META-VIDEO: EFFICIENT STANDARDIZED EVALUATION FOR VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video understanding benchmarks are the primary yardstick for measuring progress in video-language modeling, and a growing body of work relies on them to claim improvements. The reported benchmark scores in these works, however, are almost always produced by frameworks (e.g., `lmms_eval`) used as *black boxes*: certain evaluation choices like frame sampling, preprocessing, and input formatting are inherited from preset defaults and seldom inspected carefully. We show these unexamined choices are a major source of variance. Even for the same model checkpoint, evaluation configuration alone can shift accuracy by up to 19.5pp. Hence, it is unclear whether reported gains reflect genuine improvements in capability or differences in evaluation setup. Moreover, many benchmark samples offer little evidence of visual capability, since they can be answered *without* the video, and the full suites are expensive to evaluate. We address these issues in two parts. First, we curate existing suites for discriminability and efficiency, producing **Meta-Video**, a balanced 1,000-question benchmark over 791 videos spanning diverse tasks and durations, which preserves model rankings, at 5× inference speedup. Second, we establish a standardized evaluation protocol by sweeping the configuration space to isolate choices that measurably affect results, distill them into best practices and reporting standards, and release reproducible reference scores across five models and seven benchmarks. We hope that the proposed evaluation framework and the **Meta-Video** benchmark will support video understanding research by providing a reproducible and efficient foundation to measure progress in a reliable way.

Table 1: **Performance reported for the Qwen2.5-VL-7B and Qwen3-VL-8B backbone.** The table shows reported results for Qwen2.5-VL-7B and Qwen3-VL-8B-Instruct across 7 video benchmarks incl. original Qwen papers, subsequent studies and our reproduction. Max Δ denotes the range of scores for each benchmark, differing by up to 19.5pp despite identical model weights.

Model	VideoMME	TempCompass	MVBench	VSIBench	VideoMMMU	LongVideoBench	LVBench
Base model: Qwen2.5-VL-7B							
Qwen2.5-VL-7B (Bai et al., 2025b)	65.1	71.7	69.6	–	47.4	56.0	45.3
VideoRFT (Wang et al., 2025a)	52.8$_{-12.3}$	69.2$_{-2.5}$	59.4$_{-10.2}$	31.8	47.4	56.0	45.3
Videochat-R1.5 (Yan et al., 2026a)	64.4$_{-0.7}$	–	67.4$_{-2.2}$	–	46.7$_{-0.7}$	–	–
Open-o3-video (Meng et al., 2025b)	62.4$_{-2.7}$	–	–	–	51.2$_{+3.8}$	–	–
Video-R2 (Maaz et al., 2025)	66.1$_{+1.0}$	74.2$_{+2.5}$	68.1$_{-1.5}$	37.7	50.3$_{+2.9}$	61.3$_{+5.3}$	–
Viss-R1 (Fang et al., 2026)	56.6$_{-8.5}$	72.6$_{+0.9}$	59.0$_{-10.6}$	30.1	48.1$_{+0.7}$	–	–
Rethinking CoT (Zhong et al., 2025)	55.5$_{-9.6}$	–	63.2$_{-6.4}$	–	–	52.5$_{-3.5}$	34.2$_{-11.1}$
STORM (Liang et al., 2026)	55.4$_{-9.7}$	69.9$_{-1.8}$	60.5$_{-9.1}$	–	–	–	–
CoVER (Wang et al., 2026b)	63.1$_{-2.0}$	–	–	–	–	56.0	42.0$_{-3.3}$
Video-VER (Luo et al., 2026)	59.2$_{-5.9}$	73.7$_{+2.0}$	63.6$_{-6.0}$	32.3	47.3$_{-0.1}$	–	–
Video-TwG (Chen et al., 2026)	46.6$_{-18.5}$	–	–	–	44.4$_{-3.0}$	–	–
SDRL (Wang et al., 2026a)	56.5$_{-8.6}$	72.5$_{+0.8}$	63.3$_{-6.3}$	29.1	48.4$_{+1.0}$	–	–
Video-ToC (Tan et al., 2026)	59.6$_{-5.5}$	72.9$_{+1.2}$	59.2$_{-10.4}$	31.4	50.4$_{+3.0}$	–	–
Max Δ	19.5	5.0	10.6	8.6	6.8	8.8	11.1
Base model: Qwen3-VL-8B-Instruct							
Qwen3-VL-8B-Instruct (Bai et al., 2025a)	71.4	–	68.7	–	65.3	–	58.0
LLaVA-OneVision-2 (An et al., 2026)	71.4	74.3	69.0$_{+0.3}$	–	–	68.0	58.0
VSeek (Goel et al., 2026)	63.7$_{-7.7}$	–	–	–	–	54.5	38.7$_{-19.3}$
OneThinker (Feng et al., 2026)	63.3$_{-8.1}$	–	–	–	63.3$_{-2.0}$	61.7	–
VideoKR (Fu et al., 2026)	68.2$_{-3.2}$	–	67.9$_{-0.8}$	–	61.8$_{-3.5}$	61.6	–
InternVideo3 (Yan et al., 2026b)	71.4	–	–	–	–	62.4	58.0
Max Δ	8.1	–	1.1	–	3.5	13.5	19.3

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

1 INTRODUCTION

Benchmarks measure progress in video understanding, with researchers often using benchmark scores to claim improvements over prior methods or base models. Such claims assume that scores for the same checkpoint on the same benchmark are fairly comparable across evaluations and papers. We find that they are not. Table 1 compares scores reported in recent papers for two widely used checkpoints, Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B-Instruct (Bai et al., 2025a), across seven video benchmarks. Several methods report gains over Qwen2.5-VL-7b baseline that fall below our reproduction of the same baseline. When we evaluate these methods under identical settings, these don't improve consistency over the baseline across varying frame counts. We see that reported performance varies substantially across papers and can sometimes even be up to 19.5pp (in the case of VideoMME) below the scores from the original Qwen papers. These differences far exceed the 2–5pp gains commonly showcased by these same papers over the base model. This makes it hard to tell whether an apparent gain stems from a real capability improvement in the model, a difference in evaluation configuration, or both.

This shows that video evaluation can be especially sensitive to eval configuration. For example, **Frame sampling** determines which events are visible to a model. Which **Video decoder** is used, how **Number of frames** and **Frame resolution** are set determine the detail and temporal structure visible to the model. Further, the **Input modality** (input a video file or sequence of frames) and text **Prompt** can also affect performance. Yet, papers use evaluation frameworks like `lmms_eval` as *black boxes*, leaving such details underspecified. Additionally, the quality of benchmark samples can affect the value and cost of evaluation. Some benchmark samples can be answered without seeing the video.Further, because model rankings can differ across benchmarks, robust comparisons demand aggregating evaluations on multiple datasets, with very large inference costs. As such, a reliable evaluation that supports fast model development calls for (1) a clean set of benchmark samples, (2) low evaluation cost while preserving ranking, and (3) a standardized evaluation protocol.

To address (1) and (2), we introduce **Meta-Video** (Sec. 3), a compact video understanding benchmark of 1,000 questions over 791 videos, balanced across source datasets and spanning diverse tasks and durations. We construct it by filtering out low-quality samples to further improve evaluation fidelity and computational efficiency (Joshi et al., 2026; Ghosh et al., 2025). Specifically, we remove questions solvable without the video, since these cannot measure video understanding. **Meta-Video** preserves model rankings over its source benchmarks, separates models more sharply, and requires $5\times$ less inference compute. To tackle (3), we identify key parameters in video evaluation pipelines and systematically measure their effects on performance (Sec. 4). We find that input representation and inference backend substantially change scores. In particular, switching between vLLM and Hugging Face while keeping their respective default settings can change performance by more than 5pp, with gains vanishing once the defaults are fixed. Further, we show that extracted `.jpg` frames yield higher scores when few frames are sampled, while `.mp4` input performs better with 64 or more frames. We observe that how many frames and which frames (frame sampling) the model sees is one of the largest, most consistent factors influencing performance. Our systematic evaluation identifies the major impact factors and proposes a concise way to compile them in the form of evaluation cards to allow for a better, transparent, and reproducible practice in video understanding benchmarking.

Our contributions are: (1) We introduce **Meta-Video**, a compact, balanced benchmark that preserves ranking trends across its source benchmarks (reliability), uses one-fifth of the inference compute (efficiency), and makes differences between models clearer (discriminability). (2) We examine the current state of video understanding evaluation and show that variation in often unspecified evaluation protocols can explain substantial discrepancies in reported scores. (3) We release a revised evaluation suite and an extensive set of reference configurations to improve reproducibility and support future comparisons. Together, our findings and standardized evaluation protocols offer a transparent and less costly way to evaluate video understanding.

2 RELATED WORK

Video Understanding Models. General purpose vision-language models have the capability to ingest multi-frame input and hence can perform video understanding tasks. Models such as Qwen2.5-VL (Bai et al., 2025b), InternVL (Zhu et al., 2025), LLaVA-OneVision (Li et al., 2024a; An et al., 2025; 2026), PerceptionLM (Cho et al., 2026), Molmo (Deitke et al., 2025), LLaVA-Next-Interleave (Li et al., 2024b) etc., are widely used as standard open-weight VLMs. Other works enable such

models to interpret and reason over dynamic visual content (Shu et al., 2025; Cheng et al., 2024; Yan et al., 2024; Chen et al., 2024; Li et al., 2024d; Zhang et al., 2024; Maaz et al., 2024a;b; Yang et al., 2025; Zhu et al., 2024). More recently, video reasoning models have surfaced with the ability to reason over temporal and causal relationships across sequences of frames (Meng et al., 2025a; Wang et al., 2025d; Sun et al., 2025; Zhang et al., 2025b; Dang et al., 2025; Wang et al., 2025a).

Video Understanding Benchmarks. Multiple benchmarks have been proposed to evaluate video understanding e.g., VideoMME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), MVBench (Li et al., 2024c), LVBench (Wang et al., 2025b) etc. Further, several benchmarks target reasoning, such as, VideoMMMU (Hu et al., 2025), VideoMathQA (Rasheed et al., 2026), SciVideoBench (Deng et al., 2025), VideoReasonBench (Liu et al., 2026), VCRBench (Qi et al., 2025), TempCompass (Liu et al., 2024b), MMVU (Zhao et al., 2025), MLVU (Zhou et al., 2025), Minerva (Nagrani et al., 2025), EgoSchema (Mangalam et al., 2023) etc. However, these benchmarks have key limitations (Udandarao et al., 2025; Yang et al., 2026). For e.g., many questions can be answered *without* any visual information (Fu et al., 2026), as we show in Sec. A. In addition, because of their large size, evaluation is extremely time-consuming, discouraging rapid iteration for fast model development.

Validity of Benchmark Evaluations. Recent works have tried to improve the validity of benchmarks. Answer-Matching (Chandak et al., 2025) reformulates multiple-choice tasks into a generative setting, MMBench (Liu et al., 2024a) uses circular evaluation techniques, VMCBench automatically scales up question generation process (Zhang et al., 2025c), OneBench proposes aggregating heteregeneous measurements across benchmarks (Ghosh et al., 2025), and DataComp-VLM (Farina et al., 2026) uses stability-monotonicity filtering. DatBench (Joshi et al., 2026) proposed a cleaned evaluation suite that expedites evaluation and maintains discriminative power. Sober Reasoning (Hochlehnert et al., 2025) provides an empirical evaluation of mathematical reasoning benchmarks showcasing sensitivity to various implementation choices like random seed, prompt formatting, and hardware configurations. All these prior works focus mainly on language and image domains, with the video domain taking a back seat. Further, evaluation repositories such as `lmms_eval` (Zhang et al., 2025a) and `VLMEvalKit` (Duan et al., 2024) provide a framework to evaluate various multimodal benchmarks with multiple models. However, they are often used as black boxes, which makes results hard to compare. We close this gap with a standardized evaluation protocol for video and a new benchmark, **Meta-Video**, built for discriminability and fast iteration.

3 META-VIDEO: AN EFFICIENT AND DISCRIMINATIVE BENCHMARK

Evaluating video models across multiple benchmarks and configurations to ensure robustness is expensive, particularly for researchers with limited compute. Reliable comparisons also require questions that test video understanding: questions that models can answer without visual input can inflate scores and obscure differences between models. To address these problems, we introduce **Meta-Video**, a compact benchmark of videos and question-answer pairs selected from existing datasets. while reducing inference cost.

3.1 SOURCE BENCHMARKS AND CONSTRUCTION

We select video QA pairs from the evaluation sets of Video-MME, LongVideoBench, LVBench, VideoMMMU, and VSI-Bench, which cover a range of video types, durations, and question categories. To filter questions that may be answerable without visual input, we replace all frames with black frames and evaluate Qwen2.5-VL-7B-Instruct, Qwen3-VL-8B-Instruct, and InternVL3.5-8B. We discard each question that all three models answer correctly, then sample QA pairs from the remaining questions in each source benchmark (see Appendix A for details).

3.2 STATISTICS

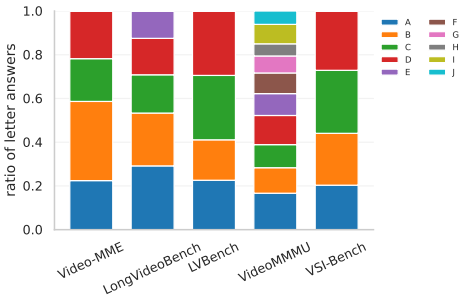
Question types. Meta-Video contains 1,000 QA pairs from 788 distinct videos, spanning multiple domains and question types, as shown in Figure 1; Figures 1a and 1b. Of these, 937 are multiple-choice questions and 63 require numerical free-form answers. Figure 1c shows the distribution of correct answer options, with an even distribution across options. The benchmark includes 564 questions with four options, 185 with five, 170 with more than five, of which 157 questions have exactly ten options, and 18 questions with fewer than four options. Random guessing on multiple-choice questions and answering 0 on numerical questions yields a score of 20.3. Always selecting option B, the most frequent correct option, and answering 0 on numerical questions yields 23.3.



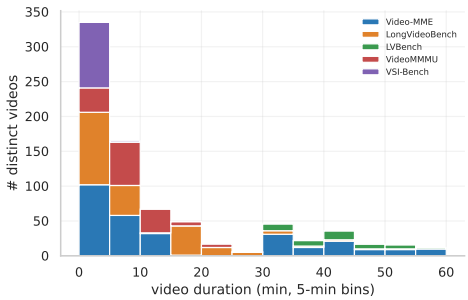
(a) Source benchmarks and their categories.



(b) Question types within each source benchmark.



(c) Distribution of correct answer letters per benchmark (63 non-letter answers excluded).



(d) Video duration distribution on the **Meta-V**.

Figure 1: **Meta-V statistics.** The benchmark contains 1,000 QA pairs over 791 distinct videos.

Video duration. Meta-Video spans video durations up to one hour, as illustrated in Figure 1d. The duration groups comprise 335 short videos (< 5 minutes), 281 medium videos (5–20 minutes), and 175 long videos (20–60 minutes). This range matters because benchmarks described as testing long-video understanding differ substantially in duration: videos average 41.1 minutes in the Video-MME long split, 7.9 minutes in LongVideoBench, and 67.3 minutes in LVBench. Appendix A provides duration statistics for each source benchmark.

Results. Table 2 evaluates models under different parameter settings and compares **Meta-Video** with the average across all the benchmarks. **Meta-Video** follows the trend of the average of multiple benchmarks, highlighting the fact that using **Meta-Video** can show the trend for any new model to be tested, with the additional advantage of being very efficient compared to running all benchmarks at all settings, which is computationally expensive.

4 Standardizing Evaluations for Video Understanding

Evaluation choices can change both model scores and rankings. We examine these choices at three stages: inference and decoding (Sec. 4.1), frame extraction (Sec. 4.2), model input (Sec. 4.3), and prompt design (Sec. 5.2). Based on this analysis, we fix the most influential factors to obtain a controlled and reproducible evaluation setting (Sec. 4.4), while documenting detailed configurations in evaluation cards. Finally, we provide a comprehensive evaluation across seven benchmarks and six backbones (Sec.5.1).

4.1 Inference and Decoding

Inference backend. Video question-answering frameworks such as `lmms_eval` support multiple inference backends, including the reference Hugging Face (HF) implementation Hugging Face (2026) and optimized engines such as vLLM (Kwon et al., 2023). These backends can differ in both

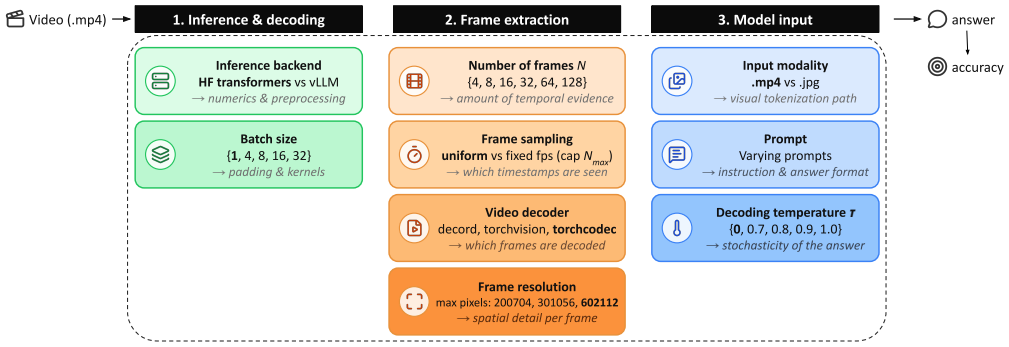


Figure 2: **Evaluation framework.** Parameters in the evaluation pipeline that affect reported accuracy for video understanding benchmarks.

numerical computation and preprocessing. In our experiments, HF produces identical scores across repeated runs with fixed seeds, whereas vLLM exhibits significant variations even with fixed seeds, potentially due to batching.

Batch size. Batch composition and size can affect padding and kernel selection, changing outputs even under greedy decoding. Although batch size has little effect on aggregate performance in our experiments, it remains a source of variation and should be reported. Table 2(a) compares the impact of inference backends and batch sizes. Scores within each framework are stable, with a mean standard deviation of 0.15 percentage points, but scores between backends differ by 4–6 points on TempCompass, MVBench, and **Meta-Video**. On **Meta-Video**, for example, the same model scores approximately 24% with HF and 30% with vLLM. We find that this discrepancy is attributable to the evaluated vLLM implementation converting video inputs into individual images, bypassing the Qwen models' spatiotemporal pooling. We provide a fix lmms_eval_video implementation restores this pooling and aligns vLLM preprocessing with HF. Appendix A describes the fix and its results.

4.2 Frame Extraction

Number of frames and **Frame sampling.** The number of input frames determines how much of a video the model sees and has one of the largest effects on accuracy. Uniform sampling selects exactly N frames from a video of duration T, independent of T, so temporal coverage becomes sparser as duration T increases. Fixed-rate sampling at r frames per second selects approximately rT frames, maintaining temporal density while allocating more frames to longer videos.

We find that the evaluated lmms_eval implementation exposes a maximum frame count, $N_{\max}$, but its backends do not enforce it, which can cause out-of-memory errors on long videos. Our implementation enforces this limit: when it is set, the model receives $\min(\lfloor rT \rfloor, N_{\max})$ frames. If $\lfloor rT \rfloor \geq N_{\max}$, the framework switches to uniform sampling over the full video. Thus, fixed-rate sampling maintains temporal density until the cap is reached. We also support clipping the video based on $N_{\max}$; Appendix A provides details.

We observe that with uniform sampling, increasing the frame count from 4 to 128 raises accuracy on Video-MME from 50.0% to 65.5% and on **Meta-Video** from 24.80% to 40.75% (Table 2(b)). Gains on TempCompass and MVBench saturate or reverse beyond 64 frames, showing that additional frames do not always improve performance. These results show that frame count is one of the strongest determinants of performance across the tested range of 4 to 128 uniformly sampled frames.

Sampling strategy also affects accuracy (Table 2(c)). Sampling at 1 fps instead of uniformly has no effect on Video-MME in this comparison. On the shorter clips in TempCompass and MVBench, it reduces both the number of frames and runtime, but lowers accuracy by 3.4–5.0 pp. Uniform sampling performs better on average, suggesting that dense sampling can help even on short videos.

Video decoder. Video decoders such as decord Decord; DMLC (2019), torchvision TorchVision (2016), and torchcodec PyTorch Foundation (2024) can differ in timestamp calculation, variable-frame-rate handling, and keyframe resolution. They may therefore extract different frames under the same sampling settings.

In our evaluation, decoder choice changes scores but does not affect the overall model ranking (Table 2(c)). Decord performs slightly better than torchcodec, while torchvision produces larger drops

Table 2: **Sensitivity of evaluation settings and parameters on Qwen2.5-VL-7B.** Each sub-table varies one setting; all other settings are fixed as listed. All runs use the official `lmms_eval` prompt. All values are reported as accuracy (%), except *Runtime*, which gives the total runtime in minutes of all runs in the block above them. Avg. is the mean accuracy over the four benchmarks; in the *Runtime* rows, Avg. gives the summed runtime over all benchmarks. In (f), LongVideoBench was not run for $T>0$, so Avg. there is over the other three benchmarks († including $T=0$, for comparability). $\pm$ denotes the standard deviation over repeated runs. Column labels: nf (number of frames), bck (backend), bs (batch size), rdr (video reader), px (max pixels), samp (sampling; $\leq n$ means 1 fps capped at n frames), T (temperature).

(a) **Inference backend** ★ and **Batch size**

Fixed: 602k px, uniform sampling, torchcodec, `.mp4`, $T=0$, 4 frames

bck	bs	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
HF	1	50.0	64.4	59.2	50.3	56.0	**24.8**
	4	50.0	64.7	59.6	50.5	56.2	24.4
	8	50.0	65.0	59.3	50.7	56.2	24.1
	16	50.1	64.8	59.2	50.4	56.1	24.6
	32	50.0	64.9	59.3	50.4	56.1	24.6
Runtime		*157*	*46.6*	*166*	*77.2*	*447*	133
vLLM	1	50.4$_{\pm 0.1}$	69.$_{\pm 0.1}$	64.4$_{\pm 0.1}$	52.1$_{\pm 0.1}$	58.9	**30.4**$_{\pm 0.1}$
	4	50.3$_{\pm 0.0}$	69.2$_{\pm 0.2}$	64.4$_{\pm 0.0}$	51.9$_{\pm 0.2}$	59.0	29.9$_{\pm 0.2}$
	8	50.2$_{\pm 0.2}$	69.2$_{\pm 0.2}$	64.3$_{\pm 0.2}$	51.9$_{\pm 0.1}$	58.9	30.0$_{\pm 0.5}$
	16	50.3$_{\pm 0.1}$	69.2$_{\pm 0.4}$	64.4$_{\pm 0.2}$	51.7$_{\pm 0.0}$	58.9	29.7$_{\pm 0.3}$
	32	50.3$_{\pm 0.1}$	69.1$_{\pm 0.1}$	64.3$_{\pm 0.1}$	51.8$_{\pm 0.1}$	58.9	29.7$_{\pm 0.1}$
Runtime		*155*	*55.6*	*257*	*84*	*551*	112

(b) **Input modality** ★ and **Number of frames** ★

Fixed: 602k px, uniform sampling, HF, batch 1, $T=0$; `.mp4`; torchcodec

in	nf	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
`.mp4`	4	50.0	64.4	59.2	50.3	56.0	24.8
	8	53.3	70.3	63.4	52.7	59.9	28.2
	16	58.0	71.8	66.7	56.1	63.1	33.4
	32	61.4	73.2	68.5	58.7	65.5	36.1
	64	63.5	74.1	69.0	60.2	66.7	39.2
	128	65.5	73.4	68.0	60.2	**66.8**	**40.8**
Runtime		*1447*	*356*	*1095*	*777*	*3675*	695
`.jpg`	4	49.7	69.4	64.0	51.5	58.7	27.8
	8	55.7	71.4	64.8	55.1	61.8	31.3
	16	58.9	70.6	63.6	56.9	62.5	37.0
	32	61.2	70.1	62.2	57.2	62.7	37.1
	64	62.6	68.3	61.5	58.1	62.6	**38.9**
	128	62.7	66.8	60.6	57.5	61.9	36.5
Runtime		*2495*	*663*	*2329*	*1337*	*6824*	1021

(c) **Frame sampling**

Fixed: 602k px, HF, batch 1, torchcodec, `.mp4`, $T=0$

samp	nf	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
Uniform	32	61.4	73.3	68.6	58.7	65.5	36.1
	64	63.5	74.2	69.0	60.2	66.7	**39.2**
Runtime		*542*	*134*	*413*	*286*	*1375*	356
1 fps	≤ 32	61.5	69.9	64.1	57.7	63.3	34.9
	≤ 64	63.5	69.9	64.0	58.7	64.0	**39.4**
Runtime		*542*	*37.7*	*135*	*238*	*953*	241

(d) **Video decoder**

Fixed: 602k px, uniform sampling, HF, batch 1, `.mp4`, $T=0$, 4 frames

rdr	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
decord	50.1	64.5	59.4	50.4	56.1	**25.2**
torchvision	50.4	57.5	49.1	48.7	51.4	24.9
torchcodec	50.1	64.4	59.2	50.3	56.0	24.8
Runtime	*100*	*30.0*	*104*	*44*	*278*	97.9

(e) **Frame resolution**

Fixed: uniform sampling, HF, batch 1, torchcodec, `.mp4`, $T=0$, 4 frames

px	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
200,704	49.4	65.5	59.3	50.1	56.1	24.9
301,056	50.1	64.5	59.0	50.5	56.0	**25.2**
602,112	50.1	64.5	59.2	50.3	56.0	24.9
Runtime	*82.5*	*30.8*	*101*	*40.7*	*255*	95.3

(f) **Decoding temperature**

Fixed: 602k px, uniform sampling, HF, batch 1, torchcodec, `.mp4`, 4 frames

T	Video-MME	TempComp	MVBench	LongVideoB	Avg.	Meta-V
0	50.0	64.4	59.2	50.3	55.9†	24.8
0.7	47.1$_{\pm 0.4}$	65.0$_{\pm 3.0}$	56.9$_{\pm 0.5}$	47.2$_{\pm 1.1}$	54.0	**25.9**$_{\pm 1.1}$
0.8	46.3$_{\pm 1.3}$	63.9$_{\pm 3.9}$	56.1$_{\pm 0.8}$	45.9$_{\pm 1.2}$	53.0	25.5$_{\pm 1.4}$
0.9	45.6$_{\pm 1.1}$	62.8$_{\pm 3.5}$	55.6$_{\pm 0.2}$	43.4$_{\pm 0.0}$	54.7	25.3$_{\pm 0.3}$
1.0	44.9$_{\pm 0.3}$	61.5$_{\pm 3.9}$	54.4$_{\pm 0.2}$	43.3$_{\pm 1.0}$	51.0	24.9$_{\pm 1.2}$
Runtime	*172*	*62.1*	*169*	*58.7*	*461*	169

on some datasets. We use torchcodec, the default in the evaluated `lmms_eval` implementation, and recommend reporting the decoder to support reproducibility.

Frame resolution. Before tokenization, frames are resized so that their area does not exceed `max_pixels`. This budget controls spatial details and, for models with dynamic-resolution encoders, the number of visual tokens per frame. Different default budgets can therefore give the same model substantially different inputs at the same frame count.

We evaluate the impact of the frame resolution strategy (Table 2(e)). We observe among the budgets tested, 200,704 pixels yields the highest average accuracy across benchmarks, while 301,056 performs best on **Meta-Video** (Table 2(e)). However, the variation is minor and frame resolution does not impact perforamnce significantly.

4.3 Model Input

Input modality. Sampled frames can be supplied as video input or as a sequence of images, and many VLMs process these modalities through different branches. The video branch may apply spatiotemporal pooling, merging patches across frames into shared visual tokens. The pooling scheme depends on the architecture: Qwen2.5 pools pairs of frames, producing half as many visual tokens as its image branch. In the implementations we evaluated, HF uses the video branch, whereas vLLM converts video inputs into individual images and bypasses temporal pooling. Appendix A describes our fix. Table 2(b) compares direct `.mp4` input with pre-extracted `.jpg` frames. Input modality changes accuracy by 4–6 percentage points in some settings. This recovers the gap between vLLM and HF in Table 2(a). Image input can perform better at low frame counts, gaining 5.0 points on TempCompass at four frames, but its gains flatten or reverse at higher frame counts. At 128 frames on MVBench, image input scores 60.6%, compared with 68.0% for video input, and takes 2,329

Evaluation Card (Table *ref*)

SCOPE	Models	Model-A-7B, Model-B-13B
	Datasets	Dataset-1, Dataset-2
RUNTIME	Inference backend	backend-name
	Batch size	{1, 8, 32}
INPUT	Input modality	.mp4
	No. of frames	{8, 16, 32}
	Frame sampling	**uniform** | *fps rate*
	Max frames	**none** | *int*
	Cap mode	*none* | *uniform* | **cutoff**
DECODING	Video decoder	decoder-name
	Frame resolution	448 *(max pixel count)*
	Temperature	0.0
	Prompt layout	*vt* | **tvt** | *vtv*
	Prompt(s)	`<video> Answer the question using a single word or phrase. {question}`
COMMENTS	Notes	Free-text remarks, e.g. deviations from the default setup, known issues, or hardware used (8×A100).

Table 3: **Evaluation card.** Reporting the settings that affects a score makes results reproducible and comparable across papers. Options in **bold** are the ones used; a set $\{\cdot\}$ denotes a parameter sweep.

rather than 1,095 minutes. The lower visual-token count from spatiotemporal pooling may help explain the advantage of video input at larger frame budgets.

Prompt. Evaluation prompts specify both the task and the required answer format, affecting predictions as well as whether those predictions are parsed correctly. Prompt layouts also vary: some place text before or after the video, while others repeat text or visual content to change which earlier inputs each modality can attend to under causal attention, resulting in prompt patterns such as T-V-T or V-T-V. We evaluate these choices in Section 5.2.

Decoding temperature. Sampling temperature affects accuracy even when the prompt requests only a single option letter. Nonzero temperatures introduce stochasticity and can increase variation across runs. We observe from Table 2(f) that $\tau = 0$ performs the best among temperatures we test. They also affect some datasets more than others: on Video-MME, accuracy falls from 50.0% at $\tau = 0$ to 44.9% at $\tau = 1$.

4.4 Best Practices: Evaluation Cards

As seen in above sections, VLMs are highly sensitive to evaluation choices, but are rarely reported in full. To make evaluations are stable and reproducible, we fix a reference configuration and vary one factor, keeping all other settings fixed. Inspired by dataset cards Pushkarna et al. (2022), we introduce *evaluation card* that summarizes the complete configuration behind its numbers. For each parameter varied within the table, all the other settings are fixed (Template in 3). We provide an evaluation card for every table in this paper in Appendix A.

5 Evaluation

5.1 Standardized Comparison of Baseline Models

Our ablations on the effect of different parameters show that evaluation settings can substantially change model scores. We therefore compared the performance of five common open MLLMs while controlling backend, resolution, temperature, and other settings. To assess the best model performance, we vary the frame budget, one of the strongest determinants of performance. This comparison reveals both how well models perform and how they use additional visual input.

Setup. We compare models at 4, 8, 16, 32, 64, and 128 frames on seven source benchmarks—Video-MME, LongVideoBench, TempCompass, MMVU, LVBench, VSI-Bench, and MLVU—and on **Meta-Video**. The models are Qwen2.5-VL-7B Bai et al. (2025b), Qwen3-VL-3B-Instruct and Qwen3-VL-8B-Instruct Bai et al. (2025a), and InternVL3.5-4B and InternVL3.5-8B Wang et al. (2025c). Each InternVL3.5 frame is encoded as one tile. Table 10 gives the full evaluation card; scores reported in the original model papers appear in gray.

Table 4: Re-evaluated vs. reported scores. For each model, the upper rows give our own evaluation under a single fixed protocol as a function of the frame budget, with all other settings held fixed; the shaded row at the bottom of each block reproduces the numbers as reported in the literature, placed next to our largest frame budget for direct comparison. Avg. is the mean over the seven benchmarks; **Meta-V** is highlighted in blue and reported separately for short, medium and long videos, together with their unweighted mean. **Bold** marks the best frame budget per model in each column. – denotes an entry that is not reported or not evaluated. Evaluation card is provided in 10.

Model	#Frames	Video-MME	Long-VideoBench	Temp-Compass	MMVU (Val)	LV-Bench	VSI-Bench	MLVU	Avg.	Meta-V			
										Short	Medium	Long	Avg.
InternVL3.5-4B	4	54.5	52.1	69.1	46.4	38.0	45.4	59.5	52.1	35.4	24.2	32.3	30.6
	8	57.7	54.5	68.9	46.7	38.9	55.0	60.6	54.6	45.0	32.7	35.1	37.6
	16	60.7	56.8	71.1	48.6	41.7	58.5	63.4	57.3	50.4	36.2	37.9	41.5
	32	63.1	58.3	**71.3**	**50.8**	42.6	**63.4**	67.1	**59.5**	**53.9**	41.2	41.5	**45.5**
	64	**64.8**	**59.1**	69.8	49.9	43.1	58.8	67.7	59.0	52.7	41.1	40.1	44.6
	128	64.1	58.8	68.9	49.3	**44.7**	56.0	**68.4**	58.6	48.1	**41.4**	**42.2**	43.9
	rep. Wang et al. (2025c)	65.4	60.8	–	–	–	–	70.4	–	–	–	–	–
Qwen3-VL-4B	4	49.0	51.5	67.8	48.5	32.5	37.1	52.7	48.4	33.0	21.4	31.4	28.6
	8	52.9	54.4	69.4	51.8	35.4	47.9	55.0	52.4	42.5	30.5	36.1	36.4
	16	57.0	59.0	70.3	51.4	37.2	52.9	58.1	55.1	52.2	42.7	37.9	44.3
	32	61.4	60.2	**70.9**	52.7	40.9	54.4	62.5	57.6	58.6	49.6	41.8	50.0
	64	64.1	61.3	70.6	53.2	40.5	61.1	66.1	59.6	61.7	53.5	**43.2**	52.8
	128	**67.5**	**64.0**	69.9	**53.4**	**43.8**	**61.3**	**70.4**	**61.5**	**67.3**	**57.1**	41.4	**55.3**
	rep. Bai et al. (2025a)	69.3	–	–	50.5	56.2	–	75.3	–	–	–	–	–
Qwen2.5-VL-7B	4	50.1	50.3	64.5	46.2	28.1	26.7	51.9	45.4	26.7	22.7	25.1	24.8
	8	53.4	52.7	70.4	48.4	33.1	30.5	54.5	49.0	30.2	26.6	27.9	28.2
	16	58.1	56.2	71.8	50.8	34.7	33.4	56.3	51.6	35.1	36.0	29.0	33.4
	32	61.4	58.7	73.3	49.7	37.4	36.7	59.5	53.8	42.1	33.3	29.3	34.9
	64	63.5	**60.2**	**74.2**	51.8	39.3	37.6	63.4	55.7	44.9	40.1	33.6	39.5
	128	**65.6**	**60.2**	73.4	**51.9**	**42.1**	**38.3**	**66.1**	**56.8**	**46.1**	**41.3**	**34.3**	**40.6**
	rep. Bai et al. (2025b)	65.1	56.0	71.7	50.1	45.3	–	70.2	–	–	–	–	–
InternVL3.5-8B	4	56.7	55.0	69.4	55.7	38.7	47.6	58.2	54.5	38.3	31.4	35.3	35.0
	8	60.0	56.4	70.2	55.4	39.6	56.3	62.5	57.2	46.5	40.7	41.3	42.8
	16	63.0	57.5	70.6	57.7	40.6	57.0	63.7	58.6	50.4	46.1	43.5	46.7
	32	64.7	59.9	**70.8**	56.4	41.8	**64.5**	67.7	60.8	56.0	48.3	47.0	50.4
	64	**66.2**	**62.9**	70.5	**59.9**	43.8	59.8	**68.5**	**61.7**	**59.1**	50.9	46.3	52.1
	128	65.1	61.6	70.1	58.3	**46.2**	56.6	**68.5**	60.9	55.1	**51.2**	**50.5**	**52.3**
	rep. Wang et al. (2025c)	66.0	62.1	–	–	–	–	70.2	–	–	–	–	–
Qwen3-VL-8B	4	53.0	52.2	70.7	53.8	33.6	37.1	53.0	50.5	37.1	28.8	41.7	35.9
	8	56.1	55.9	72.3	56.7	36.0	45.4	54.7	53.9	44.6	36.9	44.2	41.9
	16	60.3	58.0	73.2	58.4	37.4	53.5	59.2	57.1	56.4	44.7	49.8	50.3
	32	64.6	59.5	**74.4**	**59.5**	39.8	59.2	63.1	60.0	61.5	51.9	51.6	55.0
	64	66.5	63.0	73.2	58.5	42.9	**62.7**	68.2	62.1	**65.5**	55.8	50.9	57.4
	128	**69.2**	**63.6**	74.2	58.7	**46.5**	59.9	**72.1**	**63.5**	62.5	**60.9**	**53.4**	**58.9**
	rep. Bai et al. (2025a)	71.4	–	–	58.7	58.0	–	78.1	–	–	–	–	–

Results. We report results in Table 4. Overall, it shows that Qwen3-VL achieves the best performance among the backbones in the 4B as well as in the 7-8B range in the given setting.

Comparing the resulting score with previously reported results, we see that on Video-MME, our results are similar to the originally reported scores. For Qwen2.5-VL-7B, our best configuration exceeds the reported scores on Video-MME, LongVideoBench, TempCompass, and MMVU. On LVBench, however, Qwen3-VL-4B scores 12.4 points below its reported result. Sparse coverage of the benchmark's long videos under our 128-frame cap may explain this gap.

Frame count also changes model rankings. The Qwen models improve almost monotonically up to 128 frames, whereas InternVL3.5 peaks at 32–64 frames and then declines. On VSI-Bench, for example, InternVL3.5-4B falls from 63.4 at 32 frames to 56.0 at 128. Across the seven source benchmarks, InternVL3.5-4B outperforms Qwen3-VL-4B at every frame count up to 32, including 52.1 versus 48.4 at four frames. The ranking reverses at 64 frames; at 128 frames, Qwen3-VL-4B leads by 2.9 points (61.5 versus 58.6). InternVL3.5-8B has the highest average score at 4–32 frames.

Meta-Video captures the broad performance trends observed across the source benchmarks: accuracy generally rises with frame count, while InternVL3.5-4B peaks earlier and then declines. These results support its use as a lower-cost comparison of the evaluated models and configurations.

Frame count and video duration on Meta-Video. Table 4 also reports performance on **Meta-Video** by video duration. The Qwen models gain most between 4 and 32 frames. InternVL3.5-8B gains only 0.2 points from 64 to 128 frames, while InternVL3.5-4B peaks at 45.5 with 32 frames and falls to 43.9 at 128. Short and medium videos benefit most from additional frames. Qwen3-VL-8B gains up to 32.1 points on medium videos. Gains on long videos are smaller: Qwen2.5-VL-7B gains 9.2 points on long videos, compared with 19.4 on short videos. Even on short videos, however, ac-

Table 5: Meta-V: Prompt, input modality and layout ablation. Average is the weighted mean of short, medium and long splits. Evaluation card is provided in 11

.

Prompt	Input modality		Layout	Number of frames	Max new token	Meta-V			
	.mp4	.jpg				Small	Medium	Long	Avg.
$P0$ (default)	✓	✗	–	4	default	26.4	22.3	24.7	24.6$_{-2.46}$
$P1$ (think step-by-step)	✓	✗	–	4	1024	17.8	27.7	30.7	24.8$_{-2.26}$
P^* (single letter only)	✓	✗	–	4	default	26.0	27.4	27.9	**27.0**
$P0$ (default)	✗	✓	VT	4	default	30.7	26.5	27.9	28.5$_{-1.23}$
$P1$ (think step-by-step)	✗	✓	VT	4	1024	21.4	31.6	26.8	26.4$_{-3.36}$
P^* (single letter only)	✗	✓	VT	4	default	30.7	30.4	27.5	**29.7**
$P0$ (default)	✗	✓	TVT	4	default	30.8	27.1	26.8	28.5$_{-1.00}$
$P1$ (think step-by-step)	✗	✓	TVT	4	1024	21.3	29.5	27.9	25.9$_{-3.55}$
P^* (single letter only)	✗	✓	TVT	4	default	29.5	31.9	26.5	**29.5**
$P0$ (default)	✗	✓	VTV	4	default	31.3	27.7	27.2	28.9$_{-1.14}$
$P1$ (think step-by-step)	✗	✓	VTV	4	1024	21.9	30.7	27.2	26.4$_{-3.70}$
P^* (single letter only)	✗	✓	VTV	4	default	29.5	31.6	28.9	**30.1**

curacy saturates or declines beyond 32–64 frames for Qwen3-VL-8B and both InternVL3.5 models. Rankings on **Meta-V** also depend on the frame budget. InternVL3.5-4B outperforms Qwen3-VL-4B at four frames (30.6 versus 28.6) and eight frames (37.6 versus 36.4). From 16 frames onward, Qwen3-VL-4B leads, reaching 55.3 versus 43.9 at 128 frames. Qwen3-VL-4B also overtakes InternVL3.5-8B at 64 frames. Meaningful model comparisons therefore require controlled settings and an explicit frame budget.

5.2 Evaluation of Prompt design

Second, we compare the impact of different prompt wordings and architectures given a fixed setting. To this end, we consider both, additional instructions as well as text resp. video input repetitions.

Prompt wording. Table 5 compares three prompts at a fixed budget of four frames. $P0$ is the dataset-specific default prompt in lmms_eval. $P1$ asks for step-by-step reasoning: "`Think step by step about what happens in the video, then end your reply with Answer:`"; its output is capped at 1,024 tokens. P^* replaces the default instruction with "`Respond with a single letter and nothing else.`" Table 11 gives the full evaluation card.

P^* achieves the highest average score for both input modalities, improving over $P0$ by 2.46 points with `.mp4` input and 1.23 points with `.jpg` input. Step-by-step prompting does not improve average accuracy: $P1$ scores 2.26 points below P^* with video input and more than 3.36 points below it with image input.The effect varies by video duration. With video input, $P1$ performs best on long videos (30.74) but worst on short videos (17.80). At this four-frame budget, image input outperforms video input for every prompt. Appendix A provides further discussion.

Prompt layout. We also compare three orders of visual (V) and textual (T) content: visual input followed by text (VT), text before and after the visual input (TVT), and visual input before and after the text (VTV). Table 11 gives the evaluation card. P^* performs best under every layout and reaches its highest score, 30.10, with VTV. Layout has a smaller effect than wording: the average score for P^* varies by only 0.6 points (29.50–30.10), and the score for $P0$ varies by less than 0.5 points. $P1$ performs worst under every layout, though repeating text or visual content improves it slightly.

6 Conclusion

We propose **Meta-Video**, a compact video understanding benchmarking which is efficient, and version of the source benchmarks. This benchmarks helps us to evaluate models efficiently while still keeping the trends of original benchmarks. We propose an open evaluation framework built on top of lmms_eval for cleaner video understanding and reasoning model and benchmark evaluation. We provide an exhaustive evaluation with various configs on seven benchmarks. We proposed **Meta-Video** a new benchmark that is efficient yet maintains the model and benchmark trends. We also provide an analysis of effect of input layout and varying prompts on **Meta-Video**. In conclusion, we show that evaluation choices such as frame sampling, input layout, prompt, backend strongly affect video benchmark scores, and are often unreported. We hope the new benchmark and evaluation reporting will help to pave the way towards more transparent, reproducible evaluation in the field.

AI USE STATEMENT

We used LLMs to assist with code development, figure plotting, table formatting and writing polishing for grammar and syntax errors. All generated outputs were reviewed and revised by the authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The code will be released alongside the camera-ready version to reproduce all results mentioned in the paper. All datasets, prompts, instructions, and others are provided in the main paper or appendix for more details. The code is available at `https://github.com/anymz1234/lmms_eval_video`.

REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Answer matching outperforms multiple choice for language model evaluation. *arXiv preprint arXiv:2507.02856*, 2025.

Houlun Chen, Xin Wang, Guangyao Li, Yuwei Zhou, Yihan Chen, Jia Jia, and Wenwu Zhu. Think with grounding: Curriculum reinforced reasoning with video grounding for long video understanding. *arXiv preprint arXiv:2602.18702*, 2026.

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

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

Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. Perceptionlm: Open-access data and models for detailed visual understanding. *Advances in Neural Information Processing Systems*, 38:23475–23537, 2026.

Jisheng Dang, Jingze Wu, Teng Wang, Xuanhui Lin, Nannan Zhu, Hongbo Chen, Wei-Shi Zheng, Meng Wang, and Tat-Seng Chua. Reinforcing video reasoning with focused thinking. *arXiv preprint arXiv:2505.24718*, 2025.

Decord; DMLC. Decord: An efficient video loader for deep learning with smart shuffling. `https://github.com/dmlc/decord`, 2019.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104. IEEE, 2025.

Andong Deng, Taojiannan Yang, Shoubin Yu, Lincoln Spencer, Mohit Bansal, Chen Chen, Serena Yeung-Levy, and Xiaohan Wang. Scivideobench: Benchmarking scientific video reasoning in large multimodal models. *arXiv preprint arXiv:2510.08559*, 2025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Bo Fang, Yuxin Song, Haoyuan Sun, Xinyao Zhang, Qiangqiang Wu, Wenhao Wu, and Antoni B Chan. Viss-r1: Self-supervised reinforcement video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11190–11200, 2026.

Matteo Farina, Vishaal Udandarao, Thao Nguyen, Selim Kuzucu, Maximilian Böther, Andreas Hochlehnert, Adhiraj Ghosh, Marianna Nezhurina, Karsten Roth, Joschka Struber, et al. Datacomp-vlm: Improved open datasets for vision-language models. *arXiv preprint arXiv:2606.28551*, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, et al. Onethinker: All-in-one reasoning model for image and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5432–5443, 2026.

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. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Lin Fu, Zheyuan Yang, Yang Wang, Tingyu Song, Arman Cohan, and Yilun Zhao. Videokr: Towards knowledge-and reasoning-intensive video understanding. *arXiv preprint arXiv:2606.05259*, 2026.

Adhiraj Ghosh, Sebastian Dziadzio, Ameya Prabhu, Vishaal Udandarao, Samuel Albanie, and Matthias Bethge. Onebench to test them all: Sample-level benchmarking over open-ended capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32445–32481, 2025.

Harsh Goel, SP Sharan, Sahil Shah, Minkyu Choi, Joungbin An, Kristen Grauman, and Sandeep P Chinchali. Incentivizing vision language models to search for long video question answering. *arXiv preprint arXiv:2607.02959*, 2026.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos.(2025). *URL https://arxiv. org/abs/2501.13826*, 3, 2025.

Hugging Face. Hugging face – the ai community building the future. `https://huggingface.co`, 2026. Accessed: 2026-09-25.

Siddharth Joshi, Haoli Yin, Rishabh Adiga, Ricardo Monti, Aldo Carranza, Alex Fang, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, et al. Datbench: Discriminative, faithful, and efficient vlm evaluations. *arXiv preprint arXiv:2601.02316*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

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

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024c.

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

Yiming Liang, Yixiao Chen, Yiyang Zhou, Yixuan Wang, Shoubin Yu, Andong Deng, Fuxiao Liu, Qin Zhang, Chen Chen, Mohit Bansal, et al. Storm: Internalized modeling for spatial-temporal reasoning in video-language models. *arXiv preprint arXiv:2605.26014*, 2026.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024a.

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? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024b.

Yuanxin Liu, Kun Ouyang, Haoning Wu, Yi Liu, Lin Sui, Xinhao Li, Yan Zhong, Y. Charles, Xinyu Zhou, and Xu Sun. Videoreasonbench: Can mllms perform vision-centric complex video reasoning? In *International Conference on Learning Representations*, volume 2026, pp. 3475–3502, 2026.

Romy Luo, Zihui Sherry Xue, Alex Dimakis, and Kristen Grauman. When thinking drifts: Evidential grounding for robust video reasoning. *Advances in Neural Information Processing Systems*, 38:83696–83727, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 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 (Volume 1: Long Papers)*, pp. 12585–12602, 2024a.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arXiv:2406.09418*, 2024b.

Muhammad Maaz, Hanoona Rasheed, Fahad Shahbaz Khan, and Salman Khan. Video-r2: Reinforcing consistent and grounded reasoning in multimodal language models. *arXiv preprint arXiv:2511.23478*, 2025.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Desen Meng, Rui Huang, Zhilin Dai, Xinhao Li, Yifan Xu, Jun Zhang, Zhenpeng Huang, Meng Zhang, Lingshu Zhang, Yi Liu, et al. Videocap-r1: Enhancing mllms for video captioning via structured thinking. *arXiv preprint arXiv:2506.01725*, 2025a.

Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3-video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025b.

Arsha Nagrani, Sachit Menon, Ahmet Iscen, Shyamal Buch, Ramin Mehran, Nilpa Jha, Anja Hauth, Yukun Zhu, Carl Vondrick, Mikhail Sirotenko, et al. Minerva: Evaluating complex video reasoning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23968–23978. IEEE, 2025.

Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pp. 1776–1826, 2022.

PyTorch Foundation. torchcodec: Easy and efficient video decoding for pytorch. PyTorch Blog, December 2024. URL `https://pytorch.org/blog/torchcodec/`. Accessed: 2026-09-25.

Yukun Qi, Yiming Zhao, Yu Zeng, Xikun Bao, Wenxuan Huang, Lin Chen, Zehui Chen, Jie Zhao, Zhongang Qi, and Feng Zhao. Vcr-bench: A comprehensive evaluation framework for video chain-of-thought reasoning. *arXiv preprint arXiv:2504.07956*, 2025.

Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad Khan. Videomathqa: Benchmarking mathematical reasoning via multimodal understanding in video. In *International Conference on Learning Representations*, volume 2026, pp. 143037–143055, 2026.

Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 4305–4314. IEEE, 2025.

Guangzhi Sun, Yudong Yang, Jimin Zhuang, Changli Tang, Yixuan Li, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn-o1: Reasoning-enhanced audio-visual large language model. *arXiv preprint arXiv:2502.11775*, 2025.

Qizhong Tan, Zhuotao Tian, Guangming Lu, Jun Yu, and Wenjie Pei. Video-toc: Video tree-of-cue reasoning. *arXiv preprint arXiv:2604.20473*, 2026.

TorchVision. Torchvision. `https://github.com/pytorch/vision`, 2016.

Vishaal Udandarao, Shyamgopal Karthik, Surabhi S Nath, Andreas Hochlehnert, Matthias Bethge, and Ameya Prabhu. Solving spatial supersensing without spatial supersensing. *arXiv preprint arXiv:2511.16655*, 2025.

Peiyao Wang, Haotian Xu, Noranart Vesdapunt, Rui Hou, Jingyi Zhang, Haibin Ling, Oleksandr Obiednikov, Ning Zhou, and Kah Kuen Fu. Reinforcing structured chain-of-thought for video understanding. *arXiv preprint arXiv:2603.25942*, 2026a.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *arXiv preprint arXiv:2505.12434*, 2025a.

Shuning Wang, Zhiheng Wu, YiNuo Lu, Naiming Liu, Chen Jia, Bowen Liu, Shuo Nie, Weijie Zhu, and Yumeng Zhang. See more, think deeper: Query-expanded visual evidence and answer-clue guided reflection for long video understanding. *arXiv preprint arXiv:2606.09064*, 2026b.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025c.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025d.

Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025e.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. In *European Conference on Computer Vision*, pp. 98–115. Springer, 2024.

Ziang Yan, Yinan He, Xinhao Li, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1. 5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. *Advances in Neural Information Processing Systems*, 38:119152–119184, 2026a.

Ziang Yan, Sheng Xia, Jiashuo Yu, Yue Wu, Tianxiang Jiang, Songze Li, Kanghui Tian, Yicheng Xu, Yinan He, Kai Chen, et al. Internvideo3: Agentify foundation models with multimodal contextual reasoning. *arXiv preprint arXiv:2606.12195*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. In *International Conference on Learning Representations*, volume 2026, pp. 78185–78225, 2026.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025a.

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

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Tinyllava-video-r1: Towards smaller lmms for video reasoning. *arXiv preprint arXiv:2504.09641*, 2025b.

Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, et al. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29580–29590. IEEE, 2025c.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8475–8489. IEEE, 2025.

Yiwu Zhong, Zi-Yuan Hu, Yin Li, and Liwei Wang. Rethinking chain-of-thought reasoning for videos. *arXiv preprint arXiv:2512.09616*, 2025.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Appendix

CONTENTS

A Appendix **16**

A.1 **Input modality** 16

A.2 **Inference backend** 16

A.3 Original vLLM vs fixed vLLM 16

A.4 Uniform vs 1fps 17

A.5 Additional details about **Meta-V** 17

A.6 Base models vs Reasoning models 18

A.7 Video duration of long video benchmarks. 19

A.8 Evaluation datasets and settings 19

A.9 Evaluation cards 21

A APPENDIX

A.1 INPUT MODALITY

How `.mp4` is processed? For inputs with .mp4, the processor first samples frames based on `nframes` argument. It then resizes the frames based on the max_pixel. Further, frames are grouped in pairs due to temporal patch size which is specific for videos in Qwen2.5-VL models. Each frame is encoded separately to obtain X_{vision} tokens per frame, meaning that there are `nframes`$\times X_{vision}$ tokens. Then, for each 2x2 group of adjacent patch embeddings (based on the temporal patch size), a spatial merge happens. The processor inserts <|vision_start|>, then one <|video_pad|> with token id 151656 per merged token, then <|vision_end|>. So the total video tokens is $(\frac{\texttt{nframes}}{2}) \times \frac{H}{28} \times \frac{W}{28}$. It also uses M-RoPE with separate temporal, height, and width components.

How `.jpg` is processed? The wrapper converts the PIL image to data and resize based on `max_pixel` to multiples of 28. Instead of the true temporal dimension which was added in the case of .mp4 as input, in .jpg, each frame is duplicated, followed similar path as video processing but since the tokens are duplicated, the total token count become $\frac{H}{28} \times \frac{W}{28}$ with M-RoPE temporal index as constant for all image tokens. The image placeholder id is 151655.

A.2 INFERENCE BACKEND

Similar to `.mp4` and `.jpg`, the choice of inference backend `HF` vs `vLLM` also affects results because the two differ in how they process visual inputs. `HF` backend accepts video natively along with images, whereas `vLLM` by default, samples frames and encodes the frames as separate images. As a result, there is no spatio-temporal pooling with `vLLM`, which leads to difference in performance based on the backend used.

A.3 ORIGINAL VLLM VS FIXED VLLM

The issue with the original vLLM backend is that it decodes video with qwen_vl_utils and sends the sampled frames to vLLM as N separate images, so Qwen2.5-VL receives N <|image_pad|> blocks with no two-frame temporal merge, no video position ids, i.e., the video silently falls back into frames condition, which is the reason original vLLM runs around 6% slower than the HF backend on the same inputs. The fixed backend, passes the .mp4 as a file (`pass_video_url=True`in the function builder), so vLLM decodes it through the model's video processor and produces <|video_pad|> tokens, matching the input used in HF backend. In Table 6, we show the com-

Batch size	HF	vLLM original	vLLM fixed	Δ
1	24.81	31.00	26.25	−4.75
4	24.34	30.52	26.16	−4.35
8	24.01	30.62	26.86	−3.77
16	24.61	30.40	26.66	−3.74
32	24.62	30.47	27.22	−3.26

Table 6: In this table, we provide the scores of **Meta-V** after re-evaluating with the fixed vLLM, and compare it with the existing vLLM scores.

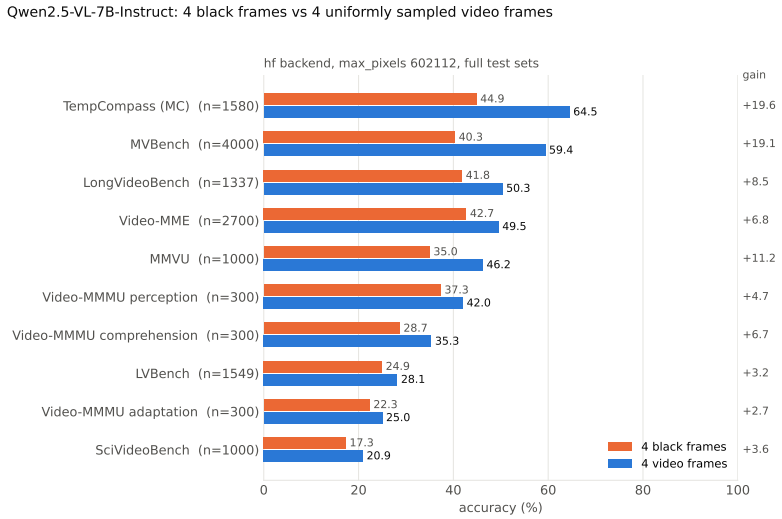


Figure 3: Blind baseline performance vs. four uniformly sampled frames, for Qwen2.5-VL-7b

parison of the HF runs, vLLM original scores (boosted scores), and vLLM fixed scores. It can be seen that the scores using the vLLM fixed version are closer to HF.

A.4 Uniform vs 1fps

As explained in the main paper, we re-implement 1fps as the original lmms_eval lacked the full implementation. We make a design choice when using 1fps. In the original lmms_eval, if the N_{max} argument was provided, it would ignore it for the cases when the duration of the video was longer than N_{max}, and would result in out of memory error. Hence, we re-implement this. In our implementation, if the 1fps is used along with N_{max} argument, all the videos which are less than N_{max} will be sampled at 1fps, but for the videos longer than N_{max}, a fallback of uniform N_{max} is done. One other way to implement this is by clipping all the frames that come after N_{max}. By doing so, we might end up not seeing the full video, and it could happen that the question asked is from the part of the video which was clipped off. Due to this issue we decided on using uniform N_{max} sampling as the fallback. To support this claim, on randomly sampled 100 videos from VideoMME long subset, we perform the clipped version vs uniform sampled version. The uniform sampled version with N_{max}=32 obtains 49.33 score whereas the clipped version obtains 47.33 score which is yy points lower. Hence we decided on uniform sampled version as the fallback. With 64 frames the uniform version is 52.33 score, whereas with the clipped version, we get 50.0 as score.

A.5 Additional details about **Meta-V**

In Figure 1, we show the statistics about our **Meta-V** dataset. **Meta-V** contains 1000 QA pairs on 788 unique videos, which are obtained from five source benchmarks. The source benchmarks are: VideoMME, LongVideoBench, LVBench, VideoMMMU, and VSIBench. In order to get a cleaner set of QA pairs, we first do a blind filtering. In this step, three models: Qwen2.5-VL-7b, Qwen3-VL-

Model	Video-MME	Long-VideoBench	Temp-Compass	MMVU (Val)	LV-Bench	MV-Bench	VideoMMMU			SciVideo-Bench
							Perc.	Comp.	Adapt.	
#Q	2700	1337	1580	1000	1549	4000	300	300	300	1000
Qwen2.5-VL-7B	42.7	41.8	44.9	35.0	24.9	40.3	37.3	28.7	22.3	17.3
Qwen3-VL-8B	39.0	41.6	37.9	39.7	15.2	31.2	26.7	29.7	26.3	23.1
InternVL3.5-8B	43.6	45.9	41.7	39.4	26.7	35.3	35.0	31.0	30.3	23.9

Table 7: **Blind baseline.** Scores when every input frame is replaced by a black frame, so the only usable signal is the question and answer text. All runs use 4 frames, HuggingFace inference, batch size 1, temperature 0; `max_pixels`= 301,056 for the Qwen models and `max_patches`= 12 for InternVL3.5. #Q is the number of questions evaluated.

	#Frames					
Model	4	8	16	32	64	128
Qwen2.5-VL-7B	49.5	53.8	58.7	**62.3**	**63.7**	**64.4**
Video-R1-7B	48.5 (-1.0)	52.0 (-1.7)	56.3 (-2.3)	60.0 (-2.3)	61.5 (-2.1)	62.5 (-1.9)
VideoChat-R1.5-7B	**51.7** (+2.2)	**54.1** (+0.3)	**58.8** (+0.1)	61.4 (-0.9)	63.4 (-0.3)	63.7 (-0.7)
Video-R2-7B	50.1 (+0.6)	53.4 (-0.4)	57.6 (-1.1)	61.2 (-1.1)	63.3 (-0.4)	64.9 (+0.5)

Table 8: VideoMME (2,700 questions) accuracy (%) vs. number of uniformly sampled frames. All reasoning models are fine-tuned from Qwen2.5-VL-7B-Instruct; parentheses give the difference to the base model. Identical protocol for all models: HF backend, max_pixels=602,112, greedy decoding, direct-answer prompt, max_new_tokens=16 (no chain-of-thought), seed 22. Best per column in bold; – = not run.

8b-Instruct and Intern3.5VL-8b, were give with black frame as input instead of the video frames. In Table **??**, we show the results, where each row shows the accuracy that can be obtained without actually looking at visual tokens. In Figure 3, we also show the performance comparison for a blind model that sees black frames, and compare it with the same model with four uniformly sampled frames. Ideally, such questions contribute to the noise and do not help in assessing the capabilities of video understanding models. Few videos in LVBench and VideoMME that were extremely long (i.e., $\geq$ 60 minutes) were also removed. From the remaining pool, the questions were sampled such that, primary category of every source is present in **Meta-V** which includes: Video-MME: 101 short / 109 medium / 102 long; all 12 task types, 6 domains, 30 sub-categories; LongVideoBench: all 17 question categories with 10 to 23 items each; duration groups 15 s / 60 s / 600 s / 3600 s with 38 / 49 / 72 / 81 items; 10 topic categories; LVBench: 6 question types, dominated by entity recognition (57) and event understanding (48), 6 genres; VideoMMMU: 60 per track, all 30 disciplines with 3 to 9 items, 6 QA types, all multiple choice; VSI-Bench: all 10 question types, 55 ScanNet / 42 ARKitScenes / 25 ScanNet++ scenes. The following is the spread of (#Videos,#QA pairs) in each split; Short - (382 videos, 382 QA pairs), Medium - (335 videos, 281 QA pairs), Long - (283 videos, 172 QA pairs) Two types of scores can be obtained for **Meta-V** and each of them is specified in the tables: (a) weighted mean of short, medium, and long videos, where the score is weighted by the number of samples in each bucket, (b) non-weighted mean of short, medium, and long videos, which is the direct mean of the three scores.

A.6 Base models vs Reasoning models

Recently, a lot of effort in video understanding community has shifted towards building better video reasoning models. Video-R1 Feng et al. (2025) introduces a T-GRPO algorithm, specifically designed to handle temporal information in videos. It utilizes two datasets: Video-R1-CoT-165k for supervised finetuning (SFT) and Video-R1-260k for reinforcement learning (RL) training. Video-RFT Wang et al. (2025a) proposes a multi-expert driven, cognition-inspired CoT curation pipeline. In this framework, an LLM first generates preliminary CoTs based on rich, structured, and literal representations of video content. A VLM then refines these CoTs by conditioning them on the actual video input. This process results in two datasets: VideoRFT-CoT-102K for SFT and VideoRFT-RL-

Duration	Model	#Frames					
		4	8	16	32	64	128
Short	Qwen2.5-VL-7B	56.9	62.2	68.2	73.9	74.7	74.8
	Video-R1-7B	55.6 (-1.3)	59.3 (-2.9)	66.0 (-2.2)	70.7 (-3.2)	73.0 (-1.7)	73.9 (-0.9)
	VideoChat-R1.5-7B	57.8 (+0.9)	62.8 (+0.6)	68.0 (-0.2)	72.1 (-1.8)	74.3 (-0.3)	74.0 (-0.8)
Medium	Qwen2.5-VL-7B	47.9	52.1	58.3	60.4	62.8	63.7
	Video-R1-7B	46.2 (-1.7)	51.7 (-0.4)	55.7 (-2.7)	58.3 (-2.1)	61.1 (-1.7)	61.2 (-2.4)
	VideoChat-R1.5-7B	50.4 (+2.6)	52.6 (+0.4)	57.9 (-0.4)	60.8 (+0.3)	63.2 (+0.4)	64.1 (+0.4)
Long	Qwen2.5-VL-7B	43.8	47.0	49.4	52.7	53.6	54.7
	Video-R1-7B	43.8 (+0.0)	45.1 (-1.9)	47.3 (-2.1)	51.1 (-1.6)	50.4 (-3.1)	52.4 (-2.2)
	VideoChat-R1.5-7B	46.9 (+3.1)	47.0 (+0.0)	50.4 (+1.0)	51.2 (-1.4)	52.7 (-0.9)	53.0 (-1.7)

Table 9: VideoMME accuracy (%) by video duration (short <2 min, medium 4–15 min, long 30–60 min; 900 questions each) vs. number of frames. Parentheses: difference to Qwen2.5-VL-7B-Instruct. Protocol as in Table 8.

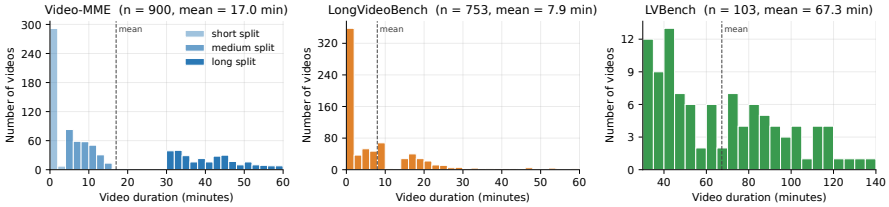


Figure 4: Video duration of VideoMME-long (Fu et al., 2025), LongVideoBench (Wu et al., 2024), LVBench (Wang et al., 2025b).

310K for RL training. In contrast, Video-RTS Wang et al. (2025e) presents a different approach by combining efficient RL with a video-adaptive test-time scaling (TTS) strategy. All of these works use Qwen2.5-VL-7b as the base model and continue training on this.

In Table 8, we show the performance of these video reasoning models on VideoMME Fu et al. (2025) dataset. It can be seen that the reasoning models provide only marginal gains over their base model, raising the concerns of necessity of such models.

A.7 Video duration of long video benchmarks.

As mention in section 3.2, different benchmarks have a different definition of long videos. From Figure 4, it can be seen that the average duration of long split of VideoMME dataset is 17 minutes, where for LongVideoBench it is 7.9 minutes, and for LVBench it is 67.3 minutes. For **Meta-V**, we divide the videos into three splits based on the duration of the video as explained in section 3.2.

A.8 Evaluation datasets and settings

The following are the datasets are their splits used for evaluation.

VideoMME. We evaluate `videomme` without subtitles. This benchmark has 2700 QA pairs, and 900 videos. The default prompt of this dataset is:

```
Select the best answer to the following multiple-choice question based on
the video and the subtitles.
Respond with only the letter (A, B, C, or D)
of the correct option.
<question>
A. <option>
B. <option>
```

```
C. <option>
D. <option>

Answer with the option's letter from the given choices directly.
```

LongVideoBench. We evaluate `longvideobench_val_v`, which contains 1337 QA pairs,and 753 videos. The default prompt of this dataset is:

```
<question>
A. <option0>
B. <option1>
C. <option2>
D. <option3>
E. <option4>
Answer with the option's letter from the given choices directly.
```

TempCompass. We evaluate `tempcompass_multi_choice`, which contains 1580 QA pairs, and 410 videos. The default prompt of this dataset is:

```
<question>
A. <option>
B. <option>
C. <option>
Please directly give the best option:
```

MMVU. We evaluate `mmvu_val`, which contains 1000 QA pairs, and 583 videos. The default prompt of this dataset is:

```
Question:<question>
A: <a>
B: <b>
C: <c>
D: <d>
E: <e>
Visual Information: processed video
Do not generate any intermediate reasoning process.
Answer directly with the option letter from the given choices.
```

For Open-ended:

```
Question:<question>
Visual Information: processed video
Do not generate any intermediate reasoning process.
Directly output the final answer.
```

LVBench. We evaluate `lvbench`, which contains 1549 QA pairs, and 103 videos. The default prompt of this dataset is:

```
<question>
(A) <option>
(B) <option>
(C) <option>
(D) <option>
Answer the question with the option letter
```

VSIBench. We evaluate `vsibench`, which contains 5130 QA pairs on 288 videos. The default prompt of this dataset is:

```
These are frames of a video.
<question>
```

```
Options:
<option A line>
<option B line>
...
Answer with the option's letter from the given choices directly.
```

For numerical question types:

```
These are frames of a video.
<question>
Please answer the question using a single word or phrase.
```

MLVU. We evaluate `mlvu_dev`, which contains 2174 QA pairs on 1222 videos. The default prompt of this dataset is:

```
<question>
(A) <option>
(B) <option>
(C) <option>
(D) <option>

Only give the best option.
Best option:
```

A.9 Evaluation cards

We provide evaluation cards as a template which can be provided with every evaluation table for better reproducibility.

Evaluation Card (Table 4)

SCOPE	Models	InternVL3.5-4B, Qwen3-VL-4B, Qwen2.5-VL-7B, InternVL3.5-8B, Qwen3-VL-8B
	Datasets	Video-MME, LongVideoBench, TempCompass, MMVU (val), LVBench, VSI-Bench, MLVU, Meta-V (short / medium / long)
RUNTIME	Inference backend	HF
	Batch size	*per no. of frames:* 4 → 16, 8 → 8, 16 → 4, 32 → 4, 64 → 2, 128 → 1
INPUT	Input modality	.mp4
	No. of frames	{4, 8, 16, 32, 64, 128}
	Frame sampling	**uniform**
	Max frames	**none**
	Cap mode	**none**
DECODING	Video decoder	torchcodec
	Frame resolution	602112 *(max pixel count)*
	Temperature	0.0
	Prompt layout	**vt**
	Prompt(s)	Dataset default prompts *(see Comments)*
COMMENTS	Protocol	One fixed protocol for all models; only the frame budget varies, all other settings are held fixed.
	Reported rows	Shaded "rep." rows are copied from the original papers, not re-run; – = not reported or not evaluated.
	Aggregates	Avg. = mean over the seven benchmarks (excl. Meta-V); Meta-V Avg. = unweighted mean of short, medium and long.
	Default prompts	*One line per dataset;* $\langle Q \rangle$ *= question,* $\langle O \rangle$ *= answer options.*
	Video-MME	`Select the best answer to the following multiple-choice question based on the video and the subtitles. Respond with only the letter (A, B, C, or D) of the correct option.` $\langle Q \rangle$ $\langle O \rangle$ `Answer with the option's letter from the given choices directly.`
	LongVideoBench	$\langle Q \rangle$ $\langle O \rangle$ `Answer with the option's letter from the given choices directly.`
	TempCompass	$\langle Q \rangle$ $\langle O \rangle$ `Please directly give the best option:`
	MMVU	*MC:* `Question:`$\langle Q \rangle$ $\langle O \rangle$ `Visual Information: processed video Do not generate any intermediate reasoning process. Answer directly with the option letter from the given choices.` *open-ended:* `Question:`$\langle Q \rangle$ `Visual Information: processed video Do not generate any intermediate reasoning process. Directly output the final answer.`
	LVBench	$\langle Q \rangle$ $\langle O \rangle$ `Answer the question with the option letter`
	VSI-Bench	*MC:* `These are frames of a video.` $\langle Q \rangle$ `Options:` $\langle O \rangle$ `Answer with the option's letter from the given choices directly.` *numerical:* `These are frames of a video.` $\langle Q \rangle$ `Please answer the question using a single word or phrase.`
	MLVU	$\langle Q \rangle$ $\langle O \rangle$ `Only give the best option. Best option:`

Table 10: Evaluation card for Table 4. Options in **bold** are the ones used; a set $\{\cdot\}$ denotes a parameter sweep (one run per value).

Evaluation Card (Table 5)

SCOPE	Models	Qwen2.5-VL-7B
	Datasets	Meta-V *(short / medium / long)*
RUNTIME	Inference backend	HF
	Batch size	1 *(input-type sweep)*; 16 *(layout sweep)*
INPUT	Input modality	{.mp4, .jpg} *(layout sweep: .jpg only)*
	No. of frames	4
	Frame sampling	**uniform** | *fps rate*
	Max frames	**none** | *int*
	Cap mode	**none** | *uniform* | *cutoff*
DECODING	Video decoder	torchcodec *(.mp4 input; n/a for .jpg)*
	Frame resolution	602112 *(max pixel count)*
	Temperature	0.0
	Max new tokens	default*; 1024 for $P1$*
	Prompt layout	{vt, tvt, vtv} *(input-type sweep: vt only)*
	Prompt(s)	{$P0$, $P1$, P^*} *(see Comments)*
COMMENTS	Protocol	Standardized configuration with two sweeps, each over all three prompts: (i) input file type (.mp4 vs. .jpg) at layout vt, and (ii) prompt layout (vt, tvt, vtv) at .jpg input. The .jpg / vt setting is shared by both. All other settings are held fixed.
	Layouts	Order of video (v) and text (t) in the prompt: vt = video then text; tvt = text before and after the video; vtv = video before and after the text.
	$P0$ (default)	Default per-dataset prompt from `lmms_eval`.
	$P1$ (CoT)	`Think step by step about what happens in the video, then end your reply with Answer:` *(max. 1024 new tokens)*
	P^* (ours)	Replaces the default prompt with `Respond with a single letter and nothing else.`
	Aggregates	Avg. = weighted mean of the short, medium and long splits; subscripts give the drop w.r.t. P^* in the same setting (input file type or layout).

Table 11: Evaluation card for Table 5. Options in **bold** are the ones used; a set {·} denotes a parameter sweep (one run per value).