

# Apollo: An Exploration of Video Understanding in Large Multimodal Models

Orr Zohar<sup>∞,†,ω</sup>, Xiaohan Wang<sup>†</sup>, Yann Dubois<sup>†</sup>, Nikhil Mehta<sup>∞</sup>, Tong Xiao<sup>∞</sup>, Philippe Hansen-Estruch<sup>∞</sup>, Licheng Yu<sup>∞</sup>, Xiaofang Wang<sup>∞</sup>, Felix Juefei-Xu<sup>∞</sup>, Ning Zhang<sup>∞</sup>, Serena Yeung-Levy<sup>†</sup>, Xide Xia<sup>∞</sup>

<sup>∞</sup>Meta GenAI, <sup>†</sup>Stanford University

<sup>ω</sup>Work done at Meta

Despite the rapid integration of video perception capabilities into Large Multimodal Models (LMMs), the underlying mechanisms driving their video understanding remain poorly understood. Consequently, many design decisions in this domain are made without proper justification or analysis. The high computational cost of training and evaluating such models, coupled with limited open research, hinders the development of video-LMMs. To address this, we present a comprehensive study that helps uncover what effectively drives video understanding in LMMs.

We begin by critically examining the primary contributors to the high computational requirements associated with video-LMM research and discover *Scaling Consistency*, wherein design and training decisions made on smaller models and datasets (up to a critical size) effectively transfer to larger models. Leveraging these insights, we explored many video-specific aspects of video-LMMs, including video sampling, architectures, data composition, training schedules, and more. For example, we demonstrated that fps sampling during training is vastly preferable to uniform frame sampling and which vision encoders are the best for video representation.

Guided by these findings, we introduce **Apollo**, a state-of-the-art family of LMMs that achieve superior performance across different model sizes. Our models can perceive hour-long videos efficiently, with Apollo-3B outperforming most existing 7B models with an impressive 55.1 on LongVideoBench. Apollo-7B is state-of-the-art compared to 7B LMMs with a 70.9 on MLVU, and 63.3 on Video-MME.

**Date:** December 16, 2024

**Correspondence:** Orr Zohar at [orrzohar@stanford.edu](mailto:orrzohar@stanford.edu), Xiaohan Wang at [xhanwang@stanford.edu](mailto:xhanwang@stanford.edu)

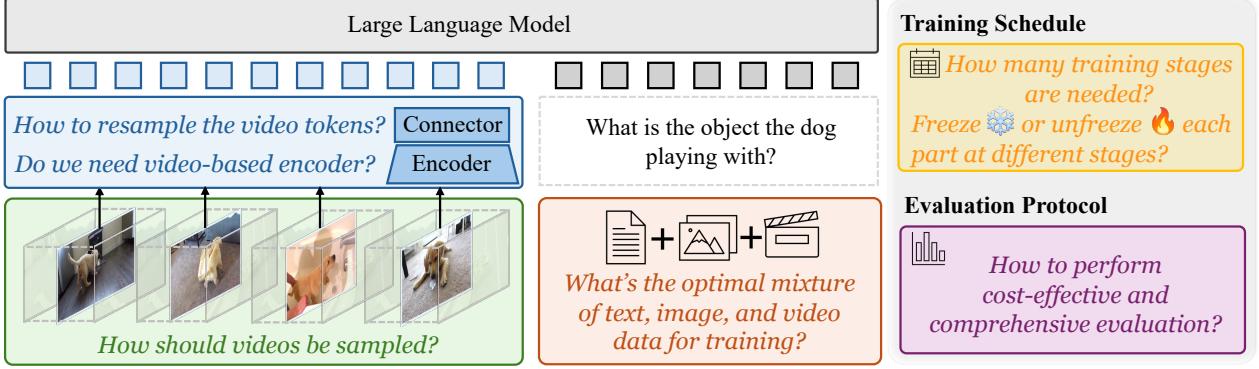
**Project Page:** <https://apollo-lmms.github.io>



## 1 Introduction

Despite the rapid advancements in language and image-language modeling (Hoffmann et al., 2022; Brown, 2020; Yang et al., 2024; Liu et al., 2024a; Alayrac et al., 2022; Laurençon et al., 2024a; OpenAI, 2024), the development of video Large Multimodal Models (video-LMMs) has not kept pace. Videos provide a rich, dynamic information source, capturing nuanced temporal and spatial features beyond the reach of static images. However, video-LMMs remain under-explored, hampered by unique challenges: notably higher computational demands and a broader, more complex design space compared to their image-based counterparts (Li et al., 2023a, 2025; Liu et al., 2024d; Li et al., 2024b; Xu et al., 2024a).

Many fundamental questions about video-LMM design remain unanswered: How should videos be sampled? Which vision encoders yield optimal representations? What are the best practices for resampling video tokens? Early approaches primarily extended image-LMMs directly (Xu et al., 2024b; Kim et al., 2024; Wu, 2024; Zhang et al., 2024e) or with video-specific fine-tuning (Li et al., 2023a; Zhang et al., 2023; Maaz et al., 2023). Recent methods introduced diverse design choices, such as longer context windows (Zhang et al., 2024e), multi-modality mixing (Li et al., 2024a,c), agent workflows (Wang et al., 2024c), self-training (Zohar et al., 2024), and more. Despite these efforts, the impact of these design decisions on video-LMM performance is poorly understood. This lack of systematic investigation motivates our study.



**Figure 1 Apollo exploration.** Schematic illustrating our comprehensive exploration of video-specific design choices; critically evaluating the existing conceptions in the field, from video sampling and model architecture to training schedules and data compositions. For example, we found that the SigLIP encoder is the best single encoder for video-LMMs but can be combined with additional encoders to improve temporal perception, and that keeping a  $\sim 10\%$  text data during fine-tuning is critical for video understanding performance. More insights can be found in Sec. 4 & 5.

To overcome the computational challenges of training video-LMMs, we explore whether design decisions from smaller models correlate effectively with larger ones. Traditional scaling laws (Hoffmann et al., 2022) predict model performance based on size, but apply to models trained from scratch and require training multiple models to predict performance. Scaling laws have also been observed in LMM pretraining (Aghajanyan et al., 2023; Yu et al., 2023). Since LMMs integrate multiple pre-trained components, it’s uncertain if these laws hold. By relaxing scaling laws, our experiments reveal that design choices made with smaller LMMs transfer to larger ones, a phenomenon we term **Scaling Consistency** (Sec. 3).

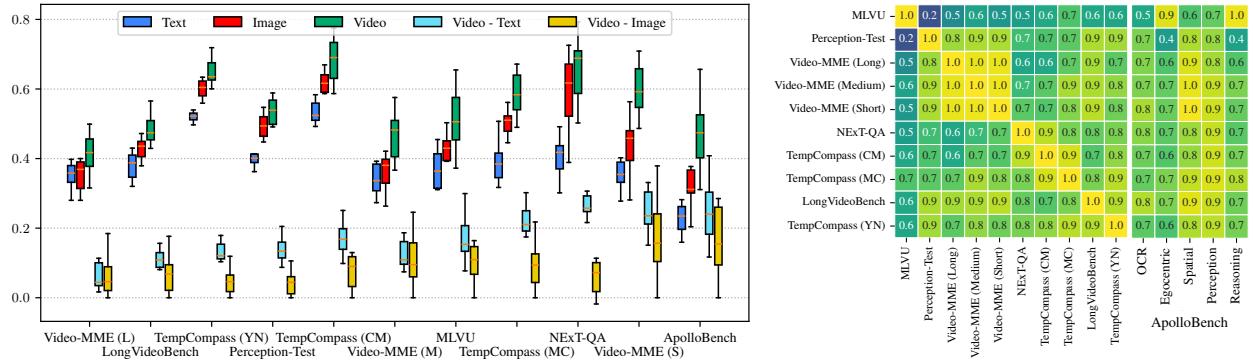
Utilizing these insights, we conduct an extensive study across the video-LMM design space, addressing essential aspects of video-language modeling, such as video sampling and encoding methods, token resampling and integration strategies, and data compositions (Sec. 4 & 5). For instance, we discover that frames-per-second video sampling significantly outperforms standard uniform sampling used in previous works (Liu et al., 2024d,b). We also find which vision encoder combinations are the most robust and that the Perceiver Resampler (Jaegle et al., 2021) outperforms average pooling. When studying the numerous benchmarks available, we discovered a large portion of the performance improvements are driven primarily via language modeling and, therefore, curate ApolloBench, which significantly reduces evaluation time while improving assessment quality (Sec. 2).

Building upon our findings, we introduce Apollo, a family of state-of-the-art LMMs capable of comprehending hour-long videos. Apollo models exhibit strong performance across various scales. Notably, Apollo-3B surpasses most existing 7B models, achieving scores of 58.4 (+12.8) on Video-MME (w/o sub.), 68.7 (+6.9) on MLVU, and 62.7 (+14.1) on ApolloBench. Apollo-7B attains impressive scores of 61.2 (+0.6) on Video-MME (w/o sub.), 70.9 (+5.4) on MLVU, and 66.3 (+2.4) on ApolloBench, making it competitive with 30B models.

#### Our contributions are as follows:

1. We conduct a systematic exploration of the video modeling design space for Large Multimodal Models, uncovering critical factors that drive performance and providing actionable insights for future research.
2. We identify Scaling Consistency, where design decisions effective for smaller LMMs and datasets are transferred effectively to larger ones, reducing computational costs and enabling efficient experimentation.
3. We address evaluation inefficiencies by curating ApolloBench, a subset of existing benchmarks that cuts evaluation time by  $41\times$  while offering detailed insights into temporal reasoning and perception tasks.
4. We introduce Apollo, a family of LMMs that achieves state-of-the-art results across video understanding multiple benchmarks. Notably, Apollo-3B surpasses nearly all 7B models, while Apollo-7B variant is state-of-the-art among models with less than 30B parameters.

In Sec 2, we analyze the state of video benchmarks and introduce ApolloBench. In Sec. 3, we show how one can relax traditional scaling laws for computational savings. In Sec. 4, we explore the architecture design space. In Sec. 5, we investigate different training protocols and data mixtures. Finally, in Sec. 6, we present Apollo, a state-of-the-art family of video-LMMs.



**Figure 2 Benchmark Analysis.** (Left) Accuracy of the open-source LMMs on various video question-answering benchmarks when provided with different input modalities: full video (green bars), a single frame from the video (red bars), and text-only input without any visual content (blue bars). The light blue shaded areas represent the difference in accuracy between video and text inputs, highlighting the extent to which video perception enhances performance over text comprehension alone. The yellow shaded areas indicate the difference between video and image inputs, quantifying the additional benefit of temporal information from videos compared to static images. (Right) The correlation matrix shows the redundancy among benchmarks by illustrating the correlation coefficients between model performances on different benchmarks. Each cell in the matrix represents how closely the two benchmarks are related in terms of model performance. Our proposed benchmark, ApolloBench, is highly correlated with all tested benchmarks, suggesting that it offers an equally effective evaluation while being more computationally efficient.

## 2 How effective are existing video question-answering benchmarks?

The rapid advancement of video Large Multimodal Models (video-LMMs) has spurred the creation of numerous video question-answering benchmarks, including Video-MME, MLVU, LongVideoBench, and others (Fu et al., 2024; Wu et al., 2024; Zhou et al., 2024; Patraucean et al., 2023; Li et al., 2024d; Wang et al., 2024b; Cai et al., 2024). While this proliferation enables comprehensive evaluation, it also introduces significant resource intensiveness and redundancy. For example, evaluating a 3B-parameter model on these benchmarks requires 184 A100 GPU hours. In this section, we first analyze the quality of existing benchmarks (Sec. 2.1), their redundancy (Sec. 2.2), and introduce ApolloBench (Sec. 2.3) by building on these insights.

### 2.1 Evaluating benchmark quality

What drives video benchmark performance is not known. As shown by Goyal et al. (2017), some image question-answering benchmarks are largely driven by text comprehension rather than image perception. Chen et al. (2024a) further showed that data leakage in either the LLM or LMM training stage may be further contaminating evaluation in image question-answering benchmarks. To evaluate the state of video question answering benchmarks, we evaluated ten open-source LMMs on several benchmarks: Video-MME (Fu et al., 2024), TempCompass (Liu et al., 2024c), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2024), NExTQA (Xiao et al., 2021), and PerceptionTest (Patraucean et al., 2023)—under three different settings:

- **Video:** Models prompted with video input using their standard video sampling. Green in Fig. 2, left.
- **Image:** Models are provided only the center frame of each video. Red in Fig. 2, left.
- **Text:** Models are prompted with only the original question, without any visual input. Blue in Fig. 2, left.

As illustrated in Fig. 2, left, a significant portion of existing benchmarks are answered solely through text comprehension alone (blue boxplots) or only using the center frame (red boxplots), indicating that LMMs do not rely on video perception in a large portion of existing benchmarks. We sorted the benchmarks by the difference between the Video and Text performance (light blue). A benchmark relies more and more on its video perception capabilities when this bar is high. When examining Fig. 2, left, it is apparent that as videos get longer, the reliance on video perception decreases (compare Video-MME S/M/L). To evaluate how much of the benchmarks require video input to answer the question, we also plot the difference between the Video

and Image performance (yellow). Some benchmarks can almost be entirely solved using a single frame. For example, in line with [Buch et al. \(2022\)](#), we find that NExTQA is solved using a single frame. Perception-test also behaves similarly. Finally, when studying Fig. 2, left, a high variance in the box plot is desired as this indicates more highly discriminative benchmarks. Among all the existing benchmarks, Video-MME (Short), MLVU, and TempCompass emerge as the top performers.

## 2.2 Redundancy in existing benchmarks

To evaluate the redundancy in video question answering benchmarks, we evaluated ten open-source LMMs on several benchmarks: Video-MME ([Fu et al., 2024](#)), TempCompass([Liu et al., 2024c](#)), LongVideoBench ([Wu et al., 2024](#)), MLVU ([Zhou et al., 2024](#)), NExTQA ([Xiao et al., 2021](#)), and PerceptionTest ([Patraucean et al., 2023](#)). We then calculated the correlation of each of the benchmarks to each other, the result of which can be seen in Fig. 2, right. Our analysis revealed significant redundancy among benchmarks, as evidenced by the block-diagonal correlation matrix, where we can identify groups of benchmarks that are highly correlated.

To evaluate the effect of different question types and video durations, we also evaluated the correlations between video duration groups. We find that the performance of models on short and long videos within Video-MME ([Fu et al., 2024](#)) exhibits an  $R^2 = 0.94$ , see App. Fig. 13, while in LongVideoBench,  $R^2 > 0.92$  between all duration groups. To assess the effect of question format, we studied the TempCompass ([Liu et al., 2024c](#)) dataset, which has different question formats (multiple-choice, yes/no, caption matching, and caption generation), and found that they are also highly correlated ( $R^2 > 0.8$ ), indicating that varying question types do not significantly diversify the evaluation (see App. Fig. 12).

## 2.3 Introducing ApolloBench

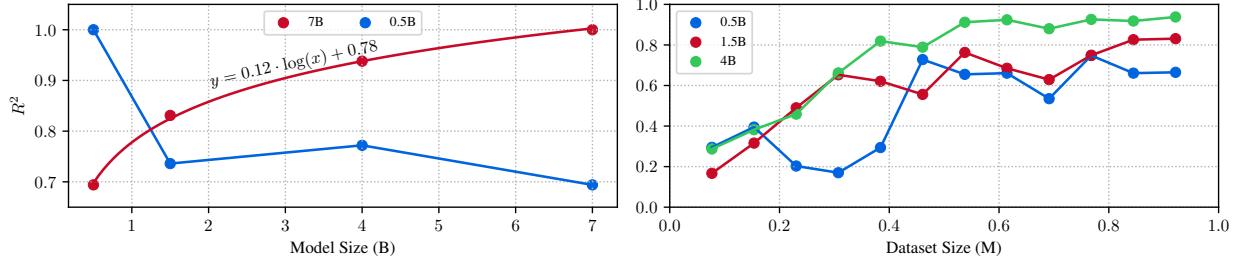
Motivated by these insights, we set out to curate a more effective and efficient benchmark suite called ApolloBench. We focused on multiple-choice questions to eliminate the need for external tools like ChatGPT, ensuring a consistent and cost-effective evaluation process ([Wu, 2024](#)).

We filtered out questions that could be correctly answered by more than 50% of the models with either text or image inputs, removing questions that do not require video perception (see Fig. 2, left, ApolloBench). Subsequently, we identified five broad temporal perception categories: Temporal OCR, Egocentric, Spatial, Perception, and Reasoning. Questions were then manually categorized into each one of these categories. We selected the top 400 questions from these categories that exhibited the most discrimination between models via entropy and manually verified each one to validate the correctness of the selected questions. Evaluating on ApolloBench is  $41\times$  faster while being highly correlated with existing benchmarks (see Fig. 2, right) and more influenced by video perception (Fig. 2, left). For more details, see App. Sec. B and App. Fig. 11.

## 3 Scaling Consistency: How small can you go during model design?

Developing Large Multimodal Models (LMMs) poses significant computational challenges, especially when training on extensive datasets with billion-parameter models. To make the research process more efficient, it is essential to determine whether smaller LMMs and datasets can reliably inform design decisions for larger ones. Traditional scaling laws require training multiple models of varying sizes for each design decision to derive how performance scales with model size. However, in the context of LMMs, which typically utilize multiple pre-trained components (e.g., vision encoders, language models), scaling each component individually is impractical due to the lack of availability of such components and the immense computational resources required. As such, we set to relax these scaling laws and instead reason about correlation or transfer of design decisions between models of different sizes.

This section investigates the correlation between design decisions made on LMMs of different sizes. Specifically, we selected 21 model variations encompassing various design aspects such as architecture, video sampling methods, training strategies, and data mixtures. Each variation was trained using four different Large Language Models (LLMs): Qwen2-0.5B, Qwen2-1.5B, Qwen1.5-4B, and Qwen2-7B ([Bai et al., 2023; Yang et al., 2024](#)), resulting in a total of 84 models. We then analyzed the correlation ( $R^2$ ) between the performance of these models (see App. Fig. 15). Our findings reveal that design decisions on models of a critical size



**Figure 3 Scaling Consistency.** We discover Scaling Consistency, where design decisions made with smaller models on smaller datasets carry over to larger models on larger datasets. (**Left**)  $R^2$  values of **7B** and **0.5B** versus other LLM sizes show an increasing correlation with larger LLM sizes for the 7B model. The same trend is not seen in the 0.5B model. Interestingly, while the Qwen1.5-4B model variants have lower/similar performance to their smaller Qwen2 – 1.5B counterparts, the correlation to larger models is still higher (See App. Fig. 15). (**Right**)  $R^2$  of 0.5/1.5/4B models to 7B vs dataset size.  $R^2$  to larger datasets starts to plateau at around 500K samples.

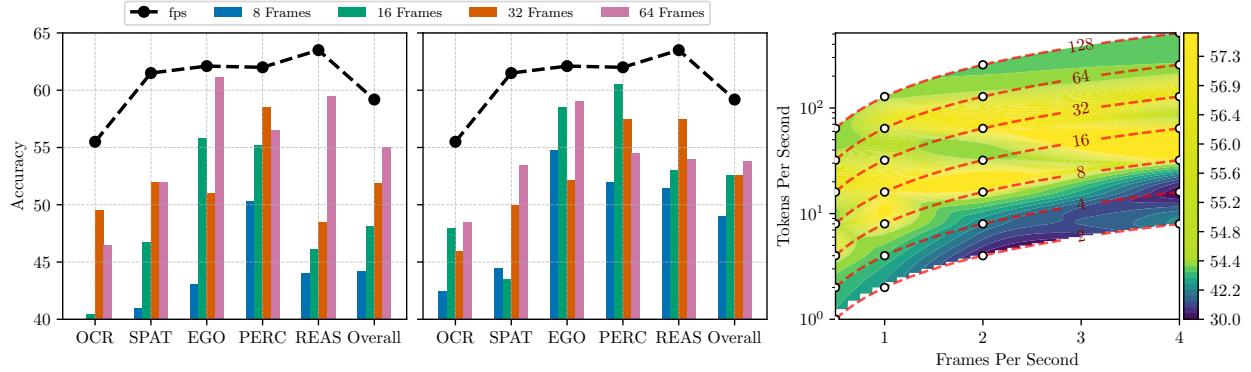
( $\sim 2 - 4$ B) correlate highly ( $R^2 > 0.9$ ) with those on larger models, a phenomenon we term Scaling Consistency (see Fig. 3). For instance, the  $R^2$  between the 4B and 7B models is 0.938, indicating a strong predictive relationship. Please refer to the App. Sec. D for a detailed analysis.

Scaling laws typically require training models of various sizes to study performance trends. However, due to limited availability and high computational cost, scaling laws are rarely applied to LMMs. In contrast, Scaling Consistency demonstrates that design decisions made on moderately sized models ( $\sim 2 - 4$ B) and datasets transfer reliably to larger models, even across different model families. This allows researchers to make informed design choices without extensive scaling studies. Our primary goal is to show that design decisions transfer reliably, reducing computational burden and accelerating research.

*Large Language Model size.* In Fig. 3, left, we plot the  $R^2$  values between models of various sizes and the 7B LLM model variant. The correlation with the 7B LLM increases approximately log-linearly with the size of the smaller LLMs and generalizes between model families. This behavior is not observed with smaller models, e.g., 0.5B, where  $R^2$  immediately drops below 0.8, and no log-linear behavior can be observed. This reinforces the existence of a critical model size ( $\sim 2 - 4$ B) where design decisions transfer reliably—a phenomenon we term Scaling Consistency. Scaling Consistency seems to generalize between model families, as a mix of Qwen1.5 and Qwen2 models were utilized in this study. For example, while the Qwen2-1.5B and Qwen1.5-4B model variants had similar performance, the 4B Qwen1.5-4B was still more correlated than the 1.5B model. Please refer to the App. Sec. D for a comprehensive analysis.

*Impact of dataset size.* We examined the impact of dataset size on model performance by training models using the same data mixture but varying the dataset size from 75K to 1M samples. The results are shown in Fig. 3, right, where the correlation of the 0.5/1.5/4B models trained on varying datasets sized to 7B trained on the full dataset can be seen as a function of dataset size. Focusing on the 4B LLM variant, we observed that the correlation ( $R^2$ ) with larger models plateaus around  $\sim 500$ K samples, indicating that increasing the dataset size beyond this point yields diminishing returns in terms of informing design decisions. In contrast, smaller models (e.g., 0.5B and 1.5B) exhibited less consistent behavior, with their  $R^2$  values fluctuating more across different dataset sizes. This suggests that a dataset size of approximately 500K samples is sufficient for moderately sized models (2–4 billion parameters) to reliably transfer design insights to larger models.

**Finding 1:** We discover Scaling Consistency, where design decisions can be made on smaller models and datasets and transfer reliably to larger ones.



**Figure 4 Video sampling.** We compare different sampling strategies and their effect on performance. **(Left)** Models were trained and tested using uniform sampling. Increasing the number of frames improves overall performance but does not reach fps sampling performance. **(Middle)** Models trained with uniform sampling but tested with fps sampling. Differences in performance are not explained by the number of frames sampled at test time. **(Right)** Analysis of the effect of frames per second (fps) and tokens per second (tps) on overall performance. The dotted red lines (--) indicate the tokens per frame. For a per-metric breakdown, please see App. Fig. 9.

## 4 Exploring the video-LMM design space: what influences effective model design?

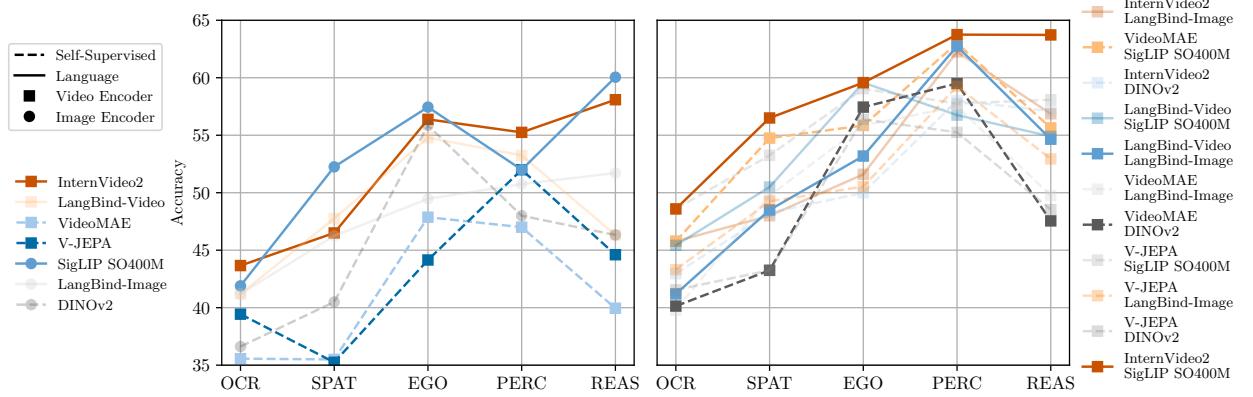
In this section, we analyze key architectural design choices shaping the performance of Large Multimodal Models (LMMs) in video-language tasks. We focus on four critical aspects: **(I) Video sampling** (Sec. 4.1) where we compare uniform and fps video sampling and evaluate the effect tokens and frames per second have on downstream performance. **(II) Video representation** (Sec. 4.2) where we explore how image and video encoders impact video representation and show which encoder and encoder pairs lead to the best performance. **(III) Video token resampling** (Sec. 4.3) where we test different visual token resamplers. **(IV) Video token integration** (Sec. 4.4) where we examine various strategies to integrate the visual token into the text tokens.

Using Scaling Consistency, we opted to perform the following exploration using Qwen2.5 3B (Yang et al., 2024) and trained on a dataset of 750K samples. As demonstrated in Sec. 3, these findings exhibit a strong correlation ( $R^2 > 0.9$ ) with results on larger models and across different model families. Unless stated otherwise, a Perceiver Resampler (Jaegle et al., 2021) was employed, with 16 tokens per frame at a frame rate of 2 fps. The dual encoders used were InternVideo2 (Wang et al., 2024d) and SigLIP-SO400M (Zhai et al., 2023). When training on images, images were duplicated before being encoded by the video encoders for fully integrated encoding, as we found it to be slightly more performant with fewer parameters and complexity (see App. Sec. C.2). This is in line with Lin et al. (2023).

### 4.1 Video sampling

Videos can be sampled in many ways, from uniform sampling - uniformly sampling  $N$  frames from the video (Li et al., 2023a; Lin et al., 2023; Jin et al., 2024; Zhang et al., 2024f), to fps sampling - sampling at a set number of frames per second. While many recent methods have preferred fps sampling (Liu et al., 2024d; Li et al., 2024a), they default to uniform sampling when video durations exceed their frame sampling capacity (usually  $\sim 64$ ). The maximum frame capacity is typically constrained due to the memory requirement at the vision encoder and or the LLMs context window.

Uniform frame sampling enables simplified training because the effective ‘vision batch size’ (i.e., the number of frames that need to be encoded) remains constant. However, training video-LMMs with uniform frame sampling means that the time difference between concurrent frames changes with each video, effectively setting a different ‘video speed’ in every iteration. As a result, when uniformly sampling  $N$  frames from videos of varying lengths, the effective playback speed represented in the sampled frames changes. For a shorter video,  $N$  uniformly sampled frames represent a slower playback (more frames per second of actual content), while for a longer video, those same  $N$  frames represent a faster playback. This will likely hamper the LMMs’ capability



**Figure 5 Vision encoders.** In our study, we tested InternVideo2 (Wang et al., 2024d), LanguageBind-Image/Video (Zhu et al., 2023a), V-JEPA (Bardes et al., 2023), Video-MAE (Tong et al., 2022), SigLIP-SO400M (Zhai et al., 2023), and DINOv2 (Oquab et al., 2023), and their combinations. (**Left**) SigLIP-SO-400M emerges as the best overall among single encoders. We also find that image encoders underperform in temporal perception compared to video encoders. (**Right**) Performance of dual-encoder configurations. Language-supervised encoders outperformed their self-supervised counterparts. Combining InternVideo2 and SigLIP-SO-400M leads to the best overall performance.

to learn about the speed of objects in videos. Meanwhile, methods employing fps sampling must either limit the maximum video duration or the maximum number of frames (above which, they default to uniform frame sampling) in training or suffer from similar issues as in uniform sampling. An alternate approach is to sample ‘video clips’ of  $N$  frames at a set fps (or duration) and, when reaching the maximum token count, space these out instead. Here, rather than uniformly spacing out the sampled video frames, the  $N$  frames encoded by the video encoder maintain the same effective fps, and only frames of concurrent ‘clips’ are spaced out. Methods that utilize video encoders (Li et al., 2023a; Lin et al., 2023), where multiple frames are encoded together, should use such frame sampling as video encoders are typically trained at a constant fps (Bardes et al., 2023; Wang et al., 2024d; Zhu et al., 2023a; Tong et al., 2022).

To evaluate the effect of fps vs. uniform sampling, we trained four models that, while training, we uniformly sampled 8, 16, 32, or 64 frames. To test whether performance differences are due to the different frame sampling at test or train time, we evaluated these models with uniform and fps sampling. The results of this experiment can be seen in Fig. 4, left and middle. We found that uniform frame sampling consistently underperformed compared to fps sampling, Fig. 4, left. As can be seen, this performance gap is not due to the different number of frames sampled at test time, Fig. 4, middle. Therefore, we conclude that the uniform frame sampling of videos causes this performance gap during training.

**Finding 2:** fps sampling is preferable over uniform sampling during model training and inference.

When training at a constant fps, the tokens per second (tps) can also be varied using the token resampler. We investigate how varying fps and tps affect the LMM’s ability to comprehend videos. As can be seen in Fig. 4, right, there appears to be a tradeoff between tps and fps, balancing short and long video performance, with 8–32 tokens per frame achieving strong performance at different fps. Surprisingly, as can be seen in the App. Fig. 10, we found little dependence on fps, with both tokens per frame (tpf) and tps being more determinate. In concurrent work, Du et al. (2024) reached similar conclusions but required more tokens per frame ( $\sim 49$ ) to achieve performance saturation, likely as they utilized only image encoder and average pooling, which is less compressible. They also utilized uniform frame sampling, which may also affect this comparison.

**Finding 3:** There is a trade-off between tps and fps, with 8–32 tokens per frame being optimal.

Some methods employ active frame selection strategies, using the initial query to guide frame sampling (Wang et al., 2024c; Atallah et al., 2024; Wang et al., 2024e), and were not included in this study. Note that these would require frame resampling at every conversational turn.

## 4.2 Video representation

Training effective video encoders is challenging due to the high memory requirements for processing large video datasets and the comparatively low quality of available supervision. While early approaches predominantly used dedicated video encoders (Li et al., 2023a,b; Lin et al., 2023), recent developments favor image encoders instead (Liu et al., 2024b; Li et al., 2024a; Liu et al., 2024d). This shift arises because image encoders, although lacking temporal integration, still produce higher-quality representations that the LLM can readily leverage. Another possibility is that in this approach, image and video datasets can be fully integrated, possibly benefiting from image-video transfer and allowing the utilization of the much larger, more diverse, and more efficient image instruction tuning datasets (Li et al., 2024a; Zhang et al., 2024e).

Multiple studies have conducted extensive investigations into visual representation within image-LMMs (Shi et al., 2024; Tong et al., 2024). Laurençon et al. (2024b) found that SigLIP outperformed even much larger encoders, such as EVA-CLIP-5B. Zhan et al. (2024) showed that input image resolution influences performance more than token count, which may have influenced Laurençon et al. (2024b)'s ablation. Wang et al. (2023) compared encoders trained with supervision and found where each is preferable. However, whether image or video encoders are preferable for video-LMMs and what influences their performance is unclear. As such, we set out to find what drives good video representation in LMMs. We trained LMMs with several image and video encoders and their combinations and evaluated how this design decision impacted the final model performance. Our study includes diverse language- and self-supervised video/image encoders:

- **InternVideo2** (Wang et al., 2024d): trained in two stages: (1) unmasked video token reconstruction, (2) crossmodal contrastive learning aligning video with audio, speech, and text. Encodes four frames.
- **LanguageBind-Video v1.5** (Zhu et al., 2023a): initialized with an OpenCLIP model, trained contrastively with a frozen text encoder. Encodes eight frames.
- **VideoMAE** (Tong et al., 2022): trained through self-supervised learning by masking video patches with a reconstruction loss. Encodes sixteen frames.
- **V-JEPA** (Bardes et al., 2023): trained through self-supervised learning by predicting masked spatio-temporal regions in a learned latent representation space. Encodes sixteen frames.
- **SigLIP-SO400M** (Zhai et al., 2023): a shape-optimized model trained using a sigmoid loss function for language-image pre-training.
- **LanguageBind-Image** (Zhu et al., 2023a): one of the OpenCLIP image encoders and is not further tuned.
- **DINOv2** (Oquab et al., 2023): trained using a self-supervised teacher-student framework where the teacher guides the student to produce consistent representations across different image views.

As seen in Fig. 5, left, language-supervised encoders consistently outperform self-supervised encoders, in line with observations in prior work (Shi et al., 2024). In the single-encoder setups, SigLIP-SO400M had the best performance compared to all image/video encoders, demonstrating that video encoders must be improved to replace image encoders. Video encoders outperform image encoders only on Temporal Perception, indicating that LLMs struggle with fine-grained temporal integration (e.g., estimating speed and direction of movement).

**Finding 4:** SigLIP-SO400M is the best single encoder for video-LMMs.

We hypothesize that combining video and image encoders could offset their limitations, where image encoders do not encode temporal information, while video encoders have weaker spatial representations. Following Shi et al. (2024), embeddings generated by each encoder were interpolated and concatenated along the channel dimension before resampling. Combining encoders consistently outperforms their single-encoder counterparts, where InternVideo2+SigLIP-SO400M was the best overall, exhibiting a  $\sim 7\%$  improvement in ApolloBench. We found that video encoders with fewer input frames perform more favorably, possibly due to better image-video transfer. This design is in line with Wang et al. (2024a), whose vision encoder encodes videos two input frames at a time.

**Finding 5:** Combining SigLIP-SO400M with InternVideo2 leads to the best overall performance.

### 4.3 Video token resampling

Vision encoders output vision embeddings in a lower dimensionality than LLMs' hidden dimension, requiring a  $2 - 4 \times$  up-projection. Early methods often up-projected all visual tokens directly into the LLM's space (Lin et al., 2023; Li et al., 2023a). This approach leads to informational waste by instilling a synthetic information bottleneck. Laurençon et al. (2024b) demonstrated that resampling image tokens (where multiple up-projected tokens are pooled into one) does not reduce performance in image-LMMs. Token resampling is even more critical in video-LMMs as this directly affects how many frames can be processed, limiting the maximum video length. Video token resampling can be text-guided (e.g., using a Q-Former) (Li et al., 2025, 2023b; Zhang et al., 2023). However, this approach does not generalize well to multi-turn conversations, as tokens will be down-sampled according to the first question. Many others do some form of average pooling (Jin et al., 2024; Lan et al., 2024; Xu et al., 2024b).

Shi et al. (2024) tested multiple encoder integration approaches and found that channel-wise concatenation was preferable in nearly all configurations. Therefore, we adopted channel-wise concatenation in our experiments. We tested three token resampling methods: mlp up-projection + average pooling, 2D conv + average pooling, and perceiver resampling. As shown in Tab. 1, the Perceiver Resampler outperforms the other methods across all metrics. While Laurençon et al. (2024a) reported that utilizing the Perceiver Resampler hurts OCR performance; this trend was not observed in videos with the limited available token count per frame. Another key difference is the initial channel-wise concatenation of encoder features before resampling. This alignment enables the Perceiver to integrate features from different encoders better as they are better spatially aligned.

**Finding 6:** Perceiver resampling shows superior performance when reducing the tokens/frame.

Some methods utilize active token pooling, where the initial question is used to guide the token pooling (Li et al., 2025, 2023b; Zhang et al., 2023), usually using a Q-Former, and were not included in this study. Note that these would require token resampling at every conversational turn.

### 4.4 Video token integration

Integrating video and text tokens is a pivotal design choice for video-LMMs, as it directly influences how effectively the model processes and interprets multimodal content. Initial works naively concatenated the text and video tokens (Jin et al., 2024; Li et al., 2023a; Lin et al., 2023). However, recent trends have begun to either use separation tokens (Liu et al., 2024b) or via text (where a prompt is inserted between frames, usually indicating either frame ID or timestamp) (Li et al., 2024a). This design choice was also systematically ablated by Zhao et al. (2024a). To identify the most robust integration strategy, we experimented with four different methods, which can be seen in Tab. 2. We evaluated four integration strategies: direct insertion, separation tokens, textual timestamps, and combining separation tokens with timestamps. As can be seen, we found that adding any text or learnable tokens between video tokens results in a  $2 - 3\%$  improvement across ApolloBench. As such, we use the clip timestamps as they do not require learning any new token embeddings.

**Finding 7:** Adding tokens (text, learned, etc.) between the video tokens derived from different frames or clips is sufficient for efficient token integration.

Connector	ApolloBench					
	OCR	Spatial	Egocentric	Perception	Reasoning	Overall
2-layer 2D Conv. + adaptive average pooling	43.0	50.5	44.5	44.0	42.0	44.7
2-layer MLP + adaptive average pooling	47.5	53.7	51.5	52.0	61.5	<u>53.2</u>
Perceiver Resampler	50.4	54.8	58.5	58.8	55.4	<b>55.5</b>

**Table 1 Video token resampling methods.** Performance of different token resampling techniques on video-LMM tasks. The Perceiver Resampler outperforms other methods across all metrics. Different encoder features are concatenated along the channel dimension, following Shi et al. (2024); Tong et al. (2024).

Format	ApolloBench						Overall
	OCR	Spatial	Egocentric	Perception	Reasoning		
<vid_token>	50.4	54.8	58.5	58.8	55.4	55.5	
<vid_start><vid_token><vid_end>	49.2	54.8	61.7	60.2	57.9	<b>56.7</b>	
clip from {MM:SS}-{MM:SS}:<vid_token>	50.0	54.0	61.7	60.8	57.9	<b>56.8</b>	
clip from {MM:SS}-{MM:SS}:<vid_start><vid_token><vid_end>	50.0	54.2	61.2	55.7	60.6	56.2	

**Table 2 Video token integration methods.** Performance of different strategies for integrating video tokens into the text sequence. Incorporating textual timestamps before each clip yields the best overall performance.

## 5 How should video-LMMs be trained?

This section explores different training schedules and protocols for video Large Multimodal Models (video-LMMs). We begin by testing different training schedules and comparing single to multi-stage training (Sec. 5.1). We then examine when video encoders should be trained and with what data (Sec. 5.2). Finally, we explore how data composition affects performance (Sec. 5.3).

### 5.1 Training schedules

We systematically evaluated the impact of different training schedules on model performance, comparing single-stage, two-stage, and three-stage training protocols. Some studies have suggested that a single-stage training protocol performs similarly to two-stage ones but is more computationally efficient (Karamcheti et al., 2024). However, Tong et al. (2024) demonstrated that two-stage training improved model performance. Since then, many methods have broken down training into more and more training stages. For example, Li et al. (2024a); Liu et al. (2024b) utilized four training stages.

Video-LMMs are typically trained on a mixture of text, image, multi-image, and video data; it is possible to break down training into even more training steps, each with different components unfrozen and trained on different data compositions. For example, many video-LMMs include an additional, final training stage on long videos as these datasets are expensive and relatively small (Li et al., 2025; Liu et al., 2024b). Others first train on exclusively image datasets before training on multiple modality mixtures (Li et al., 2024a). We tested seven possible training configurations to evaluate the effect of these different training strategies.

As shown in Tab. 3, we found that gradually training the model yields the best performance. Specifically, we found that training the model over three stages yields the best performance, closely followed by the two-stage training schedules. Please note that different stages have different data compositions; specifically, whenever the LLM is frozen, the other components are tuned only on video data, and when the LLM is tuned, a mixture of text, image, multi-image, and video (following Sec. 5.3) is used.

**Finding 8:** Progressively unfreezing the different components in different stages leads to superior model training dynamics.

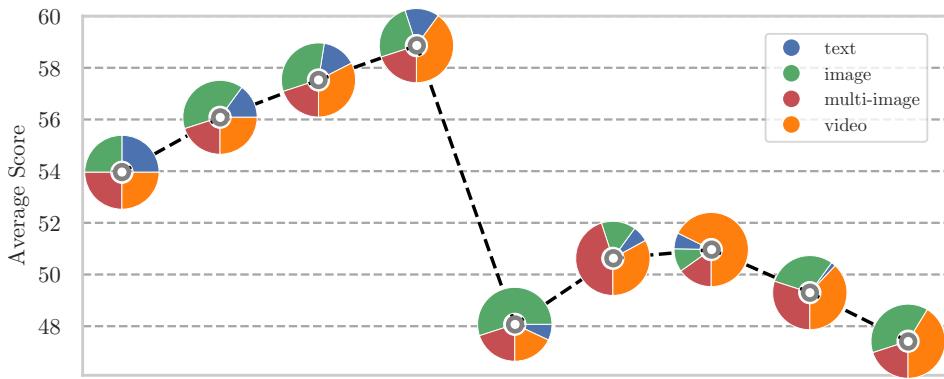
### 5.2 Training video encoders

It is unclear when and with what data one should train the video encoders. Tong et al. (2024) reported that training image encoders is beneficial in image-LMMs. However, video-LMMs are trained on a mixture of video, multi-image, and image data. Furthermore, to have a unified encoding scheme, images are replicated  $N$  times to be encoded by the video encoder. As such, these models have additional dimensions of the data mixture on which the encoders can be trained. We compared training vision encoders on either the data mixtures or exclusively on video data and whether first aligning the connector improves performance in Tab. 3.

In all experiments, if the LLM is frozen, the model is trained only with video data. When the LLM is unfrozen, we use a data mixture of text, image, multi-image, and video data as described in Sec. 5.3. As such, if both the video and LLM are unfrozen simultaneously, the vision encoders will be trained on a combination of image and video data. We found that this significantly hurts LMM performance. Training the encoders improves

	Training Stages			ApolloBench					
				OCR	Spatial	Egocentric	Perception	Reasoning	Overall
1 stage		-	-	42.0	46.5	54.9	50.0	49.5	48.7
		-	-	28.8	29.2	18.8	35.5	22.6	30.8
2 stage			-	52.2	54.5	55.9	60.3	58.4	56.3
			-	51.6	54.5	58.0	62.1	60.2	<u>57.8</u>
			-	42.2	48.9	61.7	43.7	52.2	48.1
3 stage				53.0	52.5	64.9	59.8	65.9	<b>59.2</b>
				44.2	37.5	43.9	56.6	38.5	44.2

**Table 3 Training schedules.** An overview of the seven different training schedules evaluated, highlighting whether the LLM and vision encoders are frozen or unfrozen during each stage and the types of data used for training. and indicate whether a module is frozen or trainable, respectively. For each training schedule, three hyperparameters were tested, and we report the best-performing model.



**Figure 6 The effect of data mixture on performance.** We find that having  $\sim 10 - 14\%$  text data is important for video understanding performance. Out of our 14% data mixtures, we tested image-heavy, balanced, and video-heavy mixtures and found that video-heavy mixtures performed the best.

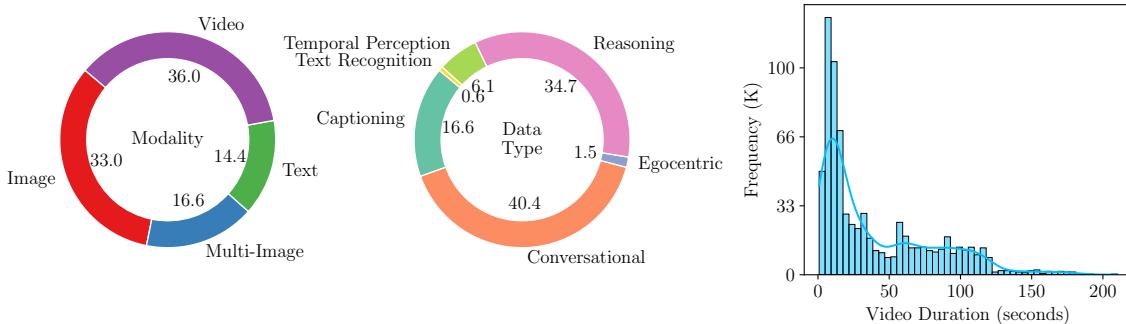
egocentric reasoning performance while the rest of the metrics remain largely unaffected, most likely due to better fine-grained vision-language alignment. These insights are in line with Zhao et al. (2024b)'s report.

**Finding 9:** Finetuning video encoders on only video data further improves overall performance, especially on reasoning and domain-specific tasks.

### 5.3 Data composition

The composition of the training data plays a significant role in the performance of LMMs, as illustrated by Zhang et al. (2024a). We investigated how the text, image, and video data mixtures affected video-LMMs performance. Specifically, we randomly selected several data compositions, as illustrated in Fig. 6. As can be seen, including 10  $\sim$  14% text data in the training mix is required for performance. This likely alleviates catastrophic forgetting. Increasing the proportion of text data beyond 14% to 25%, or decreasing it below 7%, harmed performance. Beyond including text data, having a slightly video-heavy mix of the remaining modalities was preferable. This balance allows the model to learn from higher-quality and diverse image datasets (Li et al., 2024a; Lin et al., 2023).

**Finding 10:** Data mixture matters, and including a moderate amount of text data and maintaining a slight video-heavy mix leads to optimal performance.



**Figure 7 Data statistics of the fine-tuning dataset.** **(Left)** Breakdown of data modalities, including text, image, multi-image, and video, illustrating the composition of the fine-tuning dataset. **(Middle)** Distribution of video annotation types, highlighting the proportions of Conversational, Reasoning, Egocentric, Temporal Perception, OCR, and Captioning annotations. **(Right)** Histogram of video durations, showing the distribution of durations in the training dataset.

## 6 Apollo: a family of state-of-the-art large multimodal models

We leverage the findings from our studies and train a family of video-centric Large Multimodal Models (LMMs), Apollo. Apollo models have state-of-the-art performance across multiple model sizes, frequently outperforming models  $2-3\times$  their size. We employed the Qwen2.5 (Yang et al., 2024) series of Large Language Models (LLMs) at varying scales to serve as the backbone for Apollo. Specifically, we utilized models with 1.5B, 3B, and 7B parameters. Following our analysis in Sec. 4, we used a SigLIP-SO400M (Zhai et al., 2023) encoder combined with an InternVideo2 (Wang et al., 2024d) video encoder. Features from each encoder are interpolated and concatenated along the channel dimension before being resampled to 32 tokens/frame using a Perciver Resampler (Jaegle et al., 2021). We utilized the 3-stage training schedule discussed in Sec. 5.1.

We curated a diverse mixture of publicly available and licensed datasets spanning text, image-text, multi-image, and video modalities. Due to licensing constraints, we omitted non-permissive sources (e.g., those reliant on ChatGPT), limiting the inclusion of some commonly used datasets. To further enhance our training corpus, we generated multi-turn video-based conversations via an annotation tool powered by LLaMA 3.1 70B (Touvron et al., 2023). Figure 7 provides a detailed overview of our data composition and statistics.

We evaluated Apollo across a suite of benchmarks to assess its performance in video-language understanding tasks, including TempCompass (Liu et al., 2024c), MLVU (Zhou et al., 2024), Perception-Test (Patraucean et al., 2023), Video-MME (Fu et al., 2024), LongVideoBench (Wu et al., 2024), and ApolloBench. As shown in Tables 4, Apollo models demonstrate strong performance across benchmarks. Notably, Apollo-3B outperforms several recently introduced 7B models, such as Oryx-7B (Liu et al., 2024d), Kangaroo (Liu et al., 2024b), and Video-XL-7B (Shu et al., 2024). For instance, on the MLVU benchmark, Apollo-3B achieves a score of 68.7, surpassing Oryx-7B’s 67.5. Similarly, Apollo-1.5B outperforms models larger than itself, including Phi-3.5-Vision (4.2B parameters) and some 7B models like LongVA-7B (Zhang et al., 2024e), indicating that smaller models can suffice for proof-of-concept implementations. We hope these results will motivate the field to utilize such smaller models for faster prototyping in the future.

Furthermore, Apollo-7B establishes a new performance frontier for models at the 7B scale, rivaling and even surpassing models with over 30B parameters such as Oryx-34B and VILA1.5-40B (Lin et al., 2024). On the MLVU benchmark, for instance, Apollo-7B scores 70.9, narrowly outperforming Oryx-34B’s 70.8. These gains highlight the potency of our design insights and confirm that carefully chosen architectural and training strategies can yield substantial improvements without resorting to larger model sizes.

Model	Existing Benchmarks					ApolloBench					
	TempCompass	MLVU	PerceptionTest		VideoME	L-VideoBench	OR	Egocentric	Spatial	Perception	Reasoning
			mc	m-avg	val	wo/w sub.	val				
<i>Proprietary</i>											
GPT-4V ( <a href="#">OpenAI, 2023</a> )	-	49.2	-	59.9/63.3	61.3	65.7	55.0	70.8	41.0	44.7	58.7
GPT-4o ( <a href="#">OpenAI, 2024</a> )	70.9	64.6	-	71.9/77.2	66.7	76.0	69.2	90.1	82.0	83.1	79.8
Gemini-1.5-Flash ( <a href="#">Team et al., 2023</a> )	-	-	-	70.3/75.0	61.6	-	-	-	-	-	-
Gemini-1.5-Pro ( <a href="#">Team et al., 2023</a> )	69.3	-	-	75.0/81.3	64.0	74.5	77.1	79.5	85.1	88.1	80.6
Claude-3.5-Sonnet ( <a href="#">Anthropic, 2024</a> )	-	36.5	-	60.0/62.9	-	-	-	-	-	-	-
<i>Open-weight</i>											
Qwen2VL-2B ( <a href="#">Wang et al., 2024a</a> )	60.6	59.5	53.9	55.6/60.4	48.5	29.0	29.0	47.0	50.0	46.0	40.2
Qwen2VL-7B ( <a href="#">Wang et al., 2024a</a> )	68.5	65.5	62.3	63.3/69.0	55.6	57.4	67.5	63.7	71.2	67.9	66.0
Qwen2VL-72B ( <a href="#">Wang et al., 2024a</a> )	-	-	68.0	71.2/77.8	-	-	-	-	-	-	-
Aria 8x3.5B ( <a href="#">Li et al., 2024b</a> )	69.9	-	53.9	67.6/72.1	64.2	-	-	-	-	-	-
Pixtral-12B ( <a href="#">Agrawal et al., 2024</a> )	-	-	-	40.7/47.5	44.9	-	-	-	-	-	-
<i>Open-source</i>											
LLaVA-OV-0.5B ( <a href="#">Li et al., 2024a</a> )	53.2	50.3	49.2	44.0/43.5	45.8	38.0	27.0	28.0	20.0	38.0	30.0
VILA1.5 3B ( <a href="#">Lin et al., 2024</a> )	56.1	44.4	49.1	42.2/44.2	42.9	31.7	33.0	29.3	38.0	44.7	36.1
InternVL2-2B ( <a href="#">Li et al., 2024a</a> )	53.4	48.2	49.6	30.8/-	44.8	40.8	46.3	34.3	44.7	45.3	42.1
Phi-3.5-Vision-4.2B ( <a href="#">Abdin et al., 2024</a> )	-	-	-	50.8/-	-	-	-	-	-	-	-
LongVU 3.2B ( <a href="#">Shen et al., 2024</a> )	-	<u>55.9</u>	-	<u>51.5</u> /-	-	-	-	-	-	-	-
<b>Apollo-1.5B</b>	<b>60.8</b>	<b>63.3</b>	<b>61.0</b>	<b>53.0/54.6</b>	<b>54.1</b>	<b>49.0</b>	<b>63.3</b>	<b>50.0</b>	<b>66.5</b>	<b>57.4</b>	<b>57.0</b>
LongVA-7B ( <a href="#">Zhang et al., 2024e</a> )	-	56.3	-	52.6/54.3	-	32.4	43.1	41.0	37.7	51.1	41.5
XComposer-8B ( <a href="#">Zhang et al., 2024d</a> )	-	37.3	34.4	55.8/58.8	-	<b>50.7</b>	42.0	54.7	54.7	40.5	48.6
Kangaroo-8B ( <a href="#">Liu et al., 2024b</a> )	61.3	61.0	-	<u>56.0</u> /57.6	54.2	-	-	-	-	-	-
Video-XL 7B ( <a href="#">Shu et al., 2024</a> )	-	64.9	-	55.5/ <b>61.0</b>	49.5	-	-	-	-	-	-
Oryx 7B ( <a href="#">Liu et al., 2024d</a> )	-	<u>67.5</u>	-	50.3/55.3	<b>55.5</b>	-	-	-	-	-	-
<b>Apollo-3B</b>	<b>62.5</b>	<b>68.7</b>	<b>65.0</b>	<b>58.4/60.6</b>	<b>55.1</b>	<b>49.6</b>	<b>68.6</b>	<b>59.3</b>	<b>67.0</b>	<b>68.4</b>	<b>62.7</b>
InternVL2-8B ( <a href="#">Chen et al., 2024b</a> )	65.3	50.8	57.4	54.0/56.9	51.8	50.0	48.4	54.3	57.7	51.8	52.8
LLaVA-OV-7B ( <a href="#">Li et al., 2024a</a> )	64.8	64.7	57.1	58.2/61.5	56.4	<b>56.0</b>	<b>69.1</b>	<b>69.0</b>	<u>63.3</u>	<u>63.2</u>	<u>64.0</u>
LongVU 7B ( <a href="#">Shen et al., 2024</a> )	-	65.4	-	60.6/-	-	-	-	-	-	-	-
LLaVA-N-Video-32B ( <a href="#">Zhang et al., 2024f</a> )	-	39.3	59.4	60.2/63.0	50.5	-	-	-	-	-	-
Oryx 34B ( <a href="#">Liu et al., 2024d</a> )	-	<u>70.8</u>	-	53.9/58.0	<b>62.2</b>	-	-	-	-	-	-
VILA-1.5-40B ( <a href="#">Lin et al., 2024</a> )	-	56.7	54.0	60.1/61.1	-	-	-	-	-	-	-
InternVL2-34B ( <a href="#">Chen et al., 2024b</a> )	-	59.9	-	<u>61.2</u> /62.4	-	-	-	-	-	-	-
<b>Apollo-7B</b>	<b>64.9</b>	<b>70.9</b>	<b>67.3</b>	<b>61.3/63.3</b>	<b>58.5</b>	<b>51.6</b>	<b>68.4</b>	<b>67.5</b>	<b>69.8</b>	<b>71.2</b>	<b>66.3</b>

**Table 4 Performance of Apollo on a diverse range of video benchmarks.** We compare Apollo to both proprietary and open-source models across multiple benchmark suites and our curated ApolloBench. **(Top)** Apollo-1.5B surpasses various small LMMs, including those with larger parameter counts (e.g., LongVU 3.2B), demonstrating robust gains even at relatively modest scale. **(Middle)** Apollo-3B maintains impressive results and competes effectively with recent 7B models such as Oryx and Video-XL, underscoring the efficiency of our design decisions in bridging performance gaps without massive scaling. **(Bottom)** Apollo-7B attains state-of-the-art performance among models of a similar size and even outperforms some models with over 30B parameters, highlighting its robustness.

## 7 Background

*Video Large Multimodal Models.* Early video-LMMs ([Yang et al., 2022](#); [Zhu et al., 2023b](#); [Maaz et al., 2023](#); [Xu et al., 2024a](#)) relied on sparsely sampled frames and MLP connectors or entirely training-free methods ([Kim et al., 2024](#); [Wu, 2024](#)). To address token count and support long-form video understanding, subsequent works introduced resampling methods such as spatio-temporal pooling ([Zhang et al., 2023](#); [Shen et al., 2024](#); [Xu et al., 2024a](#); [Jin et al., 2024](#); [Zhang et al., 2024g](#); [Xu et al., 2024b](#)). Most approaches ([Fei et al., 2024](#); [Jin et al., 2024](#); [Liu et al., 2024b,d](#); [Shen et al., 2024](#)) use image-based encoders, with only a few ([Lin et al., 2023](#); [Chen et al., 2024b](#); [Li et al., 2023b](#)) employing video-specific encoders to capture temporal dependencies.

Training schedules typically involve alignment followed by supervised fine-tuning (Lin et al., 2023; Li et al., 2024a; Zhang et al., 2024c; Shu et al., 2024) to adapt connectors and LLMs for video understanding. Earlier video-LMMs (Lin et al., 2023; Zhang et al., 2023) were trained on small-scale video instruction datasets, while recent efforts have expanded both dataset scale (Zhang et al., 2024g; Liu et al., 2024d) and quality (Zhang et al., 2024h), leveraging multi-image datasets (Lin et al., 2024; Li et al., 2024a; Shu et al., 2024) to enhance model capabilities further. Benchmarks have also evolved, shifting from short-video tasks (Yu et al., 2019; Xu et al., 2017, 2016) to long-video tasks (Zhou et al., 2024; Wu et al., 2024; Cai et al., 2024). Despite these advances, many design decisions in video-LMM remain with limited analysis or justification. This work addresses these gaps by systematically exploring the design space for video-LMMs.

*Design Exploration for Large Multimodal Models.* Recent studies have highlighted the importance of systematically exploring the design space for image-based LMMs (Karamchetti et al., 2024; Laurençon et al., 2024b; Tong et al., 2024; Shi et al., 2024), focusing on key components such as encoder selection, training strategies, and data mixtures. Laurençon et al. (2024b) introduced perceiver resamplers as an effective method for reducing token counts and enabling efficient long-context modeling. Karamchetti et al. (2024); Tong et al. (2024) examined trade-offs between single versus multiple encoders, training versus freezing encoders, and one-stage versus multi-stage training, with Cambrian-1 also analyzing the influence of data mixtures and vision-centric benchmarking. Shi et al. (2024) extended these efforts by evaluating the impact of diverse encoder architectures and their combinations. While these works provide a strong foundation for image-based LMMs, the design space for video-LMMs remains underexplored. Unlike images, videos require specialized strategies for frame sampling, token resampling, encoder selection, and efficient training and evaluation. This work addresses these gaps by systematically investigating the unique challenges and opportunities in designing video-LMMs, paving the way for scalable and effective solutions in video understanding.

## 8 Conclusion

In the study, we critically evaluated the current state of the video Large Multimodal Model (video-LMM) field, from architecture design and training schedules to data mixtures and evaluation. In part, we hope that concepts such as Scaling Consistency encourage researchers to utilize smaller LMMs in their research, while ApolloBench will allow for faster and more comprehensive evaluation. We hope our insights into the key aspects of video-LMM design, encompassing video sampling, encoder selection, token resampling, and token integration, will further democratize video-LMM research, further accelerating research in the field.

Building upon these insights, we developed Apollo, a family of state-of-the-art LMMs capable of advanced video-language understanding. Notably, Apollo-3B outperforms most advanced 7B models, while Apollo-7B outperforms all 7B models and many recent 30B models. Our findings highlight that careful design and training strategies can yield superior performance without necessitating larger model sizes. We believe that our work provides valuable guidelines and resources for future research, advancing the development of efficient and effective video-LMMs.

## References

- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.
- Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning*, pages 265–279. PMLR, 2023.
- Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Moncault, Saurabh Garg, Theophile Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- Anthropic. Claude-3.5. <https://www.anthropic.com/news/clause-3-5-sonnet>, 2024.
- Kirolos Atallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. *arXiv preprint arXiv:2407.12679*, 2024.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *arXiv preprint arXiv:2404.08471*, 2023.
- Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Shyamal Buch, Cristobal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the “Video” in Video-Language Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.
- Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024a.
- Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.
- Yifan Du, Yuqi Huo, Kun Zhou, Zijia Zhao, Haoyu Lu, Han Huang, Wayne Xin Zhao, Bingning Wang, Weipeng Chen, and Ji-Rong Wen. Exploring the design space of visual context representation in video mllms. *arXiv preprint arXiv:2410.13694*, 2024.
- Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024.
- Chaoyou Fu, Yuhan Dai, Yondong 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.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

- Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.
- Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univ: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024.
- Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning (ICML)*, 2024.
- 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.
- Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Vidcompress: Memory-enhanced temporal compression for video understanding in large language models. *arXiv preprint arXiv:2410.11417*, 2024.
- Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. *arXiv preprint arXiv:2408.12637*, 2024a.
- Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024b.
- 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*, 2024a.
- 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*, 2024b.
- 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*, 2024c.
- KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023a.
- Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2023b.
- 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*, pages 323–340. Springer, 2025.
- 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*, 2024d.
- Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.
- Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024a.
- Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024b.
- 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*, 2024c.
- 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*, 2024d.
- Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.
- OpenAI. Gpt-4v. <https://openai.com/index/gpt-4v-system-card/>, 2023.
- OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and Joao Carreira. Perception test: A diagnostic benchmark for multimodal video models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 42748–42761. Curran Associates, Inc., 2023. [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/8540fba4abdc7f9f7a7b1cc6cd60e409-Paper-Datasets\\_and\\_Benchmarks.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/8540fba4abdc7f9f7a7b1cc6cd60e409-Paper-Datasets_and_Benchmarks.pdf).

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

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024.

Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Shengbang Tong, Ellis L Brown II, Penghao Wu, Sanghyun Woo, ADITHYA JAIRAM IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. <https://openreview.net/forum?id=Vi8AepAXGy>.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Guangzhi Wang, Yixiao Ge, Xiaohan Ding, Mohan Kankanhalli, and Ying Shan. What makes for good visual tokenizers for large language models? *arXiv preprint arXiv:2305.12223*, 2023.

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*, 2024a.

Weihang 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*, 2024b.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*, 2024c.

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*, 2024d.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024e.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. <https://arxiv.org/abs/2407.15754>.

Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024.

- Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, June 2021.
- Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueling Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.
- 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.
- Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, and Jiashi Feng. Plava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv e-prints*, pages arXiv–2404, 2024a.
- Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024b.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022.
- Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023.
- Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueling 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.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.
- Yufei Zhan, Yousong Zhu, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon v2: Advancing multimodal perception with high-resolution scaling and visual-language co-referring. *arXiv preprint arXiv:2403.09333*, 2024.
- Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2023.
- Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024a.
- Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024b. <https://arxiv.org/abs/2407.12772>.
- Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, Songyang Zhang, Wenwei Zhang, Yining Li, Yang Gao, Peng Sun, Xinyue Zhang, Wei Li, Jingwen Li, Wenhui Wang, Hang Yan, Conghui He, Xingcheng Zhang, Kai Chen, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024c.
- Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024d.
- 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*, 2024e.
- 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, April 2024f. <https://llava-vl.github.io/blog/2024-04-30-llava-next-video/>.
- Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024g. <https://arxiv.org/abs/2410.02713>.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024h. <https://arxiv.org/abs/2410.02713>.

Tiancheng Zhao, Qianqian Zhang, Kyusong Lee, Peng Liu, Lu Zhang, Chunxin Fang, Jiajia Liao, Kelei Jiang, Yibo Ma, and Ruochen Xu. Omchat: A recipe to train multimodal language models with strong long context and video understanding. *arXiv preprint arXiv:2407.04923*, 2024a.

Yue Zhao, Long Zhao, Xingyi Zhou, Jialin Wu, Chun-Te Chu, Hui Miao, Florian Schroff, Hartwig Adam, Ting Liu, Boqing Gong, et al. Distilling vision-language models on millions of videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13106–13116, 2024b.

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.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023a.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023b.

Orr Zohar, Xiaohan Wang, Yonatan Bitton, Idan Szpektor, and Serena Yeung-Levy. Video-star: Self-training enables video instruction tuning with any supervision. *arXiv preprint arXiv:2407.06189*, 2024.

# Appendix

This document provides more details of our approach and additional experimental results, organized as follows:

- § [B Analyzing the benchmarks](#). We provide an in-depth analysis of the factors affecting evaluations, such as video duration and format. We then give a detailed overview of how we curated ApolloBench.
- § [C Apollo implementation details](#). We provide an in-depth description of Apollo, along with all the hyperparameters needed to reproduce Apollo.
- § [D Scaling Consistency](#). We provide an in-depth analysis of the correlations between models of different sizes, compare Scaling Consistency to traditional scaling laws, and motivate their usage in future experiments.
- § [E Video sampling analysis](#). We expand on our Video Sampling experiments and add a per-metric breakdown.
- § [F Raw results](#). We provide all the raw data used in our study for further analysis. For Sec. 3: Tab. [14](#) & [15](#), Sec. 4.1: Tab. [10](#) & [11](#), Sec. 4.2: Tab [9](#), Sec. 4.4: Tab. [2](#), Sec. 5.1: Tab. [12](#), Sec. 5.3: Tab. [13](#).

## Part

### Table of Contents

---

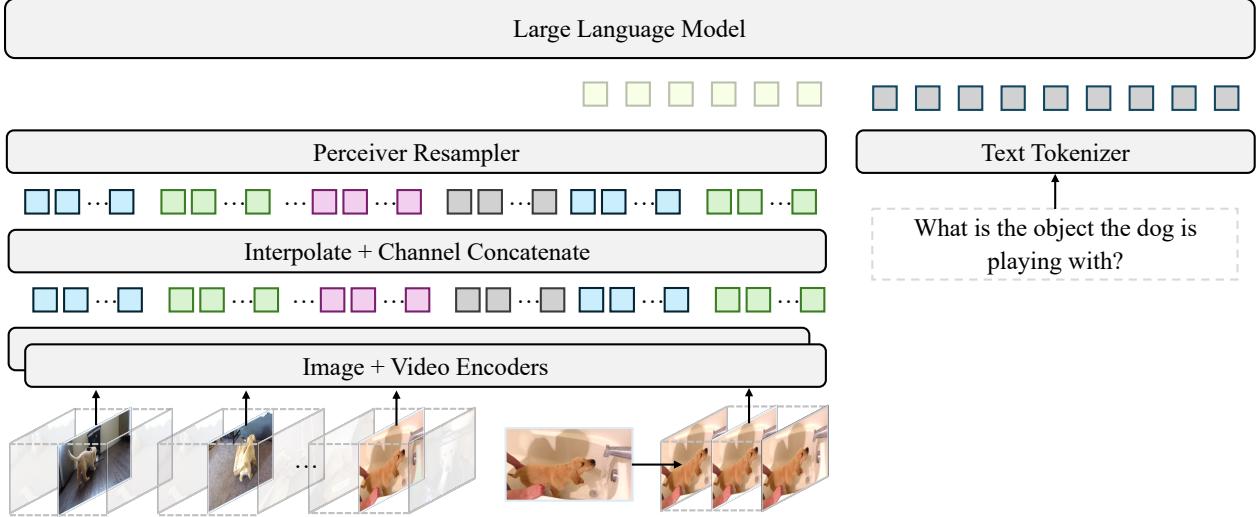
<b>A Future work</b>	<b>20</b>
<b>B Analyzing the benchmarks</b>	<b>21</b>
B.1 Correlations within existing benchmarks . . . . .	21
B.2 Raw evaluations . . . . .	23
B.3 ApolloBench curation . . . . .	23
<b>C Apollo implementation details</b>	<b>24</b>
C.1 Architecture . . . . .	24
C.2 Unified vs. Split Architectures . . . . .	25
C.3 Data . . . . .	26
C.4 Training . . . . .	26
<b>D Scaling Consistency: efficient model design with smaller models</b>	<b>27</b>
<b>E Effect of video sampling on the different dimensions of video perception</b>	<b>27</b>
<b>F Raw results</b>	<b>28</b>

---

### A Future work

Several promising directions emerge from our study on Large Multi-modal Models (LMMs). First, we employed a fully unified architecture, using the video encoder for videos and images by replicating images N times. Exploring separated architectures, where images are processed with an image encoder and videos with both image and video encoders, could reveal performance benefits and better modality handling.

Second, in separated architectures, training the video and image encoders during supervised fine-tuning (SFT) and evaluating their individual contributions to performance could identify optimal training strategies. Similarly, training both encoders on mixed image and video data within unified architectures may help determine which encoder influences observed performance drops, enabling targeted improvements.



**Figure 8 Apollo architecture overview.** Apollo encodes clips of  $N$  (dependent on the video encoder) frames. Output features are interpolated and concatenated along the channel dimension before being fed to a connector. The connector up-projects the features to the Large Language Models’ hidden dimension and then resamples them into a pre-set number of  $T$  tokens/clip. Images are duplicated  $N$  times and encoded the same way as video clips.

Further investigation into Scaling Consistency is necessary to confirm its applicability across a broader range of model sizes, ensuring its reliability for even larger models. We did not explore memory-based LMM approaches, such as memory banks or frame retrieval methods like text-conditioned pooling in Q-Former. Evaluating these techniques could test our hypothesis that they may not generalize well to multi-turn conversations.

Lastly, current benchmarks primarily use academic multiple-choice formats, which inadequately assess conversational abilities. Developing a dedicated conversational evaluation benchmark for LMMs is essential to more accurately measure and enhance models’ dialogue performance in real-world scenarios.

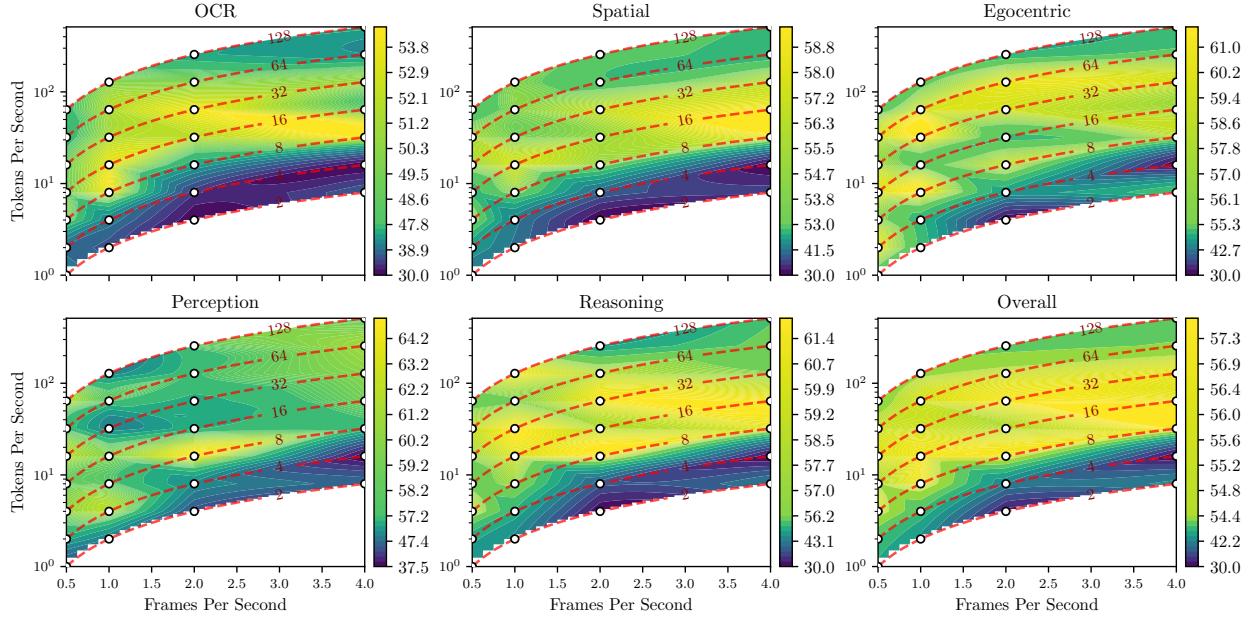
## B Analyzing the benchmarks

### B.1 Correlations within existing benchmarks

*Video Duration.* We were interested in how video length affected model performance to see if existing benchmarks test long video perception capabilities. In the large language model field, testing long-context has been non-trivial, where many benchmarks do not need information integration across the entire model’s context window and instead devolve to effectively needle-in-a-haystack style experiments. We hypothesized that long video benchmarks may behave similarly. As such, we compared Video-MME short/medium/long and LongVideoBench’s different duration groups (see Fig. 13 and Fig. 14). We found that the two are highly

Model	NExT-QA			Perception-Test			TempCompass (CM)			TempCompass (MC)			TempCompass (YN)		
	Video	Image	Text	Video	Image	Text	Video	Image	Text	Video	Image	Text	Video	Image	Text
InternVL2 2B <a href="#">Chen et al. (2024b)</a>	68.9	61.1	42.8	49.6	46.0	38.6	67.2	63.3	51.9	53.4	47.5	35.9	62.3	59.3	51.2
InternVL2 8B <a href="#">Chen et al. (2024b)</a>	70.8	72.6	49.1	57.4	52.8	41.3	77.4	66.9	58.3	65.3	54.9	43.7	68.6	62.6	52.1
LLaVA-OV 0.5B <a href="#">Li et al. (2024a)</a>	57.3	50.7	31.9	49.1	44.8	40.4	61.9	58.9	51.3	53.2	44.6	34.1	60.0	55.9	49.7
LLaVA-OV 7B <a href="#">Li et al. (2024a)</a>	79.3	70.0	48.7	57.1	49.7	41.4	73.8	60.8	56.8	64.9	51.6	41.4	69.8	57.8	53.3
LongVA 7B <a href="#">Zhang et al. (2024e)</a>	50.2	38.9	36.6	50.6	50.3	50.1	60.7	51.1	50.9	56.1	52.2	50.7	62.9	61.6	60.9
Qwen2-VL 2B <a href="#">Wang et al. (2024a)</a>	68.7	62.1	44.0	53.1	47.5	39.8	70.9	62.5	54.3	60.6	50.4	40.1	63.7	58.6	52.3
Qwen2-VL 7B <a href="#">Wang et al. (2024a)</a>	78.9	68.5	42.6	58.9	52.6	38.4	76.6	64.3	56.5	67.2	52.3	41.6	71.9	61.8	54.0
VILA-1.5 3B <a href="#">Lin et al. (2024)</a>	56.9	56.7	30.1	49.1	49.1	36.2	66.3	66.3	52.9	56.1	56.1	36.8	63.4	63.4	51.1
VILA-1.5 8B <a href="#">Lin et al. (2024)</a>	63.1	63.1	38.2	54.7	54.7	41.2	58.7	58.7	33.6	49.0	49.0	18.8	62.5	62.5	50.6
XComposer-8B <a href="#">Zhang et al. (2024c)</a>	71.1	47.3	41.0	55.9	45.3	39.6	72.2	59.3	49.2	61.1	39.4	31.7	64.5	57.8	52.3

**Table 5 Benchmark evaluation for different models across input modalities (1/2).** This table reports the performance of various models on the NExT-QA, Perception-Test, and TempCompass benchmarks with video, image, and text inputs.



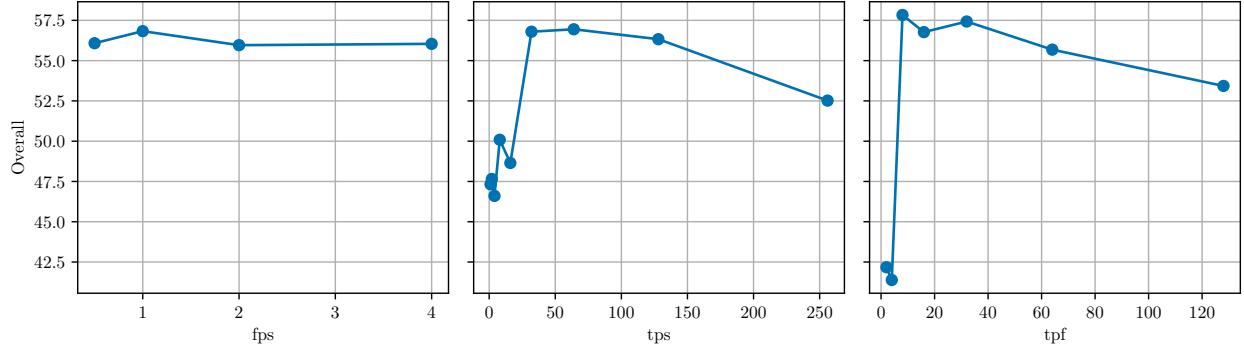
**Figure 9 Video fps sampling analysis.** Full analysis on the effect of frames per second (fps, x-axis), tokens per second (tps, y-axis), and tokens per frame (tpf, dotted red lines) on each of ApolloBench’s dimensions. The number of tokens/frames is highlighted via the dotted red lines.

correlated, where  $R^2 > 0.92$  between all duration groups in LongVideoBench (Fig. 14). On Video-MME, whether using or not using subtitles,  $R^2 > 0.83$ . When closely examining Video-MME short/medium/long in Fig. 2, one can see that the most significant difference between them is the video modality performance decreasing, with text and image modalities being mostly unchanged. This indicated a greater and greater reliance on the text model’s performance rather than any vision capabilities.

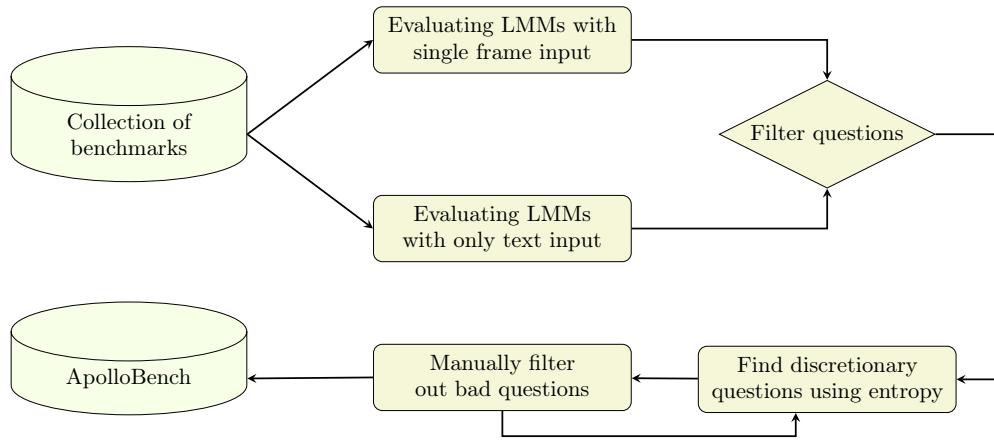
*Question types.* There are currently two prevalent methods for evaluating LMMs—either open-ended questions or close-ended (multiple choice, yes/no). Scoring open-ended QA is challenging because the score is ultimately subjective. The dominant way of evaluating open-ended QA is using another language model (e.g., chatGPT) to rate the prediction and decide if it is correct. As shown by Wu (2024), GPT versioning strongly impacts the resulting scores that are even 10% apart. As such, recent trends show greater reliance on multiple-choice QA. However, are we losing something when evaluating methods only on multiple-choice? As seen in Fig. 12, we find these are highly correlated, with  $R^2 > 0.81$ . While multiple-choice appears to be a good option for benchmarking the video perception capabilities of video-LMMs, models overly optimized to multiple-choice will not be good conversational agents. As such, a benchmark focusing solely on a conversation is needed, ideally, one that does not suffer from high API costs and GPT versioning noise.

Model	LongVideoBench			MLVU			Video-MME (Long)			Video-MME (Medium)			Video-MME (Short)		
	Video	Image	Text	Video	Image	Text	Video	Image	Text	Video	Image	Text	Video	Image	Text
InternVL2 2B Chen et al. (2024b)	44.8	37.9	32.8	48.2	41.5	32.6	33.1	30.9	31.4	38.2	32.2	28.7	51.3	39.1	32.8
InternVL2 8B Chen et al. (2024b)	51.8	45.0	40.2	50.8	40.0	37.5	42.0	40.0	38.6	50.6	39.6	38.6	62.1	48.2	39.4
LLaVA-OV 0.5B Li et al. (2024a)	46.0	40.5	37.4	50.3	39.2	35.3	37.2	31.3	33.1	40.0	32.0	30.2	54.6	37.1	30.1
LLaVA-OV 7B Li et al. (2024a)	56.5	45.1	41.2	65.1	50.3	45.5	49.9	36.9	39.8	54.6	39.4	38.3	70.9	47.4	40.2
LongVA 7B Zhang et al. (2024e)	45.2	44.2	43.0	51.9	45.1	44.1	41.4	38.1	36.7	45.9	39.9	38.4	55.1	45.3	40.0
Qwen2-VL 2B Wang et al. (2024a)	48.5	40.8	40.4	59.5	45.1	38.4	43.2	36.9	33.3	51.0	35.0	32.3	65.3	40.4	34.8
Qwen2-VL 7B Wang et al. (2024a)	54.8	44.7	41.5	65.5	49.1	42.4	49.8	40.0	38.4	57.6	41.2	39.2	70.7	46.3	37.6
VILA-1.5 3B Lin et al. (2024)	42.9	42.9	33.8	23.3	23.3	13.6	31.6	28.0	28.0	36.7	27.3	27.3	48.7	27.8	27.8
VILA-1.5 8B Lin et al. (2024)	47.2	47.2	37.1	44.4	44.4	31.1	39.3	36.6	36.6	42.1	32.3	32.3	56.3	34.3	34.3
XComposer-8B Zhang et al. (2024c)	47.6	30.0	32.0	37.2	8.5	7.3	46.4	28.0	35.1	50.9	26.3	35.0	66.0	28.1	36.1

**Table 6 Benchmark evaluation for different models across input modalities (2/2).** This table reports the performance of various models on the LongVideoBench, MLVU, and Video-MME benchmarks with video, image, and text inputs.



**Figure 10 Video fps sampling analysis.** Comparison of overall performance across different parameters. The first plot illustrates the impact of frames per second (fps) on performance, while the second and third plots show performance trends with varying tokens per second (tps) and tokens per frame (tpf), respectively.



**Figure 11 Flowchart illustrating the curation process of ApolloBench.** Starting with a collection of benchmarks, we evaluate Large Multimodal Models (LMMs) using the full video, single-frame, and text inputs. Questions requiring video perception were filtered based on model performance, and discretionary questions were identified using entropy. After manual verification and categorization into five temporal perception categories, the top 400 questions were selected for the benchmark, and manually inspected.

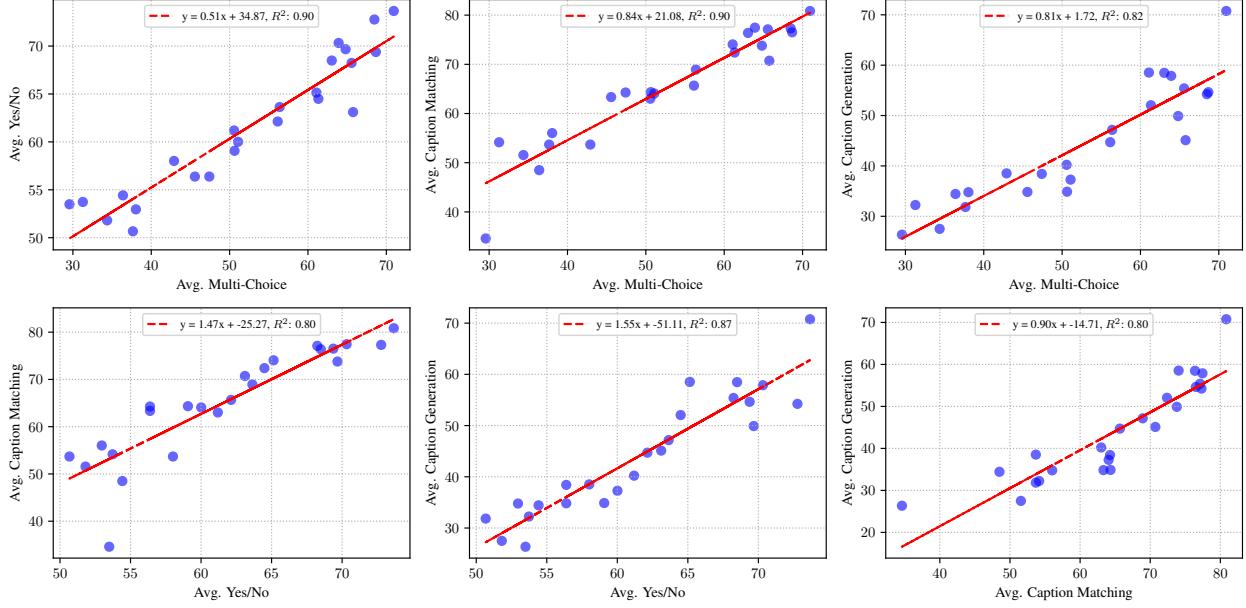
## B.2 Raw evaluations

We evaluated InvernVL2 2&8 B (Chen et al., 2024b), LLaVA LLaVA-OV 0.5 & 7B (Li et al., 2024a), VILA-1.5 1.5 3 & 8B (Lin et al., 2024), Qwen2-VL 2 & 7B (Wang et al., 2024a), LongVA 7B (Zhang et al., 2024e) and XComposer-8B (Zhang et al., 2024c) on NExTQA (Xiao et al., 2021), PerceptionTest (Patraucean et al., 2023), TempCompass (Liu et al., 2024c), Video-MME (Fu et al., 2024), MLVU (Zhou et al., 2024), and LongVideoBench (Wu et al., 2024). All evaluations were done using lmms-eval (Zhang et al., 2024b). Full evaluations of all models on the benchmarks can be seen in Tab. 5 & 6.

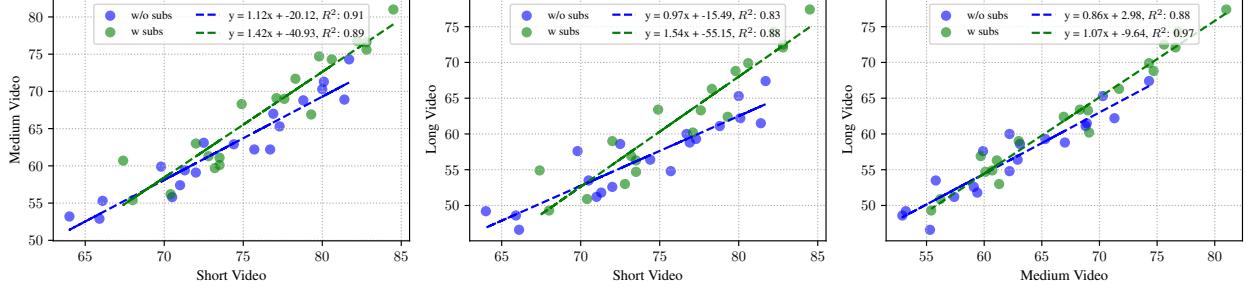
## B.3 ApolloBench curation

The creation process of ApolloBench is depicted in Fig. 11. The process begins with a collection of multiple-choice benchmarks. To eliminate the reliance on external tools like ChatGPT, we focus exclusively on multiple-choice questions, ensuring a cost-effective and consistent evaluation process Wu (2024).

We first evaluated several Large Multi-modal Models (LMMs) with text-only, center-frame, and full-video inputs. Questions that could be answered correctly by more than 50% of the models using either of these modalities were filtered out, as these questions did not require video perception. Next, we categorized the remaining questions into five temporal perception categories: Temporal OCR, Egocentric, Spatial, Perception, and Reasoning. Using entropy, we identified questions with high discrimination power between models and



**Figure 12 Effect of question type on model performance.** Correlations between different question types (multiple-choice, yes/no) on the TempCompass are shown. The high correlation indicates consistency in evaluating model performance across various question formats, indicating that multiple choice is a reasonable option in existing benchmarks.



**Figure 13 Correlation between Video-MME duration groups.** The correlations between short, medium, and long video duration groups on the Video-MME benchmark. The analysis highlights how model performance scales with video length, emphasizing the reliance on text and image modalities as video duration increases.

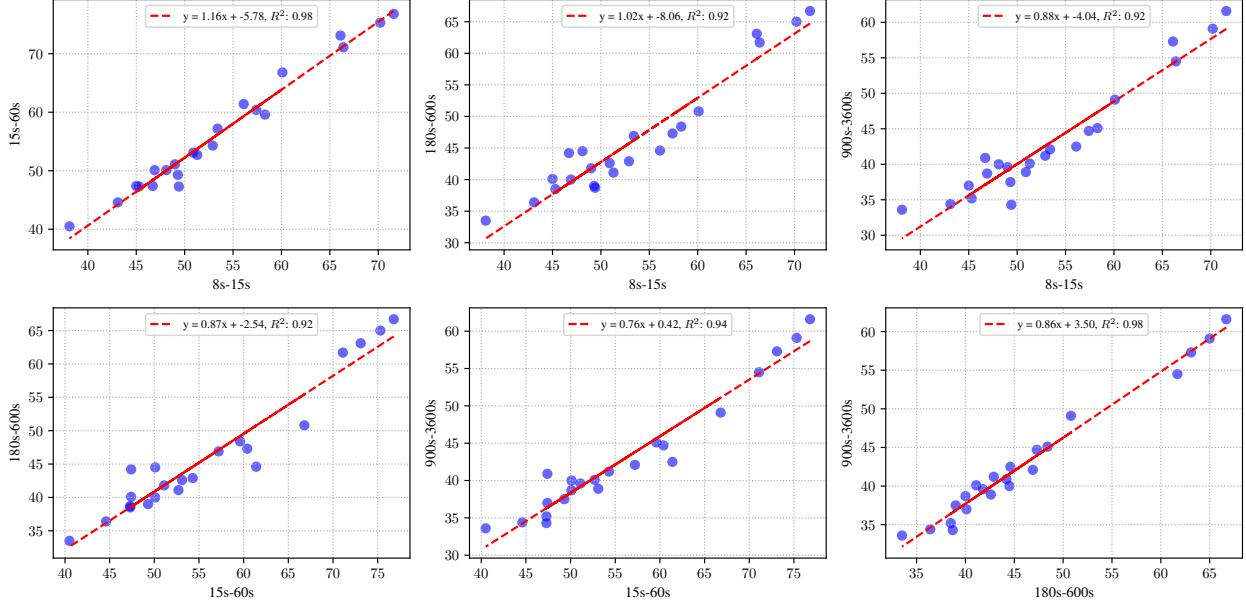
manually verified them to ensure accuracy and quality. From this, we selected the top 400 questions with the highest entropy to form the final ApolloBench dataset. This curated benchmark is 41× faster to evaluate compared to existing benchmarks while maintaining a high correlation with their results (see Fig. 2, right). Additionally, ApolloBench emphasizes video perception, as shown in Fig. 2, left.

## C Apollo implementation details

In this section, we provide detailed descriptions of all the design decisions in Apollo, including implementation specifics, hyperparameters, and other relevant details.

### C.1 Architecture

Apollo encodes clips consisting of  $N$  frames, where  $N$  depends on the video encoder used ( $= 4$  for InternVideo2+SigLIP-SO400M). We opted for a fully shared pipeline for both images and videos, so when encoding images, we replicate the image  $N$  times to match the clip length. Video frames are then encoded



**Figure 14 Correlation between LongVideoBench duration groups.** Correlations between different video duration categories on LongVideoBench are depicted, with  $R^2 > 0.92$  across groups. This consistency suggests that performance trends remain stable across varying video lengths.

independently with the InternVideo2 and SigLIP-SO400M encoders. The output features are interpolated and concatenated along the channel dimension before being fed to a connector module. The connector projects the features to match the hidden dimension of the Large Language Model, and the resampler resamples them into a predetermined number of  $T$  tokens per clip using the Perciver Resampler. An overview of Apollo is shown in Fig. 8. For vision-text token integration, we utilize the clip from {MM:SS}-{MM:SS}: <vid\_token>.

Apollo effectively samples videos as a series of independent clips. By keeping the clip sampling frames per second (fps) constant, the model learns to reason about fine-grained temporal aspects, such as the speed of objects. Many previous methods employ uniform frame sampling, especially when handling long videos, effectively changing the “playback speed” between iterations. In contrast, we sample clips uniformly spaced throughout the video, and if the video is too long, we distribute the individual clips uniformly rather than adjusting the frame sampling rate. We, therefore, sample clips concurrently until reaching the maximum number of clips (see Tab. 7), at which point we start uniformly distancing the clips.

## C.2 Unified vs. Split Architectures

While previous sections focused on different aspects of design and training protocols, we also investigated the impact of using a unified versus a split architecture for integrating image and video modalities. A unified architecture processes both image and video inputs through the same set of encoders and token resamplers, ensuring a single consistent visual representation path. In contrast, a split architecture separates the processing streams for images and videos, potentially offering more specialized representations at the cost of increased complexity. Previously, Lin et al. (2023) advocated for sharing the mlp connector between images and videos, claiming that this leads to better transfer. Jin et al. (2024) performed the same token merging and utilized the same connector for both images and videos. Recent works encode video frames entirely independently, completely removing the need for separate architectures for image and video inputs.

As shown in Tab. 8, our experiments revealed that the unified architecture performs slightly better or on par with the split architecture across key benchmarks. The unified approach strikes an appealing balance between performance and simplicity, offering a more elegant and parameter-efficient solution. Given these findings, we adopt the unified architecture as our default setting for Apollo.

	Align			Vision Pretraining			SFT			
	1.5B	3B	7B	1.5B	3B	7B	1.5B	3B	7B	
<i>Sampling</i>	<b>Max clips</b>	25	25	25	25	25	25	200	200	150
	fps	2	2	2	2	2	2	2	2	2
	tps	32	32	32	32	32	32	32	32	32
	tpf	16	16	16	16	16	16	16	16	16
<i>Data</i>	<b>Dataset</b>	A	A	A	VpT	VpT	VpT	SFT	SFT	SFT
	#Samples	198K	198K	198K	396K	396K	396K	3.2M	3.2M	3.2M
	Type	I+V	I+V	I+V	V	V	V	T+I+MI+V	T+I+MI+V	T+I+MI+V
<i>Model</i>	<b>Trainable</b>	38.4M	63.6M	177M	1.4B	1.5B	1.6B	1.6B	3.2B	7.8B
	$\psi_{\text{vision}}$	–	–	–	1.4B	1.4B	1.4B	–	–	–
	$\theta_{\text{connector}}$	38.4M	63.6M	177M	38.4M	63.6M	177M	38.4M	63.6M	177M
	$\phi_{\text{LLM}}$	–	–	–	–	–	–	1.54B	3.09B	7.62B
<i>Training</i>	<b>Batch Size</b>	256	256	256	256	256	256	256	256	256
	<b>LR: <math>\psi_{\text{vision}}</math></b>	0	0	0	$5 \times 10^{-6}$	$5 \times 10^{-6}$	$5 \times 10^{-6}$	0	0	0
	<b>LR: <math>\theta_{\text{connector}}</math></b>	$1 \times 10^{-4}$								
	<b>LR: <math>\phi_{\text{LLM}}</math></b>	0	0	0	0	0	0	$5 \times 10^{-5}$	$2 \times 10^{-5}$	$1 \times 10^{-5}$
	<b>Epoch</b>	1	1	1	1	1	1	1	1	1

**Table 7 Detailed configuration for each training stage of Apollo.** The table summarizes the maximum clips per video, frames per second (fps), dataset information, trainable parameters, and training hyperparameters across different stages of training (**Alignment**, **Vision pretraining**, **SFT**) for Apollo models of varying sizes (1.5B, 3B, and 7.6B).

Architectural Type	ApolloBench					
	OCR	Spatial	Egocentric	Perception	Reasoning	Overall
Split	46.2	55.7	62.3	59.0	58.1	56.2
Unified	50.0	54.0	61.7	60.8	57.9	<b>56.8</b>

**Table 8 Split vs Unified Architectures on ApolloBench.** A comparison of the performance across different tasks, including OCR, Spatial, Egocentric, Perception, and Reasoning, as well as the overall score.

### C.3 Data

We utilized a diverse mix of publicly available and licensed datasets across text, image-text, multi-image, and video modalities. Due to licensing restrictions, we excluded non-permissive datasets—such as those leveraging ChatGPT—which limited our inclusion of some commonly used datasets. We generated multi-turn conversations to enrich our training data by leveraging Large Multimodal Models (LMMs), such as Qwen2VL-7B, for captioning. Then, we used LLaMA 3.1 70B (Touvron et al., 2023) to convert these captions into conversations. Detailed data statistics are presented in Fig. 6. It is possible that performance could be further improved without such restrictions and by training on larger datasets like those introduced in LLaVA-OneVision Li et al. (2024a) and Cambrian1 Tong et al. (2024).

Our training process comprised three distinct stages:

1. **Alignment:** In this phase, we trained on a 198K mixture of 50/50 image and video captions.
2. **Vision Pretraining:** We tuned the encoders using a video-only caption dataset of 396K samples.
3. **Supervised Fine-tuning (SFT):** We trained on a mixture of text, image, multi-image, and video data, with a total of 3.2 million samples.

### C.4 Training

We trained our models using 128 NVIDIA A100 GPUs. Due to the large-scale nature of this study, we automated model training to be spawned from csv files, which would automatically update with the final evaluations. Most experiments were done with ZeRO2 optimization, as full model sharding was unnecessary for our models, but ZeRO3 is supported for future researchers interested in training larger models. We

utilized the AdamW optimizer for all training stages with a gradient clipping threshold of 1. We applied a warm-up ratio of 0.03 and a cosine learning rate schedule. The training objective was the cross-entropy loss for autoregressive text generation only. We adjusted the learning rates of the Large Language Model (LLM) components proportionally to the square root of their relative model sizes. We found that employing a higher learning rate for the connector module yielded the best performance.

## D Scaling Consistency: efficient model design with smaller models

Developing Large Multi-modal Models (LMMs) with billions of parameters is computationally intensive. A key question is whether smaller models can reliably inform design decisions for larger ones. We introduce Scaling Consistency, a phenomenon where design choices evaluated on moderately sized models (approximately 2–4 billion parameters) correlate highly with those on larger models, enabling efficient model development.

To investigate Scaling Consistency, we conducted extensive experiments varying key aspects of LMM design, such as architecture, video sampling, training strategies, and data mixtures. We selected 21 distinct model variations encompassing these design dimensions. Each variation was trained using four different Large Language Models: Qwen2-0.5B, Qwen2-1.5B, Qwen1.5-4B, and Qwen2-7B, resulting in a total of 84 models.

Unlike traditional scaling laws—which typically require training multiple models from within the same model family to understand how performance scales with size—Scaling Consistency allows us to transfer design insights without such extensive efforts. In scaling laws, researchers train around 3–5 models of different sizes to establish scaling relationships, and only then can they determine which design decisions are beneficial at larger scales. In contrast, Scaling Consistency shows that design decisions on moderately sized models transfer well to larger ones, even across different model families. Our primary goal is to show that design decisions transfer reliably, reducing computational burden and accelerating research.

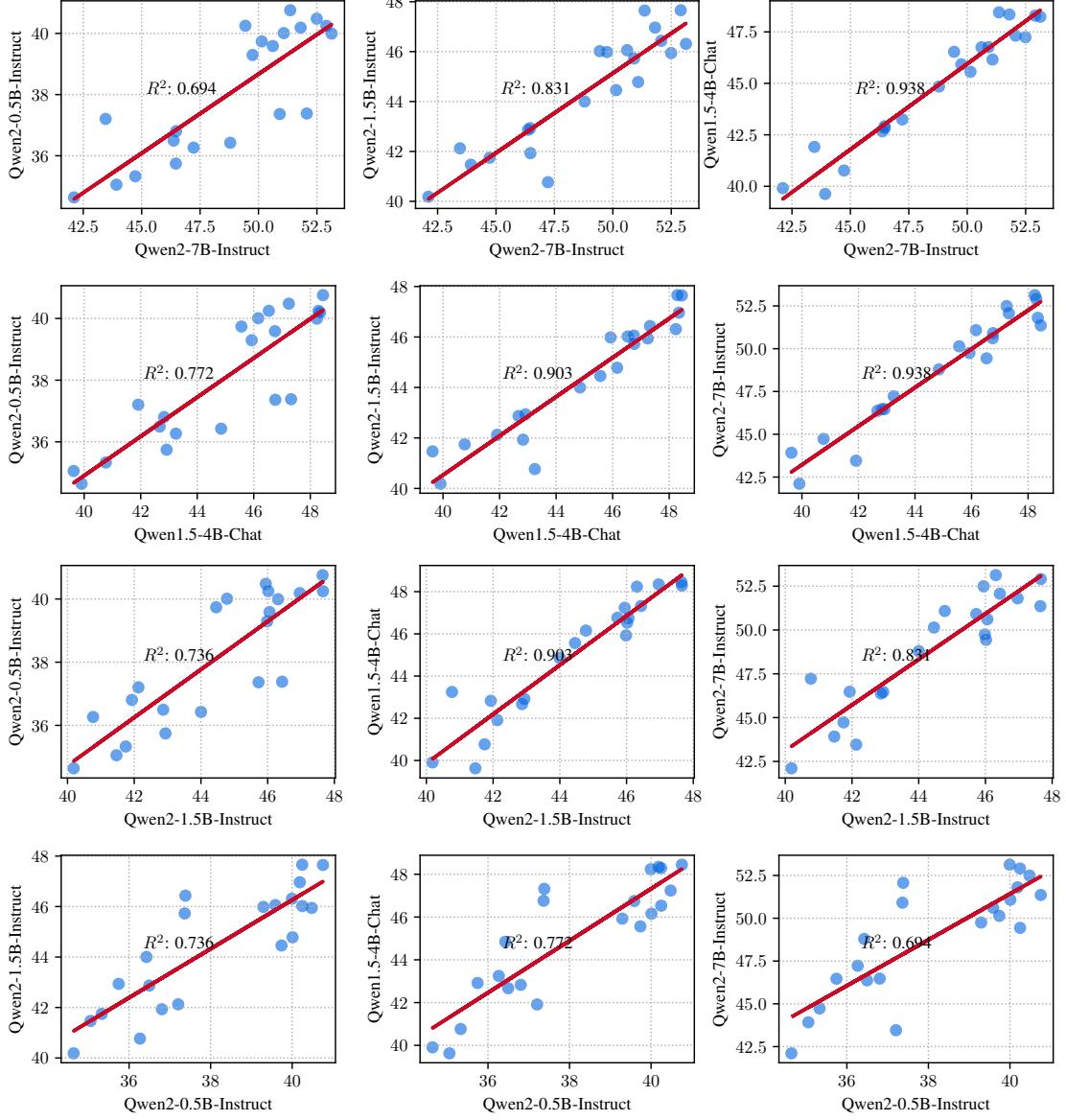
In Fig. 15, we present all the correlation plots from our study. When comparing the 7B model to smaller ones (first row), we observe that the  $R^2$  progressively increases with model size. A similar pattern is seen when comparing the 4B model to smaller models. For the 1.5B model, however, the  $R^2$  decreases when compared to larger models, and with the 0.5B model, the  $R^2$  is essentially random. We find that the  $R^2$  behaves log-linearly with model size. This suggests that at around 3 billion parameters, we can expect an  $R^2$  greater than 0.9 when compared with the 7B model. Since the behavior is log-linear, models above the 3–4 billion parameter range can be expected to have high correlation even with much larger models, such as 32B ( $> R^2 \simeq 0.86$ ) or 72B parameters ( $> R^2 \simeq 0.84$ ).

## E Effect of video sampling on the different dimensions of video perception

Fig. 9 presents a detailed analysis of how varying frames per second (fps) and tokens per second (tps) impact our model’s performance across different video perception tasks: Optical Character Recognition (OCR), Spatial Understanding, Egocentric Understanding, Perception, and Reasoning. Our findings indicate that OCR and Spatial Understanding tasks exhibit a uniform and steep decline in performance when tps is reduced, particularly noticeable at lower values of 2–4 tps, regardless of fps settings. This suggests that these tasks are highly sensitive to the amount of visual information encoded per frame, significantly affecting performance by the number of tokens per frame.

In contrast, Egocentric Understanding and Reasoning tasks show a less severe performance drop when tps is reduced, especially at lower fps values. This implies that these tasks are less sensitive to the number of tokens per frame and are more influenced by the temporal resolution provided by fps, with the ability to capture temporal dynamics being more critical than the density of visual information per frame. The Perception metric behaves as an outlier; apart from an anomalous data point at 1 fps, perception performance tends to favor lower fps values and is less affected by variations in tps. This indicates that for certain perceptual tasks, higher temporal sampling does not necessarily provide additional benefits, and effective performance can be achieved with fewer frames and tokens.

Overall, these results highlight the importance of tailoring video sampling strategies to the specific requirements of different video perception tasks to optimize model performance.



**Figure 15 Scaling Consistency.** Average accuracy for each one for each design variation, we can tell model’s correlation gets progressively better. When comparing two small models (1.5B and 0.5B), we do not see a good correlation, confirming that the Scaling Consistency is not due to the models being of similar size but larger than a certain size.

## F Raw results

We provide the raw evaluations of all the models utilized in our study. Many investigations required multiple experiments to test whether design decisions hold under multiple hyperparameters. We provide all the raw data used in our study for further analysis. For Sec. 3: Tab. 14 & 15, Sec. 4.1: Tab. 10 & 11, Sec. 4.2: Tab 9, Sec. 4.4: Tab. 2, Sec. 5.1: Tab. 12, Sec. 5.3: Tab. 13.

LLM	Vision Encoders	Hyperparameters			ApolloBench				
		OCR	Spatial	Egocentric	Perception	Reasoning	Overall		
1	Qwen2.5-3B-Instruct	DINOv2	36.6	40.5	55.9	48.0	46.3	45.5	
2	Qwen2.5-3B-Instruct	LanguageBind-Image	41.2	46.2	49.5	51.0	51.7	47.9	
3	Qwen2.5-3B-Instruct	SigLIP SO400M	41.9	52.2	57.4	52.0	60.0	52.7	
4	Qwen2.5-3B-Instruct	VideoMAE	35.6	35.5	47.9	47.0	40.0	41.2	
5	Qwen2.5-3B-Instruct	V-JEPA	39.4	35.2	44.1	52.0	44.6	43.1	
6	Qwen2.5-3B-Instruct	LanguageBind-Video	41.2	47.8	54.8	53.2	46.3	48.7	
7	Qwen2.5-3B-Instruct	InternVideo2	43.7	46.5	56.4	55.2	58.1	52.0	
8	Qwen2.5-3B-Instruct	VideoMAE + DINOv2	40.1	43.2	57.4	59.5	47.5	49.6	
9	Qwen2.5-3B-Instruct	VideoMAE + LanguageBind-Image	39.8	49.8	55.9	57.5	49.8	50.5	
10	Qwen2.5-3B-Instruct	VideoMAE + SigLIP SO400M	45.8	54.8	55.9	63.0	55.6	55.0	
11	Qwen2.5-3B-Instruct	V-JEPA + DINOv2	41.5	43.2	56.4	55.2	48.5	49.0	
12	Qwen2.5-3B-Instruct	V-JEPA + LanguageBind-Image	43.3	49.2	50.5	59.2	52.9	51.1	
13	Qwen2.5-3B-Instruct	V-JEPA + SigLIP SO400M	48.6	53.2	59.0	57.8	58.1	55.3	
14	Qwen2.5-3B-Instruct	LanguageBind-Video + DINOv2	41.5	44.9	54.6	57.6	51.0	50.0	
15	Qwen2.5-3B-Instruct	LanguageBind-Video + LanguageBind-Image	41.2	48.5	53.2	62.7	54.7	52.1	
16	Qwen2.5-3B-Instruct	LanguageBind-Video + SigLIP SO400M	45.4	50.5	59.6	56.8	54.9	53.4	
17	Qwen2.5-3B-Instruct	InternVideo2 + DINOv2	43.0	48.2	50.0	58.0	57.1	51.3	
18	Qwen2.5-3B-Instruct	InternVideo2 + LanguageBind-Image	45.8	48.0	51.6	62.3	56.9	52.9	
19	Qwen2.5-3B-Instruct	InternVideo2 + SigLIP SO400M	48.8	56.4	59.9	64.1	64.5	57.9	

**Table 9 Raw results for vision encoders experiment.** The table presents performance scores on ApolloBench at a tokens-per-second (TPS) rate of 32. Metrics include OCR, spatial understanding, egocentric reasoning, perception, reasoning, and overall performance. The encoders are grouped and ordered as follows: single image encoders, single video encoders, and dual encoder configurations.

LLM	Vision Encoders	Hyperparameters			ApolloBench						
		tps	fps	tpf	OCR	Spatial	Egocentric	Perception	Reasoning	Overall	
1	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	512.0	4.0	128.0	46.0	51.0	52.1	59.0	54.0	52.4
2	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	256.0	2.0	128.0	45.5	53.5	51.5	59.0	49.0	51.7
3	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	128.0	1.0	128.0	51.0	55.0	55.3	51.0	62.5	54.9
4	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64.0	0.5	128.0	48.0	52.0	54.2	63.0	56.0	54.6
5	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	256.0	4.0	64.0	43.5	50.0	55.3	62.0	56.0	53.3
6	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	128.0	2.0	64.0	51.0	52.0	61.6	58.0	59.5	56.4
7	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64.0	1.0	64.0	52.5	55.0	60.6	58.5	57.0	56.7
8	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32.0	0.5	64.0	47.5	56.5	60.0	58.0	60.0	56.4
9	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	128.0	4.0	32.0	52.0	57.5	60.6	61.0	57.5	57.7
10	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64.0	2.0	32.0	55.0	58.0	60.6	55.5	62.5	58.3
11	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32.0	1.0	32.0	52.5	54.5	62.7	51.0	63.0	56.7
12	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16.0	0.5	32.0	50.0	56.0	58.4	63.0	58.0	57.1
13	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64.0	4.0	16.0	49.5	60.5	58.4	60.0	62.5	58.2
14	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32.0	2.0	16.0	53.0	56.0	53.1	56.0	59.5	55.6
15	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16.0	1.0	16.0	54.5	58.5	55.3	61.0	61.0	58.1
16	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8.0	0.5	16.0	50.0	50.5	61.1	59.5	55.5	55.3
17	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32.0	4.0	8.0	55.5	59.5	59.0	57.5	61.5	58.6
18	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16.0	2.0	8.0	45.5	55.5	60.0	66.0	62.5	57.9
19	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8.0	1.0	8.0	54.5	55.0	62.7	59.0	58.0	57.8
20	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	4.0	0.5	8.0	50.5	56.0	57.4	61.5	60.0	57.1
21	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16.0	4.0	4.0	29.5	25.0	1.0	38.5	12.5	21.5
22	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8.0	2.0	4.0	35.0	40.5	48.9	52.0	40.0	43.2
23	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	4.0	1.0	4.0	41.5	43.5	52.1	63.0	51.5	50.3
24	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	2.0	0.5	4.0	39.5	47.0	61.6	55.0	50.0	50.5
25	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8.0	4.0	2.0	38.5	36.5	54.2	47.5	44.5	44.1
26	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	4.0	2.0	2.0	26.5	23.5	30.8	44.9	27.5	32.4
27	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	2.0	1.0	2.0	37.3	41.8	53.1	50.3	45.9	44.8
28	Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	1.0	0.5	2.0	41.0	42.0	54.2	45.0	48.2	47.3

**Table 10 Raw results of video sampling experiment.** ApolloBench breaks down metrics to OCR, spatial understanding, egocentric reasoning, perception, reasoning, and overall performance. The table highlights the impact of frames per second (fps), tokens per second (tps), and tokens per frame (tpf).

LLM	Vision Encoders	Hyperparameters		ApolloBench					
		Uniform (Train)	Frames (Test)	OCR	Spatial	Egocentric	Perception	Reasoning	Overall
1 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8	8	38.0	41.0	43.1	50.3	44.0	44.2
2 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16	16	40.5	46.7	55.9	55.3	46.1	48.1
3 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32	32	49.5	52.0	51.1	58.5	48.5	51.9
4 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64	64	46.5	52.0	61.2	56.5	59.5	55.1
5 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	8	No	42.5	44.5	54.8	52.0	51.5	49.0
6 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	16	No	48.0	43.5	58.5	60.5	53.0	52.6
7 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	32	No	46.0	50.0	52.1	57.5	57.5	52.6
8 Qwen2.5-3B-Instruct	InternVideo2-1B + SigLIP SO400M	64	No	48.5	53.5	59.0	54.5	54.0	53.8

**Table 11 Raw results of uniform sampling experiment.** ApolloBench evaluates metrics including OCR, spatial understanding, egocentric reasoning, perception, reasoning, and overall performance. Top half are results when models are both trained and tested with uniform frame sampling. The bottom half is when the models are trained with uniform frame sampling but tested at an fps of 2.

	Training Stages			ApolloBench					
	1	2	3	OCR	Spatial	Egocentric	Perception	Reasoning	Overall
1 $0, 1e^{-4}, 3e^{-5}$	-	-	-	42.0	46.5	54.9	50.0	49.5	48.7
2 $1e^{-6}, 1e^{-4}, 3e^{-5}$	-	-	-	28.8	29.2	18.8	35.5	22.6	30.8
3 $5e^{-6}, 1e^{-4}, 3e^{-5}$	-	-	-	26.8	23.2	12.1	21.4	24.5	22.2
4 $1e^{-5}, 1e^{-4}, 3e^{-5}$	-	-	-	24.9	16.1	26.4	39.9	18.4	25.0
5 $0, 1e^{-4}, 0,$	$0, 1e^{-4}, 3e^{-5}$	-	-	52.2	54.5	55.9	60.3	58.4	56.3
6 $1e^{-6}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	-	49.6	54.2	61.4	63.3	59.5	57.6
7 $5e^{-6}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	-	51.6	54.5	58.0	62.1	60.2	57.8
8 $1e^{-5}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	-	50.0	50.0	44.5	55.3	47.6	49.7
9 $1e^{-6}, 1e^{-4}, 0$	$1e^{-6}, 1e^{-4}, 3e^{-5}$	-	-	42.2	48.9	61.7	43.7	52.2	48.1
10 $5e^{-6}, 1e^{-4}, 0$	$5e^{-6}, 1e^{-4}, 3e^{-5}$	-	-	32.2	37.5	50.0	40.4	44.2	40.3
11 $1e^{-5}, 1e^{-4}, 0$	$1e^{-5}, 1e^{-4}, 3e^{-5}$	-	-	30.3	23.8	49.5	41.9	30.3	33.7
12 $0, 1e^{-4}, 0,$	$1e^{-6}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	46.7	50.7	60.6	57.6	61.8	55.4
13 $0, 1e^{-4}, 0,$	$5e^{-6}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	52.4	55.4	62.8	63.5	61.4	59.2
14 $0, 1e^{-4}, 0,$	$1e^{-5}, 1e^{-4}, 0$	$0, 1e^{-4}, 3e^{-5}$	-	53.7	54.0	47.3	56.2	52.9	53.2
15 $0, 1e^{-4}, 0,$	$1e^{-6}, 1e^{-4}, 0$	$1e^{-6}, 1e^{-4}, 3e^{-5}$	-	44.2	37.5	43.9	56.6	38.5	44.2
16 $0, 1e^{-4}, 0,$	$5e^{-6}, 1e^{-4}, 0$	$5e^{-5}, 1e^{-4}, 3e^{-5}$	-	32.7	36.9	49.8	40.1	44.5	39.8
17 $0, 1e^{-4}, 0,$	$1e^{-5}, 1e^{-4}, 0$	$1e^{-5}, 1e^{-4}, 3e^{-5}$	-	32.4	36.6	30.1	42.3	33.5	35.4

**Table 12 Raw results of training schedules experiments.** Results of training models across 1, 2, and 3 stages with varying learning rates (LR) and data mixtures. The table highlights OCR, spatial understanding, egocentric reasoning, perception, reasoning, and overall performance metrics. Each stage utilizes different LR configurations and data distributions, showing the benefits of multi-stage training for optimizing performance across all metrics (see Tab. 7 and Sec. C.3 for details).

	Data Composition				ApolloBench					
	Text	Image	Multi-Image	Video	OCR	Spatial	Egocentric	Perception	Reasoning	Overall
1	25.0	25.0	25.0	25.0	41.0	49.5	59.0	57.0	59.5	54.1
2	15.0	25.0	20.0	40.0	47.5	59.0	60.6	66.0	62.0	59.0
3	15.0	32.5	20.0	32.5	46.5	52.0	58.0	65.5	63.5	57.1
4	15.0	40.0	20.0	25.0	45.0	57.3	52.1	60.2	61.1	56.2
5	7.0	38.7	20.0	34.3	44.5	53.0	54.3	58.0	55.5	53.0
6	7.0	55.0	20.0	18.0	39.5	45.0	46.7	56.0	54.0	48.3
7	7.0	18.0	48.0	27.0	40.2	48.0	53.2	57.2	53.5	50.9
8	7.0	0.0	0.0	93.0	37.5	33.5	52.7	40.5	45.5	41.8
9	7.0	0.0	20.0	73.0	37.0	44.0	51.1	45.0	49.0	45.1
10	7.0	14.0	18.0	61.0	41.0	48.5	54.2	56.8	54.5	51.2
11	5.0	10.0	40.0	45.0	40.0	47.5	53.2	57.3	51.7	50.4
12	2.0	30.0	30.0	38.0	35.5	47.0	55.0	56.0	49.5	48.7
13	0.0	38.7	20.0	41.3	35.4	44.1	54.1	54.2	49.0	47.5

**Table 13 Raw results of data composition experiments.** Performance outcomes of video-based Large Multi-modal Models (LMMs) trained with varying proportions of Text, Image, Multi-Image, and Video data mixtures. The table presents benchmark scores across OCR, Spatial, Egocentric, Perception, Reasoning, and Overall performance metrics for each distinct data composition. These results emphasize the critical role of balanced data mixtures in optimizing model performance (see Sec. 5.3 for details).

	LLM	Vision Towers	Vision Freeze	Clip Duration	Tokens /Clip	fps	tps	Tokens /Frame	Data Mixture	Average
1	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	A	46.37
2	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	A	46.46
3	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	A	48.79
4	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	A	47.22
5	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	A	43.46
6	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	A	46.47
7	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	A	42.11
8	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	B	49.75
9	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	B	50.61
10	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	B	50.91
11	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	B	49.44
12	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	B	50.14
13	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	B	51.08
14	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	B	43.92
15	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	C	51.80
16	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	C	52.91
17	Qwen2-7B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	C	52.07
18	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	C	51.36
19	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	C	52.49
20	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	C	53.13
21	Qwen2-7B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	C	44.73
22	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	A	42.67
23	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	A	42.92
24	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	A	44.85
25	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	A	43.25
26	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	A	41.91
27	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	A	42.83
28	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	A	39.91
29	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	B	45.90
30	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	B	46.75
31	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	B	46.76
32	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	B	46.53
33	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	B	45.56
34	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	B	46.16
35	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	B	39.63
36	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	C	48.34
37	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	C	48.29
38	Qwen1.5-4B-Chat	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	C	47.32
39	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	C	48.45
40	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	C	47.24
41	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	C	48.24
42	Qwen1.5-4B-Chat	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	C	40.76

**Table 14 Raw results of Scaling Consistency experiments (1/2).** This table presents the raw performance data of 42 model configurations used in the Scaling Consistency experiments. Each configuration explores the effect of various parameters, including the LLM size (Qwen variants), vision tower configurations, freezing or training vision encoders, clip duration, tokens per clip, frames per second (fps), tokens per second (tps), tokens per frame, and data mixture. The “Average” column reports the overall performance score. These results support the investigation into how smaller models can serve as proxies for larger models in determining effective design decisions.

	LLM	Vision Towers	Vision Freeze	Clip Duration	Tokens /Clip	fps	tps	Tokens /Frame	Data Mixture	Average
43	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	A	42.87
44	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	A	42.94
45	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	A	44.00
46	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	A	40.77
47	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	A	42.13
48	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	A	41.93
49	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	A	40.18
50	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	B	45.98
51	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	B	46.06
52	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	B	45.73
53	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	B	46.02
54	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	B	44.46
55	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	B	44.78
56	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	B	41.47
57	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	C	46.96
58	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	C	47.66
59	Qwen2-1.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	C	46.43
60	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	C	47.65
61	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	C	45.94
62	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	C	46.31
63	Qwen2-1.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	C	41.76
64	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	A	36.50
65	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	A	35.75
66	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	A	36.43
67	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	A	36.27
68	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	A	37.21
69	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	A	36.80
70	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	A	34.64
71	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	B	39.29
72	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	B	39.59
73	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	B	37.36
74	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	B	40.25
75	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	B	39.74
76	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	B	40.01
77	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	B	35.05
78	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	32	1.6	6.4	4	C	40.19
79	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	❄️	5	64	1.6	12.8	8	C	40.25
80	Qwen2-0.5B-Instruct	LanguageBind-Video-v1.5 + SigLIP SO400M	🔥	5	64	1.6	12.8	8	C	37.38
81	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	32	3.2	6.4	2	C	40.76
82	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	10	64	1.6	6.4	4	C	40.48
83	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	❄️	5	64	3.2	12.8	2	C	39.99
84	Qwen2-0.5B-Instruct	VJEPAP-H@384 + SigLIP SO400M	🔥	5	64	3.2	12.8	2	C	35.33

**Table 15 Raw results of Scaling Consistency experiments (2/2).** This table presents the raw performance data of 42 model configurations used in the Scaling Consistency experiments. Each configuration explores the effect of various parameters, including the LLM size (Qwen variants), vision tower configurations, freezing or training vision encoders, clip duration, tokens per clip, frames per second (fps), tokens per second (tps), tokens per frame, and data mixture. The “Average” column reports the overall performance score. These results support the investigation into how smaller models can serve as proxies for larger models in determining effective design decisions.