Supplementary Materials
POINTS-Long: Adaptive Dual-Mode Visual Reasoning in MLLMs

Haicheng Wang[1,2*], Yuan Liu[2*✉], Yikun Liu[1,2*], Zhemeng Yu[1], Zhongyin Zhao[2],
Yangxiu You[2], Zilin Yu[2], Le Tian[2], Xiao Zhou[2], Jie Zhou[2], Weidi Xie[1], Yanfeng Wang[1✉]
[1] SAI, Shanghai Jiao Tong University, China [2] WeChat AI, Tencent, China

In this supplementary material, we first provide a comprehensive description of our base model, POINTS1.5-8B-Instruct. Subsequently, we elaborate on the architectural details and training protocols of POINTS-Long. Finally, we present additional ablation studies and visualizations.

1. Details about POINTS1.5-8B-Instruct

1.1. Model Architecture

The POINTS [26] series is a family of advanced multimodal large language models (MLLMs) that was first released in September 2024. The POINTS1.5-8B-Instruct model employed in this work is an enhanced iteration of POINTS1.5 [26] (Fig. 1). It is initialized from Qwen3-8B-Base [49] and Qwen2-VL-ViT [43]. The model applies 1D RoPE [39] for visual tokens within the LLM backbone and 2D RoPE within the ViT image encoder. Furthermore, the intermediate projector utilizes a pixel-shuffle operation to reduce the visual sequence length by a factor of 4.

1.2. Model Training Dataset

POINTS1.5-8B-Instruct underwent comprehensive multimodal training, organized into four distinct stages:

Visual-textual Alignment In this initial phase, the parameters of both the Vision Transformer (ViT) and the Large Language Model (LLM) remained frozen, with training optimized solely on the alignment projector. We utilized Laion-5B [34] as seed data, which was subsequently processed using CapFusion [52] for recaptioning and perplexity filtering [27] for quality control. For this stage, we employed a sequence length of 8192.

Multimodal Continue Pre-training To construct our image-text pre-training dataset, we sourced raw PDFs from the CC-MAIN-2021-31-PDF-UNTRUNCATED[1] dataset, retaining only Chinese and English documents. Our processing pipeline utilized PaddleOCR [9] for image extraction and the POINTS-Reader [28] document OCR model for text extraction. For each document, we concatenated the extracted images (placed at the beginning) with the corresponding text and page format. This process yielded a pre-training corpus containing approximately 400 billion tokens. Analogous to LLM pre-training, this stage utilizes massive unlabeled web data to expose the model to broad world knowledge.

Multimodal Decay Following pre-training, we initiated a Decay stage designed to bolster the MLLM's performance across a spectrum of capabilities, including grounding, OCR, GUI navigation, reasoning, video understanding, and text-based CoT.

To achieve this, we constructed specialized training data from diverse sources, including open-source datasets such as Wukong [15], Object365 [35], and Koala-36M [44], alongside proprietary in-house data. Training in this stage was conducted in two steps: first, we focused on fine-grained image understanding with a context length of 8k. Subsequently, we expanded the context length to 32k, incorporating data for complex video understanding tasks (e.g., dense captioning and temporal grounding) and long-context, text-only CoT data.

Multimodal Supervised Instruction Tuning The Multimodal Supervised Fine-Tuning (SFT) stage is designed to utilize high-quality data to teach the model to follow instructions and align with human preferences.

In this phase, we utilize a large volume of high-quality image-text and video QA data. For the video domain, in addition to proprietary in-house data, we primarily leverage open-source datasets, including FineVideo [12], Vript [50], ShareGPT4Video [6], OpenVid-1M [32], VideoUFO [45], CinePile [33], VideoChat2IT [21], LLaVA-Hound [56], LLaVA-Video-178K [57], and Ego4D [14]. We conduct training with a 32K context length in this stage. For video preprocessing, we split ultra-long videos into shorter segments and sample frames at 1 fps. Due to sequence length constraints, we set the maximum frame limit to 128; videos exceeding this limit are uniformly downsampled on temporal dimension.

Multimodal Post-training We apply RFT (Rejection Sampling Fine-Tuning) and RL (Reinforcement Learning) to en-

[1] https://digitalcorpora.org/corpora/file-corpora/cc-main-2021-31-pdf-untruncated/

hance the model's reasoning and cognitive capabilities. For RFT, we utilize open-source synthetic reasoning datasets such as AM-DeepSeek-R1-Distilled-1.4M [58], Reason-RFT [40], and VisualWebInstruct [19], covering a wide range of disciplines. For RL, we train on a diverse range of tasks, including STEM (e.g., mathematics, physics, chemistry), puzzle solving, and OCR-based reasoning.

1.3. Training Recipe

We conduct training using our in-house framework, which is analogous to Megatron [37]. We set the Tensor Parallel (TP) degree to 2 during the Alignment stage and 4 for all subsequent stages. During the long-context training phase, we enable sequence parallelism and utilize activation checkpointing to minimize memory overhead. We employ a "pack-to-pack" (or sample packing) training strategy, utilizing a learning rate of 3e-4 for the Alignment stage and 5e-5 for all other phases.

1.4. Chat Template

We adhere to the standard chat template of Qwen2.5-VL [1], with the primary distinction lying in the representation of video inputs. Instead of treating the video as a monolithic entity, we enclose each input frame within `<|vision_start|> <|vision_end|>` tags. To enable the model to explicitly perceive temporal information, we prepend a metadata string to the video input: `Video of x fps:`. This prefix identifies the modality and specifies the framerate. Furthermore, we interleave textual timestamps between video frames. To conserve token usage, these timestamps are inserted directly as numerical values representing seconds (e.g., `1<frame1>2.5<frame2>4<frame3>`).

2. Details about POINTS-Long

In the supplementary material, we provide more details about POINTS-Long.

2.1. POINTS-Long Architecture

POINTS-Long is built upon the POINTS1.5-8B-Instruct architecture. As illustrated in Fig. 2 of the main paper, the primary modification involves the vision backbone, where n additional learnable tokens—termed "standby tokens"—are concatenated with the original patchified visual sequence. Within each layer, duplicated MLPs are introduced to process these standby tokens independently. Furthermore, a temporal modeling attention block is inserted into the final 5 layers of the ViT to encode standby tokens across 8 adjacent frames. Crucially, the attention mechanism in this temporal block is causal, enabling efficient processing of streaming inputs without the need for re-computation. Unlike full attention, which necessitates simultaneous access to a window of 8 frames during the forward pass—an approach ill-suited for frame-by-frame streaming—causal attention allows the model to simply cache the standby representations of the preceding 7 frames. This results in negligible memory overhead while significantly enhancing the model's capability to handle streaming scenarios.

To maintain architectural consistency, we apply the same duplication strategy to the projection layer. It is important to clarify the notation regarding n: in our experimental tables, the reported token count refers to the final input tokens to the LLM. Since the projection layer employs a pixel-shuffle operation that aggregates 4 neighboring tokens into 1, the number of learnable standby tokens initialized in the ViT is 4 times the final token count in the LLM. For instance, in Tab. 1 in the main paper, a "Num/Frame" of 8 corresponds to an initialization of $n = 32$ standby tokens in the vision backbone.

Here we express this encoding process in a mathematical way. The input image I_q is transformed into a patchified visual sequence with o as sequence length: $Z_{q0} = \{z_{q01}, ..., z_{q0o}\} \in \mathbb{R}^{o \times d}$ by patch embedding layer (the 3 subscripts represent frame index, layer index, and sequence index, respectively). We initialize n learnable tokens $L_{q0} = \{l_{q01}, ..., l_{q0n}\} \in \mathbb{R}^{n \times d}$ and prepend them to the original sequence $\{L_{q0}, Z_{q0}\} = \{l_{q01}, ...l_{q0n}, z_{q01}, ..., z_{q0o}\}$, where normally $o \gg n$. The two parallel sequences share the same attention block:

$$\{L'_{qi}, Z'_{qi}\} = \text{Attention\_Block}_i(\{L_{qi}, Z_{qi}\}), \quad (1)$$

where i is the layer/block index. The resultant sequences are processed by different MLPs:

$$\{L_{q(i+1)}, Z_{q(i+1)}\} = \{\text{MLP}_{Li}(L'_{qi}), \text{MLP}_{Zi}(Z'_{qi})\}. \quad (2)$$

Note that the parameter of MLP_{Li} is initialized by MLP_{Zi}. In the last 5 blocks, we add one temporal attention between attention and MLP, taking only the learnable tokens of the adjacent 8 frames:

$$\begin{aligned} &\{L''_{(q-w)i}, ..., L''_{qi}, ..., L''_{(q+v)i}\} = \\ &\text{Attenion\_T}_i(\{L'_{(q-w)i}, ..., L'_{qi}, ..., L'_{(q+v)i}\}), \end{aligned} \quad (3)$$

where $w + v \leq 8$, depending on the position of current input image/frame I_q. For image understanding, the input is L'_{qi} only, and for video inputs, we group the neighboring 8 frames without overlap. Since we use pack-to-pack parallel computing technique, the temporal attention only needs to be calculated once per 8 frames. With temporal modeling, the subsequent MLP layer becomes:

$$\{L_{q(i+1)}, Z_{q(i+1)}\} = \{\text{MLP}_{Li}(L''_{qi}), \text{MLP}_{Zi}(Z'_{qi})\}. \quad (4)$$

For the projection layer, we also apply the same parallel encoding strategy. We note $\{z_1, ..., z_o\}_q$ the resultant original visual sequence for each image q and $\{l_1, ..., l_n\}_q$ the

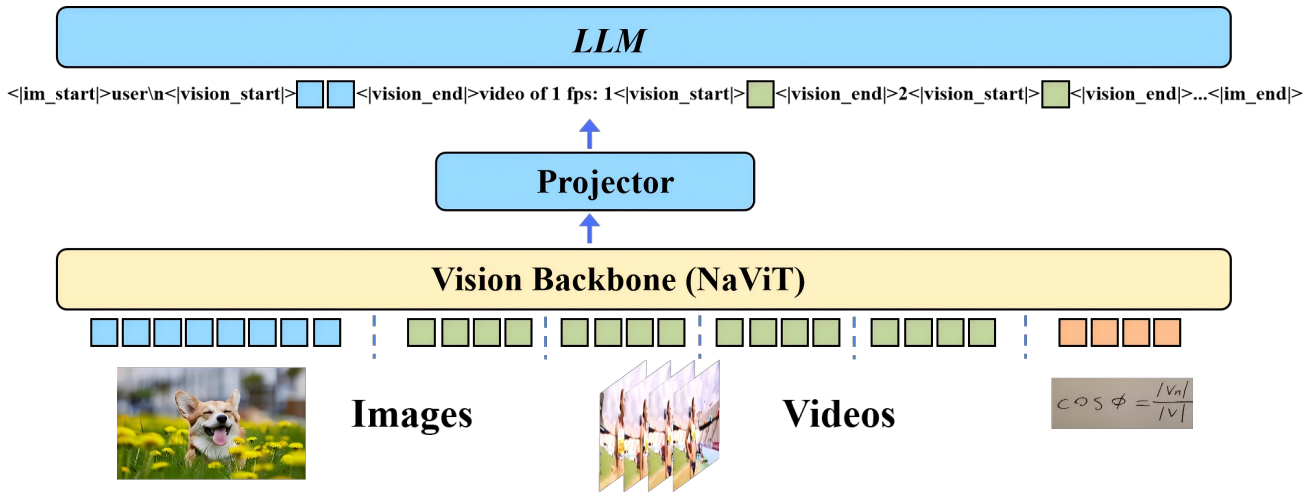

Figure 1. **POINTS1.5-8B-Instruct Architecture.** POINTS1.5-8B consists of a native-resolution image encoder (initialized from Qwen2-VL-ViT), a pixel-shuffle projector reducing the token count by a factor of 4, and an LLM initialized from Qwen3-8B-Base. The architecture employs 1D RoPE for the LLM and 2D RoPE for the ViT.

learnable standby tokens, after being encoded by ViT and projector. During the two-stage training process, we activates different modes. In stage 1, we only pass the learnable standby tokens to LLM:

$$\text{Loss} = LLM(\{l_1, ..., l_n\}_q \forall q \in Q, \text{Text}), \tag{5}$$

where Q is the image/frame set in the sample. In stage 2, we apply the 2-forward training strategy:

$$\begin{aligned} \text{Loss}_1 &= LLM(\{l_1, ..., l_n\}_q \forall q \in Q, \text{Text}), \\ \text{Loss}_2 &= LLM(\{l_1, ..., l_n, z_1, ..., z_o\}_q \forall q \in Q, \text{Text}), \\ \text{Loss} &= \frac{1}{2}(\text{Loss}_1 + \text{Loss}_2) \end{aligned} \tag{6}$$

2.2. Training Dataset

The training of POINTS-Long is conducted in two distinct stages.

Stage 1: Visual Distillation and Alignment. In this phase, all parameters of the original architecture—including the LLM backbone—remain frozen. Optimization is restricted exclusively to the newly introduced learnable tokens, the duplicated MLPs, and the projection layer. The objective is to enable these "standby tokens" to effectively aggregate and distill visual information from the original sequence, a process analogous to the alignment phase in MLLM training. To achieve this, we utilize the complete alignment dataset alongside a subset of data from the multimodal decay stage (detailed in Sec. 1.2) to ensure robust visual distillation. Since all parameters governing the original inference path remain frozen, the model's baseline performance remains strictly preserved during this stage.

Stage 2: LLM Mode Adaptation. In the second stage, we fine-tune the LLM using a reduced learning rate. We employ a dual-path forward strategy: computing the average loss derived from both Standby and Focus forward passes before backpropagating the gradients. This mechanism enables the LLM to adapt simultaneously to both inference modes. For this stage, we incorporate a high-quality subset of the multimodal decay data alongside the full Supervised Fine-Tuning (SFT) dataset. Notably, all training data employed across both stages is derived exclusively from the training set of the baseline model, POINTS1.5-8B. No external data is introduced, thereby ensuring a fair comparison.

2.3. Dual-Mode Inference

Here, we detail the inference protocols for standby mode and focus mode. When operating in standby mode, we feed only the compressed, short learnable token sequence to the LLM as visual input for inference. Conversely, in focus mode, the entire sequence—comprising both the learnable tokens and the original visual tokens—is passed to the LLM. Formally, for focus mode, we pass $\{l_1, ..., l_n, z_1, ..., z_o\}_q$ to LLM. While for standby mode, we only pass $\{l_1, ..., l_n\}_q$. Formally, we can express the inference of the two modes as follows:

$$\begin{aligned} &\text{Standby} : \text{Output} = \text{LLM}(\{l_1, ..., l_n\}_q, \text{Text}), \\ &\text{Focus} : \text{Output} = \text{LLM}(\{l_1, ..., l_n, z_1, ..., z_o\}_q, \text{Text}) \end{aligned} \tag{7}$$

In practice, we could use only the original visual sequence (without the learnable tokens) for inference. However, including the learnable tokens provides a significant advantage for streaming visual inference: we can leverage

Table 1. **Performance of Different Inference Mode.** In standard focus mode, we concatenate the learnable standby tokens with the original visual tokens and pass to LLM. Nevertheless, it makes no big difference to inference with only the original visual tokens. *ori-seq means original sequence without standby tokens.

Model	MMBench	MMStar	MMMU_val	MathVista	OCRBench	AI2D	HallusionBench	MMVet	Avg
POINTS1.5-8B (baseline)	81.9	65.7	53.2	70.9	85.8	83.9	50.1	64.7	69.5
POINTS-Long (focus)	82.1	66.1	53.7	70.6	85.5	84.2	48.3	66.7	69.7
POINTS-Long (ori-seq)	82.1	65.8	53.0	69.7	85.0	83.8	48.0	67.4	69.4

Table 2. **Learning Rate & Model Performance.** We train the model under different learning rates (1e-5, 2e-5, 5e-5) in stage 2. Performance differences were minimal, proving the training scheme's robustness.

Model	Num Frame	Token/ Frame	Learning Rate	MVBench	Video-MME	MMBench-Video	Tempcompass	MLVU	LongVideoBench	Avg
POINTS-Long (standby)	64	16	1e-5	59.7	65.0	59.3	69.1	71.7	58.9	**63.9**
POINTS-Long (standby)	64	16	2e-5	58.3	64.9	59.3	69.3	71.8	58.2	63.6
POINTS-Long (standby)	64	16	5e-5	59.7	64.9	60.0	69.4	70.3	58.4	63.8
POINTS-Long (standby)	64	32	1e-5	60.8	65.7	60.9	70.3	71.6	59.5	**64.8**
POINTS-Long (focus)	64	32	1e-5	61.0	66.1	60.3	71.3	73.2	59.4	65.2
POINTS-Long (standby)	64	32	2e-5	61.3	65.9	60.3	70.6	70.8	59.5	64.7
POINTS-Long (focus)	64	32	2e-5	62.1	66.1	61.3	71.4	72.5	58.8	**65.4**
POINTS-Long (standby)	64	32	5e-5	58.8	65.8	61.3	71.3	71.1	59.4	64.6
POINTS-Long (focus)	64	32	5e-5	61.2	67.0	61.3	71.1	72.2	59.5	65.4

the "detachable KV cache" technique (described in the main paper Sec. 3.4) and avoid re-computation. Given that the learnable token sequence is significantly shorter than the original sequence, this method has a negligible impact on accuracy and computational overhead, as we show in Tab. 1.

2.4. Training-free Token Pruning

Following the two-stage training, the standby tokens have effectively absorbed critical visual information from the original sequence. Previous research has indicated that the attention distribution of learnable global representation tokens correlates strongly with the most salient information. This implies that we can leverage the attention distribution of the standby tokens relative to other tokens to identify the most important visual tokens within the long sequence.

This enables us to perform a training-free pruning of visual tokens based directly on this distribution. Specifically, within the final layer of the Vision Transformer (ViT), we calculate the mean attention score for each token in the original visual sequence relative to the standby tokens. These scores are then sorted, and we retain only the top m% of tokens to be fed into the LLM.

It is important to note that the pixel-shuffle operation performs a projection on adjacent groups of four tokens. To avoid disrupting this projection, our compression method treats these four-token groups as atomic units. We calculate a group attention score (by averaging the attention of the four constituent tokens), ensuring that we prune at the granularity of these post-pixel-shuffle units. As demonstrated in Tab. 4 of the main paper, this straightforward approach outperforms other token compression methods.

3. Details on Evaluation

In this section, we explain in detail our evaluation metric.

3.1. Evaluation Benchmark

Fine-grained Image Benchmarks We leverage Opencompass [8] image benchmark for evaluation, including MMBench [23], MathVista [30], HallusionBench [16], OCRBench [25], AI2D [20], MMVet [53], MMStar [5], MMMU [54]. Note that MMMU is evaluated on validation set, MMBench is the average of MMBench_test_EN and MMBench_test_CN. We use VLMEvalKit [10] for all the image evaluation.

Video Benchmarks We evaluate on a wide range of video benchmarks, including Opencompass video leaderboard: VideoMME [13], Tempcompass [24], MVBench [22], MMBench-Video [11], MLVU [59], LongVideoBench [48], and other commonly used video benchmarks: MovieChat1K [38], CG-Bench [4], EgoSchema [31], TemporalBench [3], Activitynet-qa [2], LVBench [46] and WorldSense [17]. We use lmms-eval [55] to evaluate LVBench, WorldSense, EgoSchema, TemporalBench and Activitynet-qa, while for the rest we use VLMEvalKit [10]. Note that we evaluate CG-Bench on its long accuracy and MLVU on M-Avg.

3.2. Streaming Video Evaluation

Streaming video understanding is an increasingly critical application for large models. Our POINTS-Long model is specifically designed and optimized for this scenario, achieving a long-term visual memory bank by leveraging a detachable KV cache and dual-mode cooperation.

Our evaluation methodology mimics a real-world streaming scenario. For the baseline model's inference, we

Table 3. **Training Data & Model Performance.** We train the model using different amount of data in stage 2. By adding more high-quality data in stage 2 (85%-100%), we witness a steady improvement in performance.

Model	Num Frame	Token/ Frame	Training Data	MVBench	Video-MME	MMBench-Video	Tempcompass	MLVU	LongVideoBench	Avg
POINTS-Long (standby)	64	16	Standard	59.7	65.0	59.3	69.1	71.7	58.9	**63.9**
POINTS-Long (standby)	64	16	Reduced	59.3	64.0	59.0	68.9	71.3	59.2	63.6
POINTS-Long (standby)	64	32	Standard	60.8	65.7	60.9	70.3	71.6	59.5	**64.8**
POINTS-Long (standby)	64	32	Reduced	60.7	65.1	59.3	69.9	70.8	60.9	64.3

Table 4. **Comparison with Visual Token Reduction Methods.** Under the same setting (or even using fewer tokens), POINTS-Long exceeds previous visual token reduction methods by a large margin. It's a natural result since the standby mode is carefully trained as a native inference mode.

Model	Num Frame	Token/ Frame	Total Num of Token	MVBench	Video-MME	MMBench-Video	Tempcompass	MLVU	LongVideoBench	Avg
POINTS1.5-8B (baseline)	64	324	$\approx$ 20K	60.3	66.1	61.0	71.1	72.0	59.8	65.0
POINTS1.5-8B (low-resolution)	64	32	2048 (10%)	54.9	61.2	51.0	67.1	67.3	53.9	59.2 (91.1%)
POINTS1.5-8B (pooling)	64	32	2048 (10%)	54.9	55.4	43.0	66.6	67.1	54.5	56.9 (87.5%)
POINTS1.5-8B (+VisionZip [51])	64	32	2048 (10%)	56.7	60.0	51.7	66.6	65.9	54.9	59.3 (91.2%)
POINTS1.5-8B (+Dycoke [41])	64	32	2048 (10%)	56.1	62.5	55.7	67.5	67.2	55.6	60.8 (93.5%)
POINTS1.5-8B (+PruneVID [18])	64	32	2048 (10%)	57.8	62.0	55.3	69.3	67.5	56.5	61.4 (94.4%)
POINTS1.5-8B (+FastVID [36])	64	32	2048 (10%)	54.9	63.9	56.0	68.3	70.1	54.5	62.7 (96.5%)
POINTS-Long (standby)	64	8	512 (2.5%)	59.4	63.5	58.0	69.9	71.9	58.2	63.5 (97.7%)
POINTS-Long (standby)	64	16	1024 (5%)	59.7	65.0	59.3	69.1	71.7	58.9	63.9 (98.3%)
POINTS-Long (standby)	64	32	2048 (10%)	60.8	65.7	60.9	70.3	71.6	59.5	64.8 (99.7%)

uniformly sample 256 frames for prefilling. Once its 64-frame context limit is exceeded, the preceding frame's KV cache is discarded.

For POINTS-Long, we uniformly sample either 256 or 512 frames. The most recent 8 frames are prefilled using focus mode. As soon as this 8-frame local window limit is surpassed, we follow the procedure illustrated in Fig. 3 of the main paper: the standby tokens of previous frames about to be discarded are detached and integrated into the memory bank, while the rest are dropped. While the precise system implementation for this detachment is complex, it can be simplified by just re-prefilling the standby tokens. This alternative method yields nearly identical results with a negligible increase in computational overhead.

It is important to note that while POINTS-Long substantially extends the visual memory capacity, high-FPS long videos may still surpass the context length limit. For this unavoidable forgetting, we recommend employing an external database, such as M3-agent [29]. Since this component is outside the scope of the core POINTS-Long solution, we do not detail a specific implementation. Our evaluation employs a fixed number of frames specifically to measure the performance gains within the POINTS-Long memory capacity; performance on content exceeding this range is expected to be no different from the baseline model.

3.3. Efficiency Benchmarking

In Sec. 4.3.6 of the main paper, we provide a detailed analysis of the advantages of POINTS-Long for industrial-grade deployment. We emphasize that POINTS-Long can significantly accelerate inference in two key ways:

Substantial Reduction in LLM Prefill Time: POINTS-Long significantly reduce the visual sequence length, thus speed up the LLM prefill phase. Benchmarks using SGLang measured a 10-20x decrease in LLM prefill latency.

Increased Decode Throughput: During the LLM decode stage, the historical visual sequence is drastically shortened. This allows us to parallelize significantly more decode requests under the same KV cache budget. Because decoding is an I/O-intensive operation, the number of parallel requests is almost directly proportional to the throughput. Even with our relatively naive implementation, we achieved a 6.2x increase in generation throughput.

For our benchmarks, we used identical samples (from VideoMME) and precisely measured LLM prefill latency using SGLang and the PyTorch profiler. To test throughput, we optimized SGLang's asynchronous visual input CPU preprocessing by using multiprocessing for frame handling, thereby increasing request parallelism. With mem-fraction-static=0.65, the baseline model using 256 frames could only decode approximately 8 requests in parallel. In contrast, POINTS-Long was able to decode over 70 requests in parallel. (It is worth noting that this number was constrained by our system's CPU performance and machine bandwidth, suggesting that the optimal parallel capacity can be higher.)

4. More Experiment

4.1. Ablation

In addition to the ablation study on the parameter module presented in the main paper, we conduct supplementary experiments regarding training data size and learning rate.

Ablation on Learning Rate As shown in Tab. 2, we evaluated the model performance using varying learning rates (1e-5, 2e-5, 5e-5) on LLM at stage 2. The results on video benchmarks indicate minimal performance variance across

Table 5. **Model Soup Performance.** We apply model soup (model merge) to two models trained by different learning rates. The model's performance can further boost in this way.

Model	Num Frame	Token/ Frame	Total Num of Token	CG-Bench	Video-MME	MMBench-Video	Tempcompass	MLVU	LongVideoBench	Avg
POINTS1.5-8B (baseline)	64	324	≈ 20K	36.7	66.1	61.0	71.1	72.0	59.8	61.1
POINTS-Long (standby)	64	16	1024 (5%)	34.6	65.0	59.3	69.1	71.7	58.9	59.8
POINTS-Long (model soup)	64	16	1024 (5%)	35.2	65.9	60.3	69.9	71.6	58.1	60.2 (+0.4)
POINTS-Long (standby)	128	16	2048 (10%)	36.2	66.4	61.0	69.6	72.7	60.3	61.0
POINTS-Long (model soup)	128	16	2048 (10%)	36.7	66.7	61.3	70.4	72.0	60.7	61.3 (+0.3)
POINTS-Long (standby)	64	32	2048 (10%)	35.7	65.7	60.9	70.3	71.6	59.5	60.6
POINTS-Long (model soup)	64	32	2048 (10%)	36.5	65.8	61.0	70.9	72.1	59.8	61.0 (+0.4)
POINTS-Long (standby)	128	32	4096 (20%)	37.3	66.9	62.0	70.1	72.5	60.4	61.5
POINTS-Long (model soup)	128	32	4096 (20%)	37.5	66.6	63.3	70.8	73.8	61.2	62.2 (+0.7)

Standby and Focus inference modes, thereby validating the robustness of our two-stage training strategy. Consequently, to preserve the model's general capabilities and minimize weight shifts, we use the smaller learning rate for stage 2.

Ablation on Training Data In Tab. 3, we demonstrate the effect of data scaling during Stage 2. Increasing the amount of high-quality image-text and video data yields consistent performance improvements. This validates the criticality of data scale and indicates promising scalability towards larger architectures and more extensive datasets.

4.2. Comparison with Visual Reduction Methods

Recent works have extensively explored visual token compression, particularly for video understanding [7, 18, 36, 41, 42, 51]. While most existing approaches are training-free—offering high compatibility—they suffer from severe performance degradation at high compression ratios (as discussed in our Introduction). POINTS-Long addresses this bottleneck via native training, effectively embedding the high-compression 'Standby' mode as a native inference mechanism. As shown in Table 6, this strategy allows POINTS-Long to significantly outperform previous methods at the same compression ratio (99.7% vs. 96.5%). Remarkably, even with 4 times fewer tokens, our model still achieves superior performance (97.7%). This native training paradigm maximizes model potential and represents the future architectural direction for MLLMs. Note that for all comparison methods, we re-implement on POINTS1.5-8B-Instruct, using only their optimization before LLM.

4.3. Model Soup Enhancement

In POINTS1.5 [26], we employ the Model Soup technique [47] to enhance performance. Model Soup involves averaging the weights of multiple fine-tuned models—often trained with different hyperparameters or data—to improve generalization without incurring additional inference costs. specifically, we performed simple parameter averaging on two model checkpoints trained with distinct learning rates. We observed a consistent and notable performance gain across benchmarks (ranging from +0.3 to +0.7). This indicates that the model has not yet reached its performance upper bound and has further capacity for optimization.

5. Visualization

We visualize the position encoding mentioned in Sec. 3.3.1. For the newly introduced learnable standby tokens, we assign positional embeddings by uniformly sampling the RoPE encodings from the original sequence. In Fig. 2, we visualize the attention distribution of these standby tokens towards other visual patches in the final layer of the ViT. We observe a distinct positional clustering effect (or spatial locality bias), where standby tokens tend to aggregate information from spatially adjacent tokens. This behavior aligns perfectly with our design expectations.

6. Failure Case Analysis

We conduct a qualitative analysis on Video-MME in Fig. 3, comparing Baseline (64 frames) with Standby mode (128 frames). As shown in the figure below, many Standby failures ($> 50\%$) are caused by deficits in spatial or fine-grained perception, whereas the Baseline fails more often on temporal and general understanding.

7. Limitation & Future Work

In this work, we provided a comprehensive analysis of training dual-mode MLLMs and validated their effectiveness across both offline and streaming scenarios. However, the full potential of this dual-mode architecture remains under-explored. For instance, future training strategies could involve interleaved mode switching or utilizing the Standby mode to scale up the number of training frames. Ideally, the model should autonomously determine the appropriate inference mode via post-training strategies, potentially achieving frame-level precision.

Consider a long video understanding scenario: the model could first ingest densely sampled frames in Standby mode, then dynamically select keyframes to examine in Focus mode based on the specific query. This concept of 'thinking with videos' mirrors human cognitive patterns: skimming the video first and answering directly if the question is general, or revisiting specific segments based on memory if the query requires fine-grained details. We plan to prioritize exploring such complex reasoning patterns in future work. Notably, this form of adaptive visual thinking

is unattainable without POINTS-Long's dual-mode design, which uniquely enables reasoning over the entire video context. We hope this establishes a new direction for the field of visual reasoning.

Figure 2. **Visualization of Position Encoding.** We initialize learnable standby tokens by uniformly sampling RoPE embeddings from the original sequence. We visualize their attention maps in the last ViT layer, marking assigned positions with a yellow square. For clarity, we display only the top 10% of attention weights, where darker red indicates higher intensity. The results reveal a strong localization effect: standby tokens primarily absorb information from their neighboring patches.

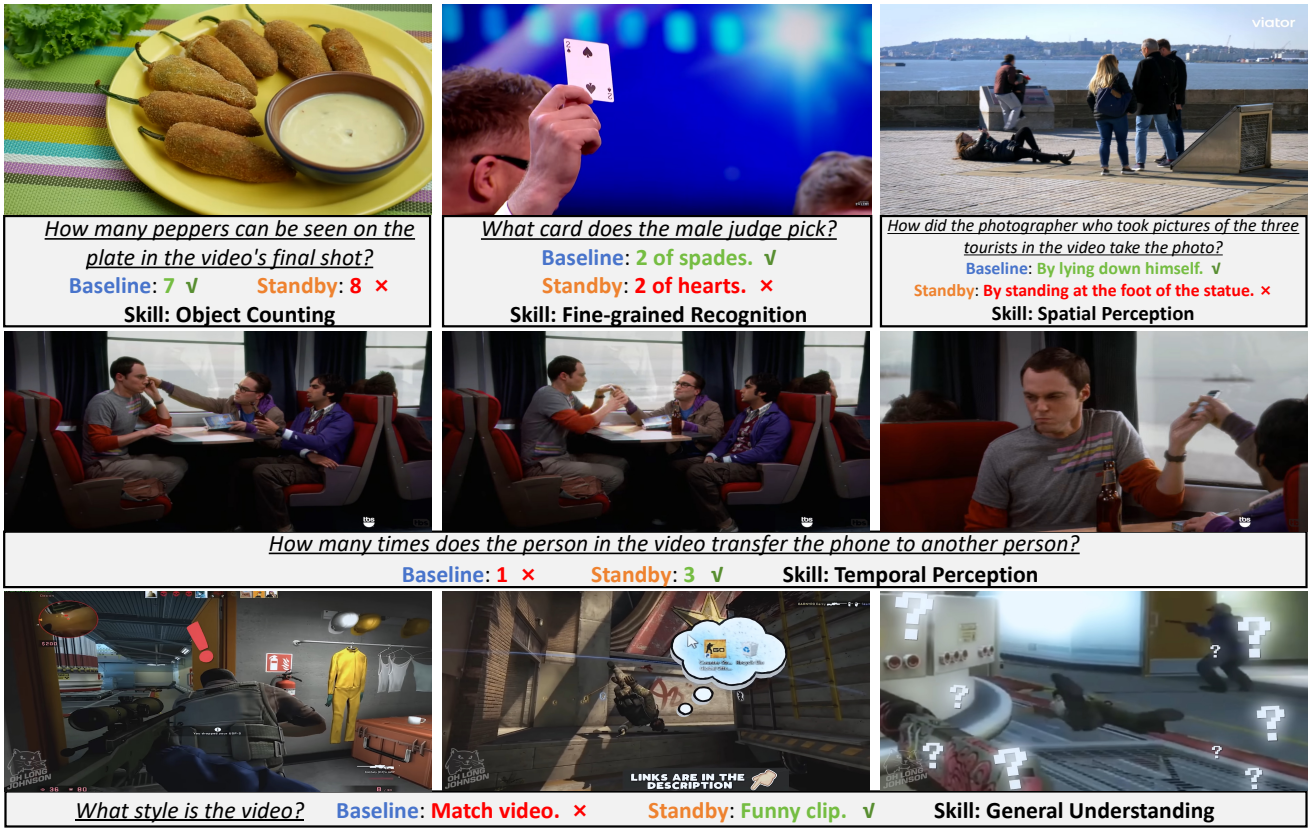

Figure 3. **Failure case analysis.** Standby mode fails on spatial or fine-grained perception while the baseline fails more on temporal and general understanding.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 4

[3] 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. 4

[4] Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. *arXiv preprint arXiv:2412.12075*, 2024. 4

[5] 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? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024. 4

[6] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024. 1

[7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 6

[8] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023. 4

[9] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025. 1

[10] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024. 4

[11] Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37:89098–89124, 2024. 4

[12] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024. 1

[13] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025. 4

[14] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. 1

[15] Jiaxi Gu, Xiaojun Meng, Guansong Lu, Lu Hou, Niu Minzhe, Xiaodan Liang, Lewei Yao, Runhui Huang, Wei Zhang, Xin Jiang, et al. Wukong: A 100 million large-scale chinese cross-modal pre-training benchmark. *Advances in Neural Information Processing Systems*, 35:26418–26431, 2022. 1

[16] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385, 2024. 4

[17] Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. Worldsense: Evaluating real-world omnimodal understanding for multimodal llms. *arXiv preprint arXiv:2502.04326*, 2025. 4

[18] Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. *arXiv preprint arXiv:2412.16117*, 2024. 5, 6

[19] Yiming Jia, Jiachen Li, Xiang Yue, Bo Li, Ping Nie, Kai Zou, and Wenhu Chen. Visualwebinstruct: Scaling up multimodal instruction data through web search. *arXiv preprint arXiv:2503.10582*, 2025. 2

[20] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016. 4

[21] 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*, 2023. 1

[22] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 4

[23] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 4

[24] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024. 4

[25] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024. 4

[26] Yuan Liu, Le Tian, Xiao Zhou, Xinyu Gao, Kavio Yu, Yang Yu, and Jie Zhou. Points1. 5: Building a vision-language model towards real world applications. *arXiv preprint arXiv:2412.08443*, 2024. 1, 6

[27] Yuan Liu, Zhongyin Zhao, Ziyuan Zhuang, Le Tian, Xiao Zhou, and Jie Zhou. Points: Improving your vision-language model with affordable strategies. *arXiv preprint arXiv:2409.04828*, 2024. 1

[28] Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Zhou Xiao, Yang Yu, et al. Points-reader: Distillation-free adaptation of vision-language models for document conversion. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1601, 2025. 1

[29] Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025. 5

[30] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 4

[31] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023. 4

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

[33] Ruchit Rawal, Khalid Saifullah, Miquel Farré, Ronen Basri, David Jacobs, Gowthami Somepalli, and Tom Goldstein. Cinepile: A long video question answering dataset and benchmark. *arXiv preprint arXiv:2405.08813*, 2024. 1

[34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 1

[35] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 1

[36] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025. 5, 6

[37] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 2

[38] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 4

[39] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 1

[40] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv preprint arXiv:2503.20752*, 2025. 2

[41] Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18992–19001, 2025. 5, 6

[42] Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, Shuai Xiao, and Enzo Tartaglione. Folder: Accelerating multi-modal large language models with enhanced performance. *arXiv preprint arXiv:2501.02430*, 2025. 6

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1

[44] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8428–8437, 2025. 1

[45] Wenhao Wang and Yi Yang. Videoufo: A million-scale user-focused dataset for text-to-video generation. *arXiv preprint arXiv:2503.01739*, 2025. 1

[46] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025. 4

[47] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022. 6

[48] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024. 4

[49] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[50] Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu, and Hai Zhao. Vript: A video is worth thousands of words. *Advances in Neural Information Processing Systems*, 37:57240–57261, 2024. 1

[51] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025. 5, 6

[52] Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14022–14032, 2024. 1

[53] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 4

[54] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 4

[55] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. 4

[56] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander G Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 694–717, 2025. 1

[57] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 1

[58] Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model training. *arXiv preprint arXiv:2503.19633*, 2025. 2

[59] 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 e-prints*, pages arXiv–2406, 2024. 4