Streaming Video Instruction Tuning

Jiaer Xia[1*] Peixian Chen[2*] Mengdan Zhang[2] Xing Sun[2] Kaiyang Zhou[1†]

[1] Hong Kong Baptist University
[2] Tencent Youtu Lab

Abstract

*We present **Streamo**, a real-time streaming video LLM that serves as a general-purpose interactive assistant. Unlike existing online video models that focus narrowly on question answering or captioning, Streamo performs a broad spectrum of streaming video tasks, including real-time narration, action understanding, event captioning, temporal event grounding, and time-sensitive question answering. To develop such versatility, we construct Streamo-Instruct-465K, a large-scale instruction-following dataset tailored for streaming video understanding. The dataset covers diverse temporal contexts and multi-task supervision, enabling unified training across heterogeneous streaming tasks. After training end-to-end on the instruction-following dataset through a streamlined pipeline, Streamo exhibits strong temporal reasoning, responsive interaction, and broad generalization across a variety of streaming benchmarks. Extensive experiments show that Streamo bridges the gap between offline video perception models and real-time multimodal assistants, making a step toward unified, intelligent video understanding in continuous video streams.*

1. Introduction

Recent advances in video large language models (LLMs) [4, 22, 39, 48] have demonstrated remarkable capabilities in analyzing complete, pre-recorded videos, which establish strong baselines for offline video understanding. These models excel at holistic reasoning over long temporal sequences when given static, temporally bounded inputs [18, 52], enabling tasks such as video captioning, summarization, and question answering. However, the requirements of real-time interactive AI assistants are fundamentally different: they must process continuous, unbounded video streams and respond to dynamic instructions as events unfold, often under strict latency constraints.

Existing offline models struggle to meet the demands of the streaming setting because they are designed to process entire clips before producing a single output [30, 36, 40]. In contrast, real-time applications require the model to continuously interpret an incoming video stream, detect when the visual context satisfies a task condition, and decide what information to output at that moment. This introduces two key challenges: 1) handling continuous, unbounded data flow without losing context, and 2) managing variable response timing and granularity across multiple tasks, which may require frame-level or longer-term temporal reasoning. A truly capable streaming video LLM must therefore integrate both task understanding and frame-level decision-making, enabling it to evaluate evolving visual contexts, determine appropriate moments to respond, and generate coherent outputs without delaying or missing critical information.

To address these challenges, recent studies [30, 36, 40] have attempted to extend offline video models for streaming by introducing a separate decision module that predicts response states before invoking the offline model to generate content. While this approach preserves the reasoning capacity of the base model, it creates a trade-off between accuracy and efficiency: lightweight decision modules often lack the capacity to fully understand complex instructions and temporal dependencies, while larger modules substantially increase computational cost and inference latency. Moreover, separating decision-making from response generation prevents tight coupling between perception and response, limiting the model's ability to seamlessly adapt to rapidly changing streaming contexts.

In this work, we propose `Streamo`, a real-time streaming video LLM that unifies decision-making and response generation in an end-to-end manner. Instead of relying on an external controller, we embed frame-level response state prediction directly into the model. Specifically, three decision heads—*Silence*, *Standby*, and *Response*—allow the model to continuously monitor the input stream and make fine-grained judgments about when to output. Once a re-

*Equal contribution †Corresponding author

The letter `o` in `Streamo` means 'omni', reflecting its multi-task and multi-modal capabilities.

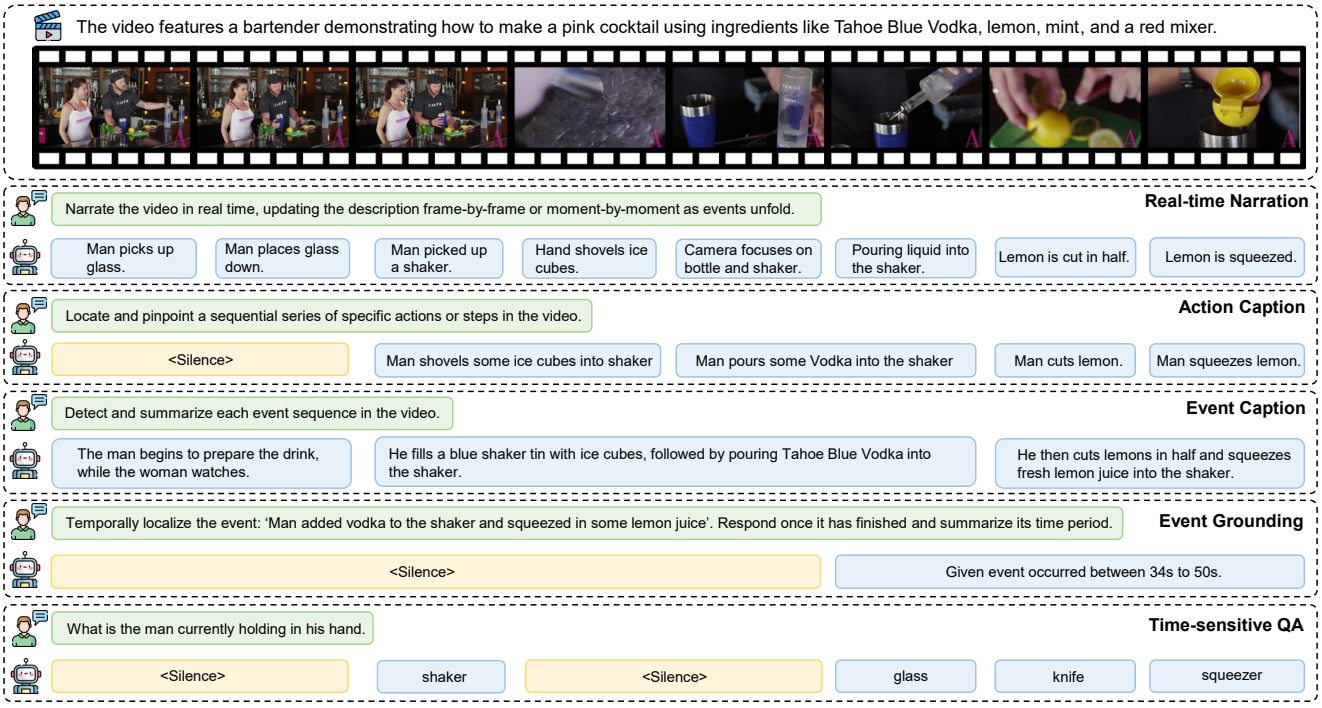

Figure 1. **An example of multi-task annotation in Streamo-Instruct-465K.** Each task is carefully labeled with the corresponding response time boundaries and content, following established annotation standards. The same video is annotated with multiple distinct tasks. The video shown in this example is sourced from ActivityNet [5].

sponse state is triggered, the model immediately produces the corresponding textual output, achieving one-pass inference that significantly improves both the accuracy of response timing and the efficiency of real-time generation.

Training `Streamo` requires high-quality, temporally consistent supervision, yet existing datasets often combine heterogeneous sources with inconsistent annotation standards [13, 15, 16]. These inconsistencies make it difficult for the model to learn precise temporal alignment or multi-task response behaviors. To overcome this problem, we construct **Streamo-Instruct-465K**, a large-scale, multi-task instruction-following dataset designed specifically for streaming video understanding and interaction. The dataset standardizes three levels of response granularity, provides unified temporal annotations for event boundaries, and covers diverse tasks including real-time narration, action and event captioning, temporal grounding, and time-sensitive question answering. Each video is annotated for multiple tasks, providing consistent guidance that strengthens both instruction-following and temporal reasoning. An example of the annotations is shown in Fig. 1.

Extensive experiments demonstrate that our end-to-end training paradigm effectively converts offline models into online streaming assistants. `Streamo` outperforms existing online approaches across both streaming and offline benchmarks, exhibiting strong temporal awareness, accurate frame-level decision-making, and robust multi-task instruction-following. To further support research in this domain, we also introduce a comprehensive streaming benchmark named **Streamo-Bench**, which evaluates instruction understanding across diverse interactive tasks.

Our contributions are threehold: 1) We propose a simple and effective end-to-end training framework that converts offline video models into real-time straeming assistants. 2) We introduce a multi-task instruction tuning dataset with unified temporal annotation and fine-grained response supervision. To our knowledge, this is the largest scale instruction tuning dataset for streaming video understanding and interaction. 3) We establish a comprehensive benchmark for streaming video instruction-following and provide strong baseline models for future research. All research resources including code, models, and datasets will be made publicly available.

2. Related Work

Video Large Language Models The field of vision foundation models [8, 24, 27, 28] has made remarkable progress in recent years, extending capabilities from static image understanding to more general video comprehension. Building on this foundation, numerous advanced video LLMs have emerged. For example, InternVideo2.5 [41] can pro-

cess videos spanning several hours, while Keye-VL-1.5 [45] demonstrates sophisticated reasoning abilities, effectively performing complex thinking process based on video content. A critical limitation, however, is that these state-of-the-art models operate in an offline fashion, requiring the entire video as input before producing any output. This single-pass approach prevents them from handling continuous video streams, as they lack mechanisms to identify the precise temporal moments for generating responses in ongoing streams.

Streaming Video Understanding To tackle real-time interaction, various methods have been proposed in the literature to turn offline video LLMs into online assistants that can identify the appropriate moment to respond in video streams. For instance, Dispider [30] and StreamBridge [36] employ an auxiliary model to segment a video stream into fixed-length clips before feeding them to an offline model. However, this strategy introduces significant computational overhead in both training and inference and often fails to maintain context during multi-turn interactions. On the other hand, VideoLLM-Online [6] and StreamingVLM [44] train the model in a supervised way to directly predict response timing using a special *[EOS]* token. However, this approach is limited to real-time narration and cannot balance between silence and response state. To overcome these problems, we propose an end-to-end training framework along with a multi-task instruction-following dataset specifically designed for streaming video understanding and interaction.

Streaming Video Benchmarks OVO-Bench [21] introduces 12 distinct tasks, incorporating tests for a model's ability to proactively respond. Similarly, STREAMBENCH [43] and SVBENCH [46] concentrate on assessing multi-turn conversational abilities within continuous video contexts. A key limitation, however, is their predominant reliance on question-answer (QA) style setups—typically requiring the model to choose an answer from given options—which does not adequately assess broader instruction-following abilities such as event grounding and captioning. Motivated by the goal that streaming video models should evolve into real-time AI assistants, we introduce Streamo-Bench, a benchmark designed to probe a model's perceptual and responsive capabilities across diverse instructions, moving beyond the constraints of traditional QA-based evaluation.

3. Streamo: Architecture and Training

3.1. Preliminaries

Traditional video understanding models [2, 7] follow an offline paradigm where the complete video V, question Q, and answer A are processed using a single-turn format. Formally, given a video $V = \{v_1, v_2, ..., v_T\}$ of length T and a question Q, the model directly generates an answer A. This approach assumes that the entire video is accessible before inference begins, which is impractical for real-time streaming scenarios where video frames arrive sequentially.

In contrast to offline settings, streaming video understanding processes video content as it arrives in a continuous stream. The model must make decisions based on partial observations $V_{:t} = v_1, v_2, ..., v_t$, where $t \leq T_t$ meaning that the model does not have access to future frames. This temporal constraint requires fundamental changes to both the data structure and training paradigm.

3.2. Data Structure

To simulate streaming scenarios during training, we reformulate the single-turn offline format into a multi-turn dialogue structure. Specifically, a complete video V is temporally segmented into N contiguous segments:

$$V = \{V^{(1)}, V^{(2)}, ..., V^{(N)}\} \tag{1}$$

where $V^{(i)}$ denotes the i-th video segment. Each segment is explicitly annotated with temporal boundaries using special markers, *e.g.*, *<2s-3s>*, to encode temporal information. The multi-turn dialogue is constructed as:

$$\mathcal{D} = \{(V^{(1)}, R^{(1)}), (V^{(2)}, R^{(2)}), ..., (V^{(N)}, R^{(N)})\} \tag{2}$$

where $R^{(i)}$ denotes the response at turn i. Questions and answers are strategically inserted at appropriate turns based on the dataset characteristics and task requirements.

To enable efficient parallel training while maintaining compatibility with standard supervised fine-tuning paradigms, we convert decision process into predictions for the following state tokens:

<Silence>: The model remains silent and continues processing incoming frames.

<Standby>: The model detects relevant video input and waits for complete information.

<Response>: The model receives enough information and will generate a response.

This design empowers the model with frame-level decision-making capabilities while maintaining the next-token prediction framework. As illustrated in Fig. 2, three discrete response states are directly integrated into the normal token prediction process: the model outputs *<Standby>* upon detecting relevant input and *<Response>* when it is ready to answer. A training example is shown in Tab. 1. With this multi-turn dialogue training format, we can simulate realistic streaming video interactions and pose questions at any point in time.

3.3. Training

The multi-turn streaming format introduces severe class imbalance among the three response states. In typical streaming scenarios, *<Silence>* tokens dominate the distribution

Table 1. The format of a multi-turn dialogue.

SYSTEM PROMPT	
USER	*<0s-1s><video>*
ASSISTANT	*<Silence>*
USER	*<1s-2s><video>* Notify me when the light turns green.
ASSISTANT	*<Silence>*
USER	*<2s-3s><video>*
ASSISTANT	*<Silence>*
USER	*<3s-4s><video>*
ASSISTANT	*<Standby>*
USER	*<4s-5s><video>*
ASSISTANT	*<Response>* The light just turned green.

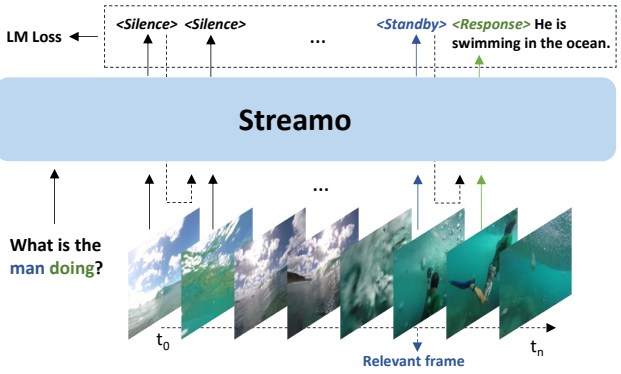

Figure 2. **Streamo's architecture.** Streaming video data is organized into an interleaved, multi-turn dialogue structure that directly integrates a response-state token into the data sequence, enabling end-to-end parallel training.

(often more than 80% of the time), while *<Response>* tokens are sparse. This imbalance biases the model toward remaining silent, making it difficult to learn response timing.

To mitigate this, we apply focal weighting [23] specifically to the three special state tokens. Let $\mathcal{S} = \{s_{\text{silence}}, s_{\text{standby}}, s_{\text{response}}\}$ denote the special token for the three states. For each prediction, we compute a focal weight that emphasizes hard examples:

$$w_{\text{focal}}(x_i) = (1 - p_{c_i})^{\gamma}, \tag{3}$$

where x_i represents the input features at position i, and p_{c_i} is the predicted probability for the true class c_i at position i. $\gamma \geq 0$ is the focusing parameter that controls the rate at which easy examples are down-weighted. To further balance the rare classes, we introduce frequency-based alpha weights. For each special token $k \in \mathcal{S}$ with count n_k in the current batch:

$$\alpha_k = \frac{1}{|\mathcal{S}|} \cdot \frac{\sum_{j \in \mathcal{S}} n_j}{n_k}, \tag{4}$$

where $|\mathcal{S}| = 3$ is the number of special states. This assigns larger weights to less frequent special tokens.

The final loss combines the focal weighting and frequency balancing:

$$\mathcal{L}_i = \begin{cases} \alpha_{t_i} \, w_{\text{focal}}(i) \, \mathcal{L}_{\text{CE}}(i, t_i), & t_i \in \mathcal{S} \\ \mathcal{L}_{\text{CE}}(i, t_i), & \text{otherwise} \end{cases}, \tag{5}$$

The two weighting mechanisms are computed independently and multiplied into the cross-entropy loss. Together, they focus the model on both challenging and infrequent tokens, improving learning of response timing despite severe class imbalance in streaming data. The $\mathcal{L}_{\text{CE}}$ is the standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(i, t_i) = -\log p_{t_i} = \log \sum_{j=1}^{|\mathcal{V}|} e^{z_{i,j}} - z_{i,t_i}, \tag{6}$$

where $z_{i,j}$ is the logit for token j at position i and $|\mathcal{V}|$ is the vocabulary size. This computes the negative log-likelihood of the true token. The total loss averages over all valid (non-masked) positions indicated by $\mathcal{M}$:

$$\mathcal{L}_{\text{total}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathcal{L}_i. \tag{7}$$

This ensures that the loss is not affected by sequence length variations across examples in the batch.

4. Streamo-Instruct-465K

4.1. Data Construction

To provide clear supervision for each round of response decisions, we re-annotated a large-scale training set with detailed temporal boundary labels based on the existing open-source video datasets. We predefined multiple tasks spanning different response granularities, assigning each video several types of task annotations. This approach offers several advantages. First, a unified annotation protocol is applied across datasets, avoiding the inconsistencies and biases that arise when naively mixing datasets with heterogeneous labeling standards. Additionally, each video carries multiple task types with clearly delineated response boundaries, enabling the model to better perceive and understand varying task requirements, develop robust instruction-following capabilities, and execute a range of real-time response tasks. Below, we detail the annotation protocol for each task.

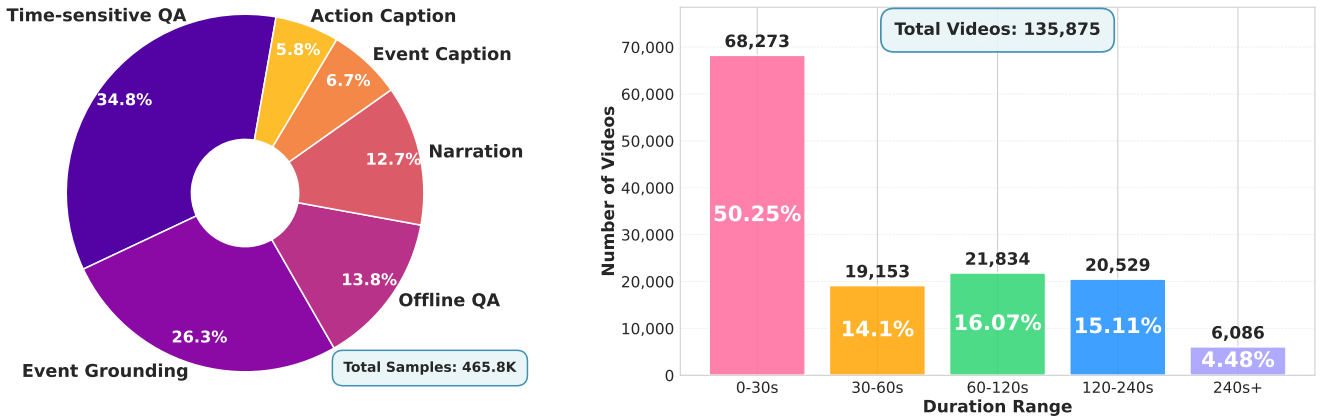

Figure 3. Dataset distribution overview. **Left**: task distribution; **Right**: video duration distribution.

Real-time Narration This task performs real-time commentary over video, requiring second-by-second descriptions that capture fine-grained visual changes. The annotation protocol is: 1) segment each video at one-second intervals; 2) for every adjacent pair of one-second segments (i.e., a two-second window), use Qwen2.5-VL-72B [3] to describe the changes observed between them; 3) concatenate the per-second outputs and send the full narration to GLM-4.5 [47] for post-processing to remove repetitions and redundancies, smooth transitions, and ensure coherent, context-aware narration.

Event Caption This task is similar to standard video captioning but requires the model to detect event boundaries and provide the corresponding caption when an event ends. To construct supervision: 1) generate segment-level captions with the ARC-Hunyuan-Video-7B [14] model; 2) temporally ground each caption using the same model; 3) retain only those videos in which all segment captions have mutually consistent, overlapping time spans that align with the original output. This yields two benefits: it filters out erroneous, noisy data and produces samples with sharper, more explicit event boundaries, enabling clearer supervision.

Action Caption This task mirrors event captioning but narrows the focus from dense events to discrete actions or procedural steps. We reuse the event-caption pipeline and augment it with action-oriented prompts and targeted filtering. This produces cleaner, step-level supervision with sharper action delineation.

Event Grounding The grounding annotation is similar to the offline setup, where each sample pairs an event caption with its corresponding temporal span. The key difference in the online setting is that the caption is provided in advance, and the model must continuously monitor the subsequent video stream to detect the specified event and localize its occurrence in time. We randomly sample captions from the event-caption annotations, rewrite them for grounding, and integrate existing datasets to broaden coverage and improve robustness.

Time-sensitive QA This task targets questions whose correct answers change over time in a dynamic video stream. To construct supervision: 1) process each video with GLM-4.5V [35] model to detect change points across multiple aspects—object attributes (e.g., color, size, state), spatial positions, actions and interactions, counts, and scene or context shifts; 2) generate question–answer pairs from these variations by posing a single, unified question and providing diverse, time-specific answers at the corresponding time points.

4.2. Statistics

Using a unified annotation standard and protocol, we labeled and curated a total of 400K valid samples and additionally merged offline video QA data from the LLaVA-Video [50] dataset, culminating in Streamo-Instruct-465K, and the task distribution is shown on the left of Fig. 3. We integrated multiple open-source video datasets as sources, including Koala [38], LLaVA-Video [50], ActivityNet [5], QVHighlight [29], YouCook2 [53], HACS [51], Ego-TimeQA [10], DiDeMo [1], and COIN [32], yielding 135,875 videos in total. The distribution of video durations is shown on the right of Fig. 3.

5. Experiments

5.1. Models and Datasets

To assess the effectiveness of our training strategy, we adopt Qwen2.5-VL [3] as our base model, across both 3B and 7B model size. Meanwhile, we additionally conduct experiments based on several existing state-of-the-art offline

Table 2. Comparison with state-of-the-art on OVO-Bench. 'Streamo Framework' denotes adapting offline models to the online setting using our training framework. ET-Instruct-3B is trained with ET-Instruct-164K and † indicates LLaVA-Video data is added as offline support. * means the model is trained at 1 fps and evaluated at 2 fps.

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 [37]	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 [50]	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 [19]	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 [37]	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 [9]	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 [31]	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 [49]	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 [6]	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 [30]	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
ViSpeak-7B [12]	1fps	**75.17**	**58.72**	**71.55**	**51.12**	**74.26**	**66.85**	**66.28**	**59.93**	**48.65**	**63.98**	**57.52**	**33.81**	**68.52**	**60.42**	**54.25**	**61.08**
Streamo Framework																	
ET-Instruct-3B [26]	1fps	65.10	35.78	56.90	35.39	24.75	60.87	46.47	41.81	35.14	8.6	28.52	20.06	52.31	67.50	46.62	40.54
ET-Instruct-3B† [26]	1fps	71.14	50.46	67.24	37.08	60.40	60.33	57.78	48.82	48.56	11.29	36.22	13.68	48.62	60.00	40.77	44.92
Streamo-3B	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	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-7B	2fps*	77.18	**66.06**	**76.72**	45.51	**66.34**	**72.83**	**67.44**	**55.56**	**58.11**	**33.87**	**49.18**	**30.84**	**57.55**	82.5	**56.96**	**57.86**

models, including Qwen3-VL [34], and InternVL-3 [54], to demonstrate the compatibility of our framework; these results are presented in the Supplementary material. In addition to training on our proposed Streamo-Instruct-465K dataset, we also compare against ET-Instruct-164K [26], a large-scale instruction-tuning dataset with rich temporal information that has been widely used in prior work to train online video models. To enable a fairer comparison with Streamo-Instruct-465K, we also report results on a mixed dataset comprising ET-Instruct-164K and LLaVA-Video.

5.2. Benchmarks

We evaluated our model across three dimensions of benchmarks: Online, Offline, and Stream Instruction. For the online setting, we adopted OVO-Bench [21], which covers three temporal perception modes, including real-time, backward, and forward, and also spans a total of 12 subtasks. The offline evaluation used standard general video understanding benchmarks, including the short-video benchmarks MVBench [20] and TempCompass [25], as well as the long-video benchmarks VideoMME [11] and LongVideoBench [42], providing a comprehensive assessment of capabilities. In addition, to assess multi-instruction following in an online context, we constructed Streamo-Bench, which includes 300 videos and 3,000 instruction tasks. Each video is paired with tasks of varying temporal scopes and granularities to measure the model's adherence to instructions, providing an important metric for building a reliable real-time AI assistant. Detailed information for Streamo-Bench is given in the Supplementary material.

5.3. Implementation Details

Across all models, we use a unified training setup. Full parameter tuning is applied with the vision encoder frozen, and only the connector and the LLM will be updated. Training runs for a single epoch with a batch size of 512 and a learning rate of $1e$-5. For multi-turn dialogue construction, each video is split into turns of one second, and frames are sampled at 1 fps. The hyperparameter gamma in Eq. (3) is set to 2. In experiments that include LLaVA-Video, we restrict the training data to the same subset used by Streamo-Instruct-465K to ensure a direct and fair comparison.

5.4. Main Results

Comparison with SOTA on Online Video Benchmarks The main results are shown in Tab. 2. Using the Streamo framework, we train the models with ET-Instruct and Streamo-Instruct datasets and compare their performance to currently available open-source offline and online models. The key findings are as follows: **1) Streamo significantly outperforms SOTA.** It is clear that our proposed Streamo-7B exceeds the previous SOTA, Dispider, by +13.83% on average performance. Moreover, we observe that the model trained at 1 fps can be directly evaluated

Table 3. Results on offline video benchmarks. The table compares converted online models with their original offline base models and SOTA models. Numbers in parentheses denote performance differences from the corresponding offline models.

Model	OVO Real-Time	OVO Backward	MVBench	TempCompass	VideoMME	LongVideoBench	Avg
Proprietary Models							
Gemini-1.5-pro [33]	69.3	62.5	60.5	67.1	75.0	64.0	66.4
GPT-4o [17]	64.5	60.8	64.6	70.9	71.9	66.7	66.6
Open-source Online Models							
Flash-VStream-7B [49]	28.4	27.4	61.2	-	61.2	-	-
VideoLLM-online-8B [6]	20.8	17.7	33.9	-	26.9	-	-
Dispider-7B [30]	54.6	36.1	-	-	57.2	-	-
StreamingVLM-7B [44]	62.0	-	69.2	-	65.1	59.0	-
Streamo Framework							
Qwen2.5-VL-3B [3]	54.6	37.8	67.0	64.4	61.5	54.2	56.6
ET-Instruct-3B [26]	46.5 (-8.1)	28.6 (-9.2)	65.8 (-1.2)	60.3 (-4.1)	56.6 (-4.9)	51.2 (-3.0)	51.5 (-5.1)
ET-Instruct-3B† [26]	57.8 (+3.2)	36.2 (-1.6)	68.1 (+1.1)	63.7 (-0.7)	59.6 (-1.9)	54.9 (+0.7)	56.7 (+0.1)
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 [3]	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)

at 2 fps without retraining, achieving an additional +4.66% performance improvement, indicating robust generalization to higher test-time frame rates; **2) Streamo-Instruct-465K dataset surpasses existing dataset.** Compared with the ET-Instruct-164K, our proposed Streamo-Instruct-465K delivers a comprehensive performance advantage, with +7.1% on forward task and +11.79% overall; **3) Offline supervision can hinder online learning.** Augmenting ET-Instruct with the offline LLaVA-Video dataset boosts real-time perceptual accuracy but compromises streaming ability, revealing a trade-off inherent to offline-only supervision. This also demonstrates that Streamo-Instruct-465K transfers effectively to online, streaming scenarios while maintaining strong offline perceptual capability.

Comparison with SOTA on Offline Video Benchmarks To evaluate the general video understanding capability of models after conversion to the online setting, we compare Streamo against the SOTA method and original offline base model on a suite of general offline video benchmarks, with results reported in Tab. 3. The findings show that, after conversion, Streamo retains strong perceptual performance on offline benchmarks across both short-form and long-form videos, surpassing the SOTA, StreamingVLM, in every benchmark. Meanwhile, models trained with our Streamo-Instruct-465K exhibit consistent improvements over base models, with Streamo-7B achieves an average improvement of +3.4% based on Qwen2.5-VL-7B. Holding architecture and training setup constant, Streamo-Instruct-465K also provides a clear advantage over alternative data recipes, outperforming ET-Instruction and LLaVA-Video by +7.8% and +2.5% on average, respectively. These results underscore that our training framework and data not only enable effective transformation of models for streaming video understanding but also preserve and enhance core perceptual capabilities on offline video tasks.

Table 4. Ablation study of loss functions for online training on OVO-Bench Forward Active tasks.

Base Model	Loss Type	REC	SSR	CRR
Qwen2.5-VL-3B	CrossEntropy	6.45	20.99	41.67
Qwen2.5-VL-3B	Loss Scale	18.62	41.02	49.17
Qwen2.5-VL-3B	Focal Loss	27.94	50.72	82.5
InternVL3-2B	CrossEntropy	9.46	20.50	40.42
InternVL3-2B	Loss Scale	21.20	31.47	48.75
InternVL3-2B	Focal Loss	29.23	47.38	80.42

Streamo-Bench To evaluate the model's ability to follow different instructions and perform varied tasks, we assign multiple instruction-driven tasks to a single video, including forward grounding, backward grounding, narration captions, dense captions, and time-sensitive question answering. Details, examples, and statistics for these tasks are presented in the Supplementary material.

As shown in Tab. 5, existing online models show deficiencies in comprehensive multi-task coverage. Our analysis indicates that these shortcomings stem largely from an inadequate ability to comprehend and follow com-

Table 5. Evaluation results on Streamo-Bench. Forward and backward grounding are determined by whether the query refers to a time point before or after the event period, and results are using the mIoU metric. Caption evaluation is conducted by calculating the win rate with Qwen2.5-VL-72B model. TSQA denotes Time-Sensitive QA, i.e., questions whose answers change over time.

Model	Grounding		Caption		TSQA		Average
	Forward	Backward	Narration	Dence Caption	Accuracy	Recall	
Flash-VStream-7B [49]	0	0	23.5	25.9	30.8	13.1	15.6
VideoLLM-online-8B [6]	0	0	42.0	6.6	19.6	7.6	12.6
Dispider-7B [30]	0	8.33	31.6	29.2	14.0	4.4	14.6
StreamingVLM-7B [44]	0	0	68.5	24.0	11.8	43.1	24.6
Streamo-3B	14.7	27.5	71.4	68.5	20.1	65.7	44.7
Streamo-7B	29.4	38.3	75.9	72.8	51.6	63.9	55.3

plex instructions. For instance, removing predefined options leads to widespread failure—as the grounding results show—highlighting a vulnerability to open-ended prompts. Furthermore, in standard QA scenarios, models frequently overlook instructions to update answers as conditions change, which severely degrades recall. We probe instruction comprehension and prompt sensitivity further with additional experiments in the Supplementary material. Collectively, these observations expose a critical gap in current capabilities. In contrast, `Streamo` demonstrates robust performance across tasks, clearly exhibiting strong instruction-following ability. This outcome validates both the diagnostic power of our benchmark and the effectiveness of our method in learning generalized instruction-following capabilities.

5.5. Ablation

To evaluate the effectiveness of our focal loss for training the three decision states, <*Silence*>, <*Standby*>, and <*Response*>, we compare it to standard cross-entropy loss. As shown in Tab. 4, training without state-aware reweighting severely limits performance due to significant class imbalance. In the Streamo-Instruct-465K dataset, the empirical ratio of state labels is approximately <*Silence*>:<*Standby*>:<*Response*> = 12:3:2, which biases conventional training toward predicting Silence and suppresses actual Response predictions.

A straightforward remedy is to assign fixed class weights inversely proportional to label frequency. Specifically, we set the weights to 0.3, 1.3, and 2.0 for silence, standby, and response, respectively, to emphasize response timing. As illustrated in the line "Loss Scale" in Tab. 4), this adjustment effectively mitigates the degradation caused by imbalance. However, fixed weighting fails to capture token-level hardness and sequence-level heterogeneity in decision-state distributions—for instance, narration tasks may contain multiple responses, whereas a QA task might include only one.

Our proposed focal loss addresses this limitation by dynamically reweighting losses based on token-level hardness and per-batch state frequency, thereby providing more adaptive supervision for response-timing decisions. Across both InternVL-3-2B and Qwen2.5-VL-3B backbones, training with the proposed focal loss consistently yields substantial improvements over both the vanilla cross-entropy and fixed-weight baselines.

6. Conclusion

Our work targets the advancement of streaming video by jointly addressing model training and data construction. We introduce an end-to-end training framework together with a large-scale instruction-tuning dataset, Streamo-Instruct-465K, enabling the conversion of multiple state-of-the-art offline models into online version. The resulting model, `Streamo`, not only excels on streaming benchmarks but also rivals top-performing offline models. Furthermore, our proposed Streamo-Bench, which simulates complex multi-instruction scenarios, showcases `Streamo`'s robust multi-tasking capabilities. Collectively, these contributions mark a significant leap towards creating general-purpose, real-time, and interactive AI assistants.

7. Limitations and Future Work

In terms of limitations, while our approach achieves strong accuracy, it is limited by the inherent challenges of streaming video's unbounded temporal context. Our current pipeline lacks specialized long-sequence optimizations, leading to significant memory and latency costs that become prohibitive as sequence length grows.

By leveraging our framework's compatibility with existing techniques, we can integrate KV-cache management and visual token pruning to reduce computational overhead, alongside exploring sliding-window attention and adaptive frame compression for refined context management. Collectively, these strategies are designed to enhance training and inference efficiency, extend the effective context length, and facilitate an unbounded, real-time data stream.

8. Acknowledgement

This research is supported by Hong Kong Research Grants Council Early Career Scheme (No. 22200824).

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 5

[2] 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. 3

[3] 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. 5, 7

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 1

[5] 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, 5

[6] 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. 3, 6, 7, 8

[7] 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. *NeurIPS*, 2024. 3

[8] 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. 2

[9] 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. 6

[10] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. In *CVPR*, 2024. 5

[11] 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:2405.21075*, 2024. 6

[12] 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. 6

[13] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 2

[14] 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. 5

[15] Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv:2312.10300*, 2023. 2

[16] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. *arXiv:2011.11760*, 2020. 2

[17] 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. 7

[18] Asif Ali Laghari, Sana Shahid, Rahul Yadav, Shahid Karim, Awais Khan, Hang Li, and Yin Shoulin. The state of art and review on video streaming. *Journal of High Speed Networks*, 29(3):211–236, 2023. 1

[19] 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. 6

[20] 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 *CVPR*, 2024. 6

[21] Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? *arXiv:2501.05510*, 2025. 3, 6

[22] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023. 1

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023. 2

[25] 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:2403.00476*, 2024. 6

[26] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang W Chen. Et bench: Towards open-ended event-level video-language understanding. *Advances in Neural Information Processing Systems*, 37:32076–32110, 2024. 6, 7

[27] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv:2409.12961*, 2024. 2

[28] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv:2406.09418*, 2024. 2

[29] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23023–23033, 2023. 5

[30] 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. 1, 3, 6, 7, 8

[31] 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. 6

[32] 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. 5

[33] 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. 7

[34] Qwen Team. Qwen3 technical report, 2025. 6

[35] V Team, Wenyi Hong, Wenmeng Yu, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025. 5

[36] Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *arXiv preprint arXiv:2505.05467*, 2025. 1, 3

[37] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[38] 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. 5

[39] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 1

[40] Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. *arXiv:2411.17991*, 2024. 1

[41] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025. 2

[42] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *NeurIPS*, 2024. 6

[43] Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv:2501.13468*, 2025. 3

[44] 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. 3, 7, 8

[45] Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025. 3

[46] Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *arXiv:2502.10810*, 2025. 3

[47] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025. 5

[48] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv:2306.02858*, 2023. 1

[49] 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. 6, 7, 8

[50] 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. 5, 6

[51] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*, 2019. 5

[52] Yucheng Zhao, Chong Luo, Chuanxin Tang, Dongdong Chen, Noel Codella, and Zheng-Jun Zha. Streaming video model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14602–14612, 2023. 1

[53] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 5

[54] 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. 6