Streaming Video Instruction Tuning

Supplementary Material

A. Streamo

A.1. System Prompt

We design a dedicated system prompt for `Streamo` that enables the model to handle dynamic streaming video content, interpret three predefined response states, and make real-time decisions at the frame level. The full prompt is provided in Tab. 5. This deliberately crafted prompt helps the model quickly adapt to the streaming input pattern and perform the required behavior transformation.

A.2. Instruction Prompt

In Tab. 9, we present the prompt templates used for all tasks. These diverse task instructions help the model better understand different task requirements, thereby fostering more general multi-task instruction-following capabilities. This goes beyond prior setups where models were confined to standalone QA, and represents a step toward general real-time interactive AI.

A.3. More Experimental Results

Our training framework converts offline models into streaming-capable models with minimal intrusive modifications, enabling these base models to process streaming video data. This design yields strong compatibility and allows direct application to a wide range of offline models. In Tab.6 and7, we further report results using InternVL3[23] and Qwen3VL[16] as `Streamo`'s base models. These results show that our framework effectively leverages the capabilities of offline models and extends them to online streaming video processing. This is particularly advantageous given the rapid iteration of offline models, as our framework can readily harness their improvements for real-time interactive video understanding.

Meanwhile, we also evaluated Streamo on ViSpeak-Bench [7], shown in Tab. 3. The results show that our method achieves a clear advantage in response-time accuracy, demonstrating the effectiveness and soundness of our response architecture.

A.4. Visualization

In Fig.3 and4, we visualize the outputs of `Streamo`, which vividly illustrate its ability to interpret and appropriately respond even to instructions that were unseen during training. When confronted with task instructions that vary in both response granularity and content, the model consistently produces suitable outputs. These visualizations provide strong evidence that `Streamo`'s training framework successfully bridges the gap between offline model capabilities and the requirements of online streaming interactions, enabling reliable real-time responses that go far beyond simple QA.

Table 1. Comparison of Existing Video Benchmarks. Streamo-Bench introduces the first mixed-task type specifically designed for streaming video.

Benchmark	#Videos	#Samples	Streaming	Task Type
MVBench	3,673	4,000	✗	QA
TempCompass	410	7,540	✗	QA
ET-Bench	7,002	7,289	✗	Mix
SVBench	1,353	49,979	✓	QA
StreamBench	306	1,800	✓	QA
OVOBench	644	2,814	✓	QA
Streamo-Bench	300	3000	✓	Mix

A.5. Further Analysis of the Three-State Design

To examine the rationale behind our training architecture more thoroughly, we compare the proposed Three-state Design with an alternative approach based on the [EOS] token. As shown in Tab. 2, the [EOS]-based model exhibits notable performance drops, particularly on proactive tasks (i.e., FAR) and grounding tasks. These results demonstrate that our three-state design consistently outperforms EOS-only training while introducing only negligible additional cost.

We attribute this gap to the fact that [EOS] maps both irrelevant and partially relevant segments to the same token. As a result, the model is encouraged to remain silent even when encountering relevant frames, causing it to miss the optimal timing for response. In contrast, the introduction of a [Standby] token alleviates this misalignment by explicitly marking relevant frames as soon as the event begins and preserving this state throughout the relevant interval. This leads to more accurate temporal alignment and more complete coverage, which is reflected in the higher grounding TIoU.

Table 2. Comparison on the same training dataset, Streamo-Instruct, where the only change is replacing the proposed three-state design with EOS-only training. Using only [EOS] degrades performance, especially on proactive prediction (FAR) and forward grounding, highlighting the benefit of the three-state design.

Model	OVOBench				Streamo-Bench
	RTVP	BT	FAR	AVG	Forward Grounding
Streamo-3B	61.51	41.76	53.72	52.33	14.7
Streamo-3B w/ EOS	60.93	39.43	45.22	48.52	9.3

A key advantage of [Standby] is that it explicitly models frames that are already relevant but not yet ready for a final response. As shown in Fig. 1, because the query is specifically about ASWIN, the model switches to [Standby] once ASWIN appears and the attempt becomes temporally relevant, even though the final outcome is still uncertain. This allows the model to preserve attention over the ongoing event instead of treating these frames as irrelevant. Meanwhile, for grounding, the continuous [Standby] state helps cover the full event span more completely, rather than activating only near the final decisive moment.

B. Streamo-Instruct

B.1. Data Generation Prompt

We next elaborate on the prompts used in our data annotation pipeline. For event caption tasks, we leverage ARC-Hunyuan[8], which is specifically trained for video segmentation and grounding, and directly adopt its official prompt for initial data processing. We then use the prompt in Tab.8 to rewrite and clean the annotated caption sentences. For narration generation, which describes inter-frame temporal changes, the generation prompt is given in Tab.10, and the prompt for merging and cleaning the resulting descriptions is provided in Tab.11. For the TSQA task, the detailed prompt is presented in Tab. 12.

C. Streamo-Bench

In Tab. 1, we compare our proposed Streamo-Bench with existing video benchmarks. Streamo-Bench is, to the best of our knowledge, the first streaming video benchmark that integrates multiple task types. Existing streaming video benchmarks typically use QA as the sole evaluation task, which mainly measures perceptual understanding rather than the ability to perform diverse open-ended tasks. However, the ability to follow varied instructions and complete multiple tasks is a key requirement for streaming video models. By filling this gap, Streamo-Bench enables more comprehensive evaluation of a model's instruction-following ability in open-ended streaming scenarios.

C.1. Statistics

Our benchmark contains 300 videos sampled from COIN[14], YouCookv2[22], and ActivityNet [2]. Each video is annotated with multiple tasks, including Grounding, Narration, Caption, and Time-Sensitive QA, yielding a total of 3,000 task-specific instances. Each video in Streamo-Bench contains 2x grounding (forward + backward) tasks, 1x dense caption task, and 1x narration task, with the rest being TSQA. This comprehensive design enables a thorough examination of a model's ability to process and respond to diverse instructions in streaming settings.

C.2. Metric

To comprehensively evaluate the performance of models on our Streamo-Bench, we detail the metrics used for each task type below.

Grounding Evaluation. For grounding tasks, we distinguish between forward (queries referring to time points before an event) and backward (queries referring to time points after an event) contexts. Performance is measured using mean Intersection over Union (mIoU), which quantifies the overlap between the model's predicted temporal interval and the ground-truth interval.

Let the predicted and ground-truth temporal intervals, t^{pred} and t^{gt}, for sample i be:

$$t_i^{\text{pred}} = [\, s_i^{\text{pred}},\, e_i^{\text{pred}} \,], \qquad t_i^{\text{gt}} = [\, s_i^{\text{gt}},\, e_i^{\text{gt}} \,], \tag{1}$$

where s and e represent the start and end timestamps, respectively. The IoU for sample i is defined as the ratio of intersection length to union length:

$$\text{IoU}_i = \frac{\max\big(0,\, \min(e_i^{\text{pred}}, e_i^{\text{gt}}) - \max(s_i^{\text{pred}}, s_i^{\text{gt}})\big)}{\max(e_i^{\text{pred}}, e_i^{\text{gt}}) - \min(s_i^{\text{pred}}, s_i^{\text{gt}})}. \tag{2}$$

The mean IoU (mIoU) over N samples is

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \text{IoU}_i. \tag{3}$$

Narration and Caption Evaluation. Because narration and captioning are open-ended generation tasks, directly evaluating output quality is challenging. Following the evaluation protocol of Chatbot Arena [21] and StreamingVLM [18], we assess narration and caption quality via pairwise comparison against a strong baseline, Qwen2.5-VL-72B [1]. The win rate is defined as the proportion of cases in which our model's output is judged superior to the baseline's output.

Time-Sensitive QA Evaluation. For Time-Sensitive QA, we require that a prediction be correct in both its content and its timestamp. Let Q be the set of TSQA questions. For each question $q \in Q$, the ground truth consists of m_q time-stamped answers:

$$G_q = \{(a_i^q, t_i^q)\}_{i=1}^{m_q}, \tag{4}$$

where a_i^q is the answer content and t_i^q is its timestamp. The model produces n_q predictions:

$$P_q = \{(\hat{a}_j^q, \hat{t}_j^q)\}_{j=1}^{n_q}, \tag{5}$$

where $\hat{a}_j^q$ is the predicted content and $\hat{t}_j^q$ is the predicted timestamp.

A predicted pair $(\hat{a}_j^q, \hat{t}_j^q)$ may match a ground-truth pair (a_i^q, t_i^q) only if it is correct in both content and time. For the

Table 3. Performance of streamo compared to various MLLMs on ViSpeak-Bench.

Method	Params	Frames	Omni	Streaming	Time Accuracy (%)							Text Score								Overall
					AW	VI	HR	VW	VT	GU	**All**	VR	AW	VI	HR	VW	VT	GU	**All**	
Human (Avg)	-	-	-	-	70.00	100.00	90.00	92.00	96.00	98.80	91.13	4.80	2.45	4.58	3.06	5.00	5.00	2.85	3.96	3.69
Human (Max)	-	-	-	-	70.00	100.00	100.00	100.00	100.00	100.00	95.00	5.00	2.71	5.00	3.62	5.00	5.00	3.19	4.22	4.01
Proprietary MLLMs																				
Gemini 1.5 pro [15]	-	-	✓	✗	46.00	60.00	85.00	84.00	48.00	97.00	70.00	3.03	2.34	2.93	1.36	4.66	4.68	2.07	3.01	2.19
GPT-4o [9]	-	-	✓	✗	48.50	82.00	96.00	99.00	100.00	99.50	87.50	3.18	2.27	3.53	1.71	5.00	4.98	2.22	3.27	2.99
Open-Source Video MLLMs																				
InternVL-2.5 [4]	8B	16	✗	✗	41.50	55.50	46.00	96.00	72.00	99.50	68.42	2.93	2.16	3.67	0.74	3.05	4.81	1.26	2.66	1.98
Qwen2.5-VL [1]	7B	1 fps	✗	✗	42.50	78.00	31.00	95.00	85.00	98.50	71.67	2.34	2.31	2.31	1.32	5.00	3.91	1.02	2.60	2.25
Qwen2.5-VL [1]	72B	1 fps	✗	✗	44.50	81.00	77.00	91.00	91.00	93.00	79.58	3.15	2.64	3.36	1.00	5.00	5.00	1.50	3.09	2.62
VITA 1.5 [6]	7B	1 fps	✓	✗	18.00	46.00	40.00	88.00	49.00	97.50	56.42	2.40	2.08	0.57	0.85	4.57	4.49	1.18	2.31	1.54
Ola [11]	7B	1 fps	✓	✗	27.00	67.00	44.00	89.00	69.00	98.50	65.75	2.95	1.81	2.67	0.55	4.71	3.67	1.52	2.55	1.86
FlashVstream [19]	7B	1 fps	✗	✓	34.00	16.00	48.00	75.00	33.00	99.50	50.92	1.75	1.63	1.31	0.67	4.88	4.61	0.70	2.22	1.24
Dispider [12]	7B	16	✗	✓	38.50	70.00	44.00	69.00	100.00	99.50	70.17	2.50	1.75	4.06	0.91	0.61	2.49	2.07	2.06	1.63
ViSpeak [7]	7B	1 fps	✓	✓	56.50	72.00	83.00	93.00	79.00	99.00	80.42	3.75	2.63	3.84	1.07	4.95	3.15	3.36	3.25	2.76
Streamo	7B	1 fps	✗	✓	59.00	79.00	82.00	97.00	86.00	100	83.83	2.73	2.31	3.62	1.33	4.96	3.62	2.97	3.08	2.71

content evaluation:

$$C(\hat{a}_j^q, a_i^q) = \begin{cases} 1, & \text{if content matches,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

For the timestamp, we define a non-negative tolerance parameter $\delta_t \geq 0$. Then we evaluate the correctness of the timestamp by:

$$T(\hat{t}_j^q, t_i^q; \delta_t) = \begin{cases} 1, & \text{if} \quad |\hat{t}_j^q - t_i^q| \leq \delta_t, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In our experimental setting, the δ_t is set to 3 seconds. For the i-th answer point of question q, we define an indicator I_i^q that checks whether there exists at least one prediction satisfying both content and temporal constraints: :

$$I_i^q = \begin{cases} 1 & \text{if} \quad C(\hat{a}_j^q, a_i^q) = 1 \wedge T(\hat{t}_j^q, t_i^q; \delta_t) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The final accuracy and recall can be given as:

$$\text{Accuracy} = \frac{1}{\sum_{q \in Q} m_q} \sum_{q \in Q} \sum_{i=1}^{m_q} I_i^q \tag{9}$$

$$\text{Recall} = \frac{1}{|Q|} \sum_{q \in Q} \left(\frac{1}{m_q} \sum_{i=1}^{m_q} I_i^q \right) \tag{10}$$

C.3. Sample Visualization

A sample instance from Streamo-Bench is illustrated in Fig. 2. Forward and backward grounding questions are randomly placed either before or after their corresponding target temporal intervals. The TSQA question is inserted before the first answer timestamp. Narration and event caption instructions are placed before the start of the video stream to capture the overall video content.

C.4. Further Analysis

We further analyze the performance of existing models on Streamo-Bench and observe that their primary failures stem from a lack of instruction–task comprehension: they struggle to distinguish different task types and to produce task-appropriate outputs. This limitation arises because these models are typically trained exclusively on captioning or QA data, which constrains them to generate outputs tailored to only those specific tasks.

Examples in Tab. 4 clearly illustrate this phenomenon: while the models can satisfy caption or narration requirements, they often fail to understand grounding instructions and instead fall back to generic video descriptions. For TSQA tasks, although models trained on QA data can answer content-related questions, they do not properly follow instructions that require real-time updates to answers over the video timeline, leading to task failure.

In summary, existing models generally lack robust multi-task understanding, whereas Streamo-Bench is specifically designed to evaluate a model's ability to interpret and respond to task-specific instructions in streaming scenarios.

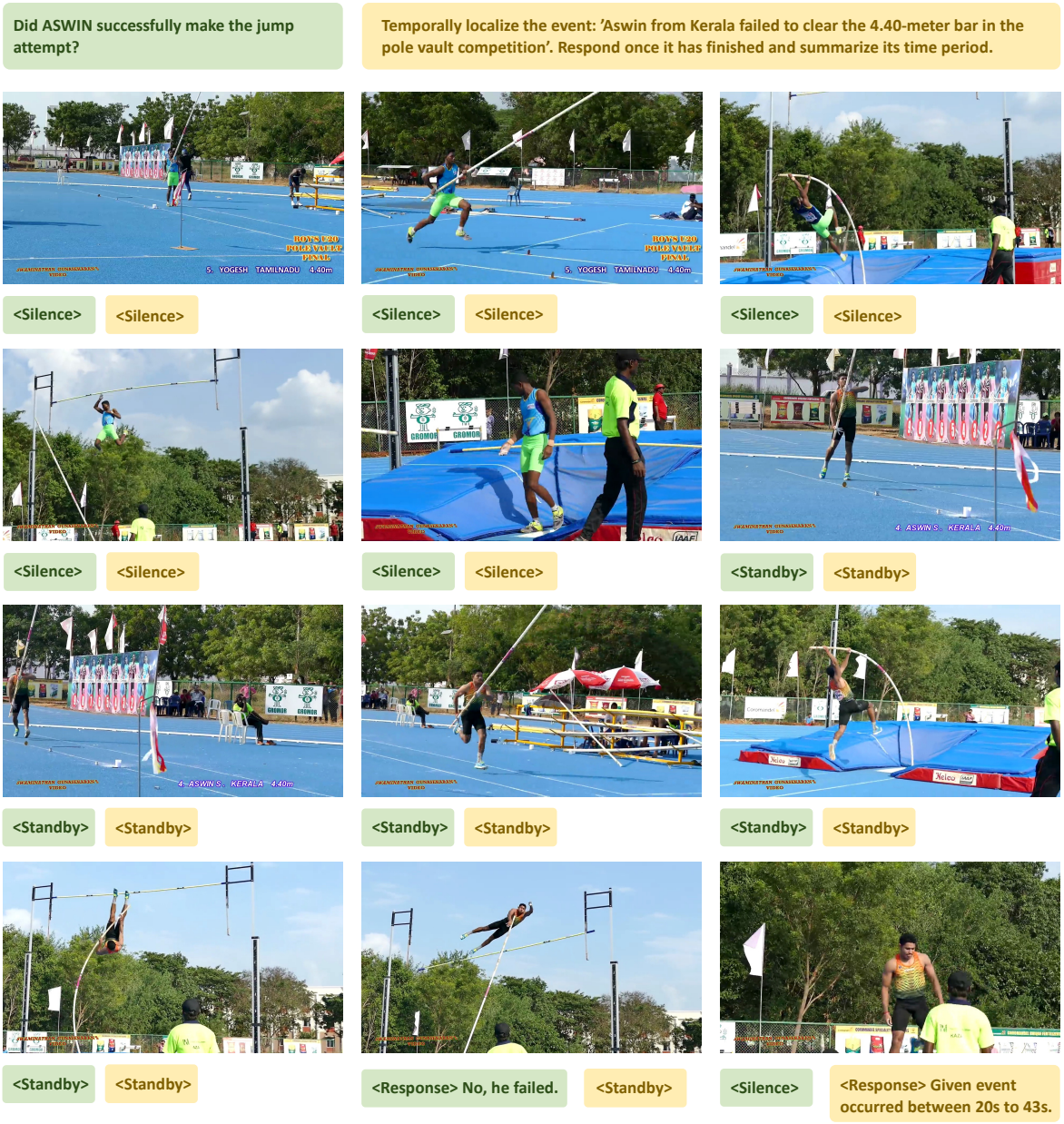

Figure 1. Visualization of the three-state decoding process. The model stays in `[Silence]` for irrelevant frames, switches to `[Standby]` once the query-relevant event involving ASWIN begins, and emits `[Response]` only after the outcome becomes clear. For the grounding task, the persistent `[Standby]` state helps preserve attention over the relevant interval and enables more complete temporal coverage of the event span.

Figure 2. Streamo-Bench example illustrating multi-task instruction-following evaluation.

Input Video				
				

Prompt	Model Output
Real-Time Narration Task: Watch the following video and provide a real-time narration.	Here's a typical mid-late 18th Century stoneware tea pot with the rounded spout and finial for pouring. After pouring when you were pouring out the last bit of liquid from the pot, you would be careful to stir away any bits of leaf which were floating around in the bottom.
Event Caption Task: List every event happening in the video.	We'll go through all the steps to make tea the way the Victorians did step by step. When you think of what they would be using here in their kitchen spaces, you can see they've got the teapot, and she's working on getting her leaves in. The method for both is basically exactly the same. With the fresh leaf version in this case, we're just adding in equal amounts of leaves and water.
Event Grounding Task: Localize this event: 'add some water to the tea' in the following video.	In the Victorian era the family kitchen was an even more important part of home life than in previous as this was where most of their time was spent. It was where the family ate their meals but also where they would come together. Whether it was afternoons at tea or a simple evening meal, the Victorian kitchen was a place that families would gather together and enjoy good company and food. Here we have made our own little tea service here, and I will be showing you how Mrs Bronte would serve it in her kitchen.
Time-Sensitive QA Task: What is the woman doing? A. Preparing tea in kitchen B. Sitting at desk writing C. Drinking tea at desk. Please provide your answer by stating the letter followed by the full option. If the correct answer later changes, update your response.	A. Preparing tea in kitchen. You will learn to make a cup of tea the Victorian way." To begin you'll need something sweet to top with your tea and milk. Tea is made with sugar or honey and the Victorians actually used a lot of cream for their tea. I'm going to use some milk for my tea today. And it can take up to two minutes to boil, so if you're making tea for more than one person.", Start with just one cup at a time and then add more when ready.

Table 4. Example output from StreamingVLM illustrating a failure to follow diverse task instructions and generate the corresponding response.

System Prompt: You are a helpful assistant specializing in streaming video analysis.
You will receive input frame by frame, each labeled with absolute time intervals
in the exact format <Xs-Ys> (e.g., <0s-1s>). Follow these rules precisely:

1. Use </Silence> when:
   - No relevant event has started, OR
   - The current input is irrelevant to the given question.

2. Use </Standby> when:
   - An event is in progress but has not yet completed, OR
   - The current input is relevant but the question cannot yet be answered.

3. Use </Response> only when:
   - An event has fully concluded, OR
   - The available information is sufficient to fully answer the question.
Provide a complete description at this point.

Do not provide partial answers or speculate beyond the given information.
Whenever you deliver an answer, begin with </Response>.

Table 5. System prompt used in `Streamo`.

Table 6. Additional online benchmark evaluation results of Streamo framework with different base models (InternVL3 and Qwen3VL). Our framework consistently enables strong real-time streaming performance across diverse offline backbones.

Model	# Frames	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall Avg.
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Overall Avg.
Open-source Offline Models																	
Qwen2-VL-72B [17]	64	65.77	**60.55**	69.83	51.69	69.31	54.35	61.92	52.53	**60.81**	**57.53**	**56.95**	**38.83**	64.07	45	49.3	**56.27**
LLaVA-Video-7B [20]	64	**69.13**	58.72	68.83	49.44	**74.26**	59.78	63.52	**56.23**	57.43	7.53	40.4	34.1	**69.95**	60.42	**54.82**	52.91
LLaVA-OneVision-7B [10]	64	66.44	57.8	**73.28**	**53.37**	71.29	**61.96**	**64.02**	54.21	55.41	21.51	43.71	25.64	67.09	58.75	50.5	52.74
Qwen2-VL-7B [17]	64	60.4	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	56.08	46.46	31.66	65.82	48.75	48.74	50.39
InternVL-V2-8B [5]	64	67.11	**60.55**	63.79	46.07	68.32	56.52	60.39	48.15	57.43	24.73	43.44	26.5	59.14	54.14	46.6	50.15
LongVU-7B [13]	1fps	53.69	53.21	62.93	47.75	68.32	59.78	57.61	40.74	59.46	4.84	35.01	12.18	69.48	**60.83**	47.5	46.71
Open-source Online Models																	
Flash-VStream-7B [19]	1fps	24.16	29.36	28.45	33.71	25.74	28.8	28.37	39.06	37.16	5.91	27.38	8.02	67.25	60	45.09	33.61
VideoLLM-online-8B [3]	2fps	8.05	23.85	12.07	14.04	45.54	21.2	20.79	22.22	18.8	12.18	17.73	-	-	-	-	-
Dispider-7B [12]	1fps	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.3	36.06	18.05	37.36	48.75	34.72	41.78
Streamo Framework																	
Streamo-3B *(Qwem2.5-VL)*	1fps	78.52	52.29	67.24	44.38	55.45	71.20	61.51	51.18	57.43	16.67	41.76	27.94	50.72	82.5	53.72	52.33
Streamo-7B *(Qwem2.5-VL)*	1fps	79.19	57.80	75.00	49.44	64.36	70.11	65.98	54.55	52.03	31.72	46.10	29.96	51.03	83.33	54.77	55.61
Streamo-2B *(InternVL3)*	1fps	77.18	55.96	62.07	41.01	60.40	70.11	61.12	48.82	47.30	13.44	36.52	29.23	47.38	80.42	52.34	49.99
Streamo-4B *(Qwen3-VL)*	1fps	82.55	69.72	74.14	52.25	73.27	81.52	72.24	58.19	52.70	17.20	42.70	31.38	53.90	84.17	56.48	55.10

Table 7. Additional offline benchmarks results of Streamo framework with different base models (InternVL3 and Qwen3VL). The results show that our training framework preserves the underlying offline capability while extending it to streaming video processing.

Model	OVO Real-Time	OVO Backward	MVBench	TempCompass	VideoMME	LongVideoBench	Avg
Proprietary Models							
Gemini-1.5-pro [15]	69.3	62.5	60.5	67.1	75.0	64.0	66.4
GPT-4o [9]	64.5	60.8	64.6	70.9	71.9	66.7	66.6
Open-source Online Models							
Flash-VStream-7B [19]	28.4	27.4	61.2	-	61.2	-	-
VideoLLM-online-8B [3]	20.8	17.7	33.9	-	26.9	-	-
Dispider-7B [12]	54.6	36.1	-	-	57.2	-	-
StreamingVLM-7B [18]	62.0	-	69.2	-	65.1	59.0	-
Streamo Framework							
Qwen2.5-VL-3B [1]	54.6	37.8	67.0	64.4	61.5	54.2	56.6
Streamo-3B	61.5 (+6.9)	41.8 (+4.0)	67.9 (+0.9)	66.2 (+1.8)	61.8 (+0.3)	56.2 (+2.0)	59.2 (+2.6)
Qwen2.5-VL-7B [1]	58.8	42.2	69.6	71.7	65.1	56.0	60.6
Streamo-7B	66.0 (+7.2)	46.1 (+3.9)	72.3 (+2.7)	71.8 (+0.1)	67.9 (+2.8)	59.2 (+3.2)	63.9 (+3.3)
InternVL3-2B [23]	59.5	36.4	70.4	57.6	58.9	55.4	56.4
Streamo-2B	61.1 (+1.6)	36.5 (+0.1)	71.4 (+1.0)	57.8 (+0.2)	60.1 (+1.2)	56.5 (+1.1)	57.3 (+0.9)
Qwen3-VL-4B [16]	66.5	42.8	68.9	65.8	69.3	53.2	61.1
Streamo-4B	72.2 (+5.7)	42.7 (-0.1)	70.4 (+1.5)	66.3 (+0.5)	68.7 (-0.6)	56.1 (+2.9)	62.8 (+1.7)

Event Rewriting Prompt: You are given a set of video captions, each describing a specific moment in a video. For each caption, perform the following tasks:

1. Remove any transition words, discourse markers, or sequence indicators (e.g., ”Finally ”Then ”Next ”Afterwards ”At the beginning ”At the end ”The video ends with ”The scene starts with etc.) at the beginning of the sentence or within the sentence, as these captions are now independent and do not need such connectors or structural descriptions.
2. Rewrite the caption to make it more concise and clear, without changing its meaning or omitting any important information.
3. Preserve all factual details and key actions described in the original caption.
4. Do not add any extra interpretation, information, or imagination not present in the original sentence. Only use the information given.
5. If the sentence includes a phrase describing the position of a shot or the sequence within the video (such as ”The video ends with ”At the start of the video ”In the next scene ”The video conclude with”), remove this part entirely. Focus only on describing the content of the shot.

Example:
Original: ”Finally, the video cuts back to the man in the indoor setting, who concludes the presentation by holding the bow.”
Optimized: ”The man in the indoor setting concludes the presentation by holding the bow.”

Process each caption in this way. Return the optimized sentence directly.
Original:{sentences}
Optimized:

Table 8. Task prompt used for rewriting event caption.

Real-time Narration Task:

- Provide a continuous, time-synchronized narration of the video, describing actions, objects, and scene changes as they occur.
- Narrate the video in real time, updating the description frame-by-frame or moment-by-moment as events unfold.
- Generate live commentary of the video, focusing on who is doing what, where, and when, and noting any transitions or new events immediately.
- Deliver an on-the-fly description of the video, highlighting salient actions, interactions, and changes in context as soon as they appear.
- Produce a running narration that captures ongoing activities, brief pauses, and resumptions, maintaining temporal alignment with the video timeline.

Action Caption:

- Find, identify, and determine the temporal boundaries of a series of distinct actions or steps occurring throughout the video.
- Locate and describe a series of actions or steps in the video.
- Locate and pinpoint a sequential series of specific actions or steps in the video.
- Identify and mark the video segments corresponding to a series of actions or steps.
- Identify and localize a series of steps or actions occurring in the video.

Event Caption:

- Identify and describe all events in the following video.
- List every event happening in the following video with descriptions.
- Detect and summarize each event sequence in the following video.
- Extract and explain all notable events in the following video.
- Find all significant events in the following video and describe them.

Event Grounding:

- Watch the following video and temporally localize the event. Respond once it has finished and summarize its time period. The given event is: '{caption}'
- Monitor the following video, identify the event, then respond after it finishes with a summary of its time window. The given event is: '{caption}'
- Analyze the following video, detect the event and report back upon its completion with its time period. The given event is: '{caption}'
- Review the following video, localize the event in time, then notify me once it ends and summarize the interval it occupies. The given event is: '{caption}'
- Identify and temporally segment the event in the following video. Report after it finishes with its time period and duration. The given event is: '{caption}'

Time-sensitive QA:

- {question} If the answer changes over time, update your response accordingly.
- {question} Update your answer if it becomes different at a later time.
- {question} If it later differs, update your response promptly.
- {question} Refresh your answer upon any change.
- {question} If the correct answer later changes, update your response.

Table 9. Prompt template used for diverse streaming video tasks.

Video Description Prompt: You are given two consecutive seconds in a video (2 frames per second). Please succinctly describe the most significant operation or change that occurred between these seconds, focusing on the following points:
1. Base your description solely on clearly observable information; avoid speculation or assumptions.
2. For each object or element that changed, briefly state what changed: position, movement, actions, shape, color, etc.
3. Only describe the main operation, event, or action that happened—avoid listing small movements or minor shifts.
4. Describe only the specific changed parts with clear and direct language; do not include unchanged content or summarize the overall scene.
5. Make your description short and focused, naming only the changes without referencing the sequence of frames or including explanations.

Example:
'A woman appears.'
'You pick up a scissor.'
'The cup moves to the left.'
'A cat enters the frame.'
'The red ball rolls closer.'
'The lamp turns on.'
'The book closes.'
'A hand takes the remote.'
'The door opens further.'

Only provide the most important description or a summary of multiple descriptions.

Table 10. Task prompt used for frame-level video description generation.

Narration Generation Prompt:
Objective:
Clean the following second-by-second video descriptions to enhance coherence and eliminate redundancy. The original descriptions were generated with visibility of only the preceding and following 2 seconds, making them repetitive and disjointed.

Task:
Transform the descriptions into a smooth, logical narrative by:
1. Removing Redundancy: Omit repeated descriptions of static or ongoing actions.
2. Filtering Insignificant Details: Exclude minor or fleeting actions that do not impact overall understanding.
3. Sentence Shortening: If a description significantly exceeds 5 words, rewrite it to approximately 5 words while preserving the main idea.
4. Merging Consecutive Events: Combine adjacent descriptions representing a continuous or complete action into a single, concise sentence (e.g., "002: Man touches socket" and "003: Socket disappears" $\rightarrow$ "003: Man removed socket").

Output Format and Rules:
1. Use the format: SSS: one-sentence description.
2. When merging or omitting descriptions, skip the corresponding timestamps.
3. Do not add explanations, notes, or blank lines.
4. If the descriptions are repetitive, monotonous, lack meaningful variation, or are confusing, ambiguous, or insufficient, output only: Negative Sample.

Description:
{Description}

Table 11. Task prompt used for merging the frame description to generate real-time narration.

TSQA Generation Prompt: You are a Time-Sensitive Video Question Generator. You need to identify all the elements in the video that change over time and formulate them into questions.

CORE REQUIREMENT
Every question MUST have answers that CHANGE over time. If something doesn't change during the video, DO NOT create a question about it.

TASK

1. Identify ONLY aspects that visibly CHANGE during the video. Ignore:
   - Static elements that remain constant
   - Transitions, previews, close-ups that don't alter facts
   - Opening/closing sequences

2. For each changing aspect, generate ONE question with MULTIPLE DIFFERENT answers:
   - Each question MUST have at least 2 DISTINCT answer values
   - Answers must represent actual changes observed at different times
   - Never repeat the same answer value

3. Question types:
   - **Descriptive**: What/Which/Who (e.g., "What color is the ball?")
   - **Counting**: How many/How much (e.g., "How many people are visible?")
   - **State**: What stage (e.g., "What is the person doing?")
   - **Action** : What is being added/used (e.g., "What ingredient is being added?")
   - **Binary**: Yes/No (e.g., "Is the bacon cooked?")

4. Answer format:
   - List answers chronologically
   - Include PRECISE time in seconds for each observed change
   - If state returns to a previous value, include it as a new entry

EXAMPLES
[{"question": "What color is the traffic light? "answers": [{"value": "red "time": 3.8}, {"value": "green "time": 8.7}, {"value": "yellow "time": 23.2}, {"value": "red "time": 26.4}]},
{"question": "How many people are in the frame? "answers": [{"value": 1, "time": 0.0}, {"value": 2, "time": 3.8}, {"value": 3, "time": 17.1}, {"value": 1, "time": 42.6}]},
{"question": "What is being poured into the glass? "answers": [{"value": "water "time": 2.3}, {"value": "milk "time": 19.7}, {"value": "orange juice "time": 31.4}]},
{"question": "Is the cake fully decorated? "answers": [{"value": "No "time": 13.7}, {"value": "Yes "time": 48.9}]}]

OUTPUT
   - Minimum 2 distinct answers per question
   - Precise timing for all changes. Times must reflect ACTUAL observed changes, not approximations
   - If unsure of exact timing, watch that segment again

Think step-by-step and ensure all requirements are met and all time are precise.

Table 12. Task prompt used for Time-Sensitive QA generation.

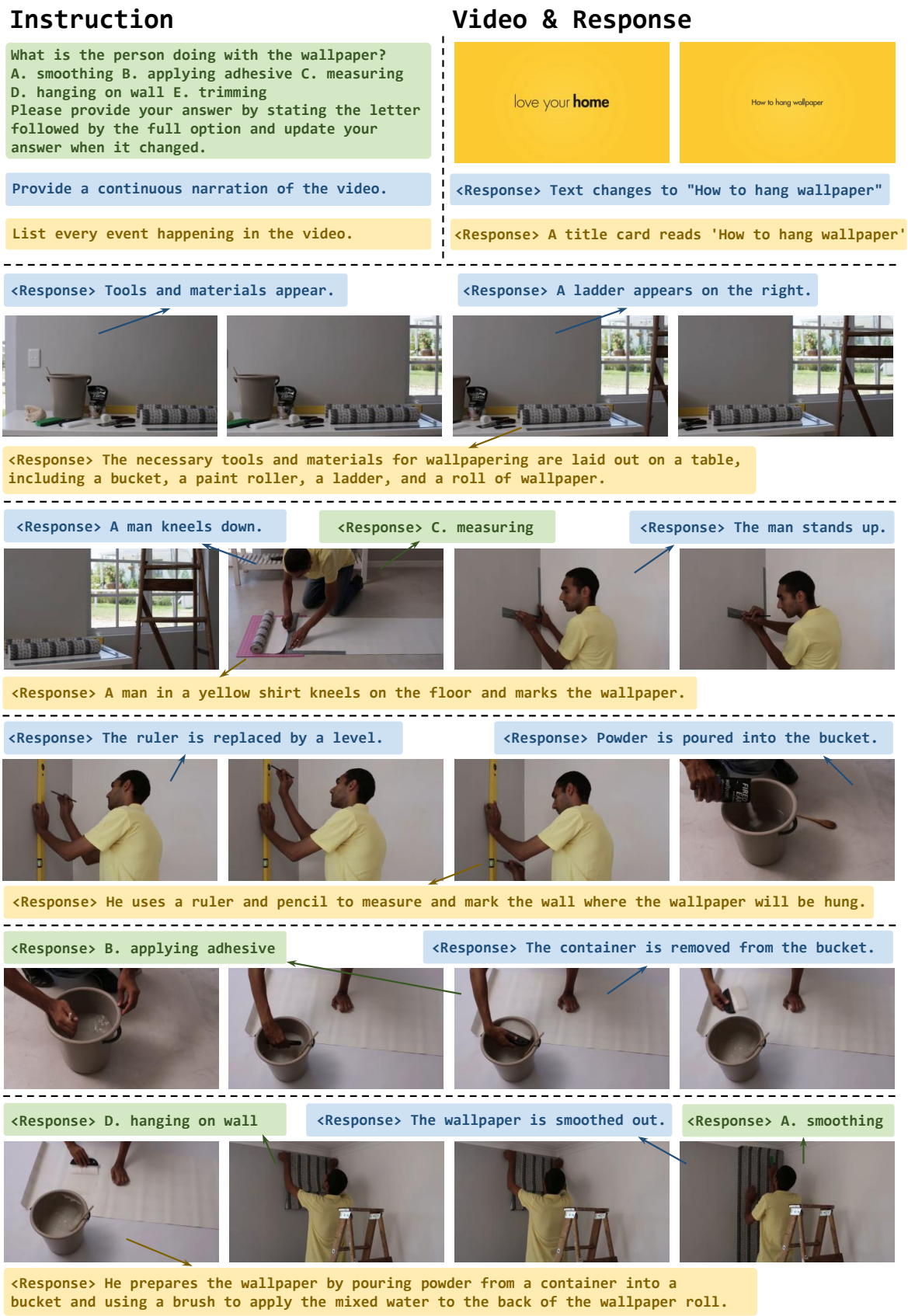

Figure 3. **Visualization of Streamo's outputs.** We present the results of Streamo processing the same video under different task instructions and distinguish them using different colors, including TSQA, Narration, and Caption. Arrows indicate the frames corresponding to the response moment.

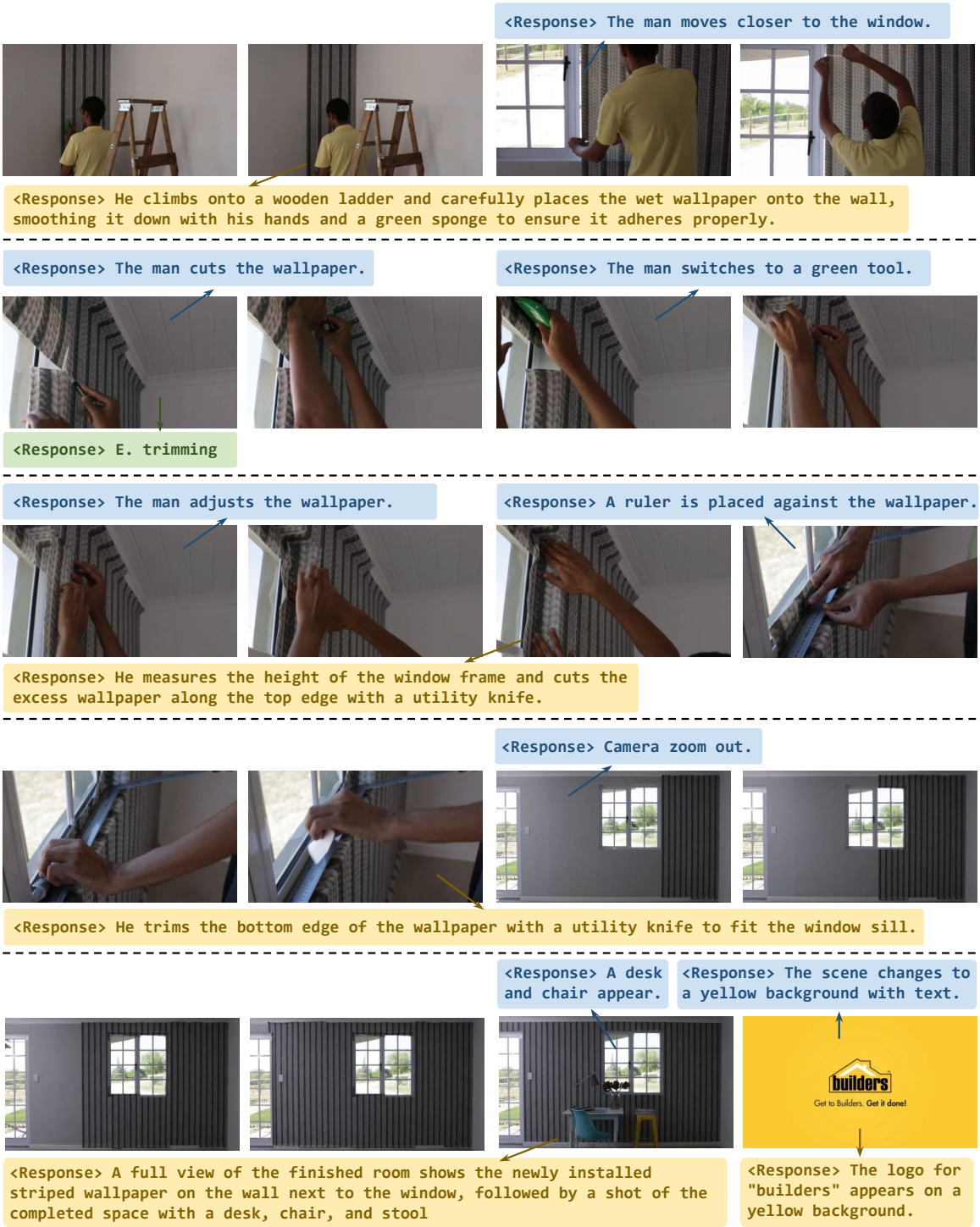

Figure 4. This is a continuation of the previous figure, showing the results for the same video.

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, 3, 8

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

[3] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. 7, 8

[4] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv:2412.05271*, 2024. 3

[5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 7

[6] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025. 3

[7] Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21778–21788, 2025. 1, 3

[8] Yuying Ge, Yixiao Ge, Chen Li, Teng Wang, Junfu Pu, Yizhuo Li, Lu Qiu, Jin Ma, Lisheng Duan, Xinyu Zuo, et al. Arc-hunyuan-video-7b: Structured video comprehension of real-world shorts. *arXiv preprint arXiv:2507.20939*, 2025. 2

[9] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv:2410.21276*, 2024. 3, 8

[10] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 7

[11] Zuyan Liu, Yuhao Dong, Jiahui Wang, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Ola: Pushing the frontiers of omni-modal language model. *arXiv preprint arXiv:2502.04328*, 2025. 3

[12] Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. *arXiv:2501.03218*, 2025. 3, 7, 8

[13] 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. 7

[14] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 2

[15] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024. 3, 8

[16] Qwen Team. Qwen3 technical report, 2025. 1, 8

[17] 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. 7

[18] Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025. 2, 8

[19] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv:2406.08085*, 2024. 3, 7, 8

[20] 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. 7

[21] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023. 2

[22] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2

[23] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 1, 8