first, then the left male removes the string
B. No movement occurs at all
C. Both actions occur simultaneously
D. The male on the left removes the string first, then the right male raises his hand

Task type: Motion Recognition

Which facial movement occurs with the woman on the right?
A. Full head tilt down
B. Slight head turn to the right
C. Eyes close briefly
D. Look straight ahead throughout

Task type: Action Order

What is the sequence of ball movement in the video?
A. The ball is thrown to the left and then rolls back from the left.
B. The ball rolls from the left and then is thrown to the right.
C. The ball is thrown to the right and rolls from the right.
D. The ball is thrown upwards and rolls down.

Task type: Action Order

What is the sequence of actions involving the two men?
A. The man on the right raises his hands first, followed by the man on the left
B. Neither man raises their hands
C. The man on the left raises his hands first, followed by the man on the right
D. Both men raise their hands simultaneously

Task type: Motion-related Objects

What did this person take with his right hand?
A. Screw
B. Thumbtack
C. Bolt nut
D. Pen core

Task type: Motion-related Objects

What does the camera reveal as it moves backward over the road?
A. A crosswalk appearing
B. A sign on top of the lead car
C. The end of a line of parked cars
D. The cars stopping abruptly

Task type: Location-related Motion

In what order does the plane appear and move across the screen?
A. From the left to completely leaving the frame
B. From the top left to bottom right
C. From the right to the upper part before disappearing
D. From the bottom left to top right

Task type: Location-related Motion

What movement trajectory does the horse follow?
A. Stays in place
B. Moves directly towards the camera
C. Gallops in circles
D. Jumps over an obstacle

Task type: Repetition Count

Please count the number of repeated actions in the video.
A. 3
B. 6
C. 9
D. 4

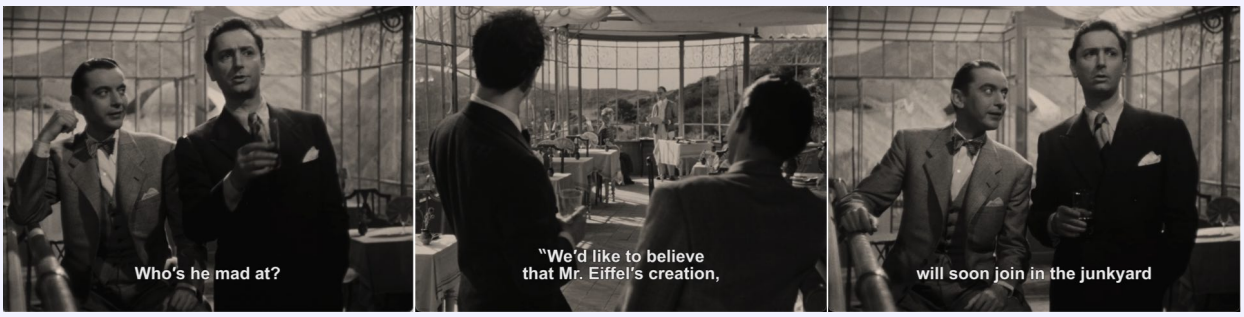
Task type: Repetition Count

Please count the number of repeated actions in the video.
A. 3
B. 2
C. 1
D. 6

Task type: Camera Motion

Does the camera perform any movement during the scene?
A. Yes, it zooms in.
B. No, it remains static.
C. Yes, it pans to the left.
D. Yes, it rotates counterclockwise.

Task type: Camera Motion



What is the sequence of the camera movements during the interaction?
A. The camera stays still, only focusing on the woman
B. The camera shifts to show the men with their backs, then returns to face the men
C. The camera shifts to show both men together, then moves back to the woman
D. The camera starts facing the men, shifts to the woman, then moves back to the men