OVIBench: Benchmarking Online Video Question Answering under Interruption

Naiming Liu[*1] **Zhiheng Wu**[*§2] **Shuning Wang**[3] **Tie Zhang**[4]
Bowen Liu[5] **Tong Wang**[†2]

[1]HIT, [2]CASIA, [3]ZJU, [4]UESTC, [5]HKUST

[*]Equal contribution. [§]Project leader. [†]Corresponding author: wangtong_change@163.com.

Abstract

Recent vision language models (VLMs) have achieved strong progress in video understanding. However, most existing video QA research and benchmarks still follow an offline, single-round paradigm, overlooking realistic interactions where users may interrupt the model during answer generation. To address this gap, we formulate the task of Online Video Question Answering under Interruption and introduce **OVIBench**, the first standardized benchmark for evaluating VLMs in this setting. OVIBench categorizes interruptions into three types: **Cancellation, False Trigger, Correction** and supports both **open-ended** and **multiple-choice** evaluations. To enable large-scale and reproducible testing, we develop an offline simulation protocol that reproduces interruption during generation under a unified temporal setup, together with a multi-dimensional metric suite for assessing interruption understanding and response generation. Experiments demonstrate that OVIBench effectively distinguishes models' interruption-handling abilities, especially in following correction requests. Finally, we construct a train set **OVI-Train** for interruption-aware fine-tuning. Models fine-tuned on this dataset achieve significant gains on OVIBench, validating the effectiveness of our benchmark and data design. OVIBench, OVI-Train, and the evaluation code are publicly available at https://github.com/9393ming/OVIBench.

1 Introduction

In recent years, multimodal language large models (MLLMs) have made significant progress in video understanding and video question answering (Zhang et al., 2024; Cai et al., 2024), enabling a range of applications such as live-stream interactions (Yang et al., 2025; Zhao et al., 2025), online education (Ray et al., 2025; Ashutosh et al., 2024; Nagarajan and Torresani, 2024), and intelligent assistants (Shu et al., 2025; Yang et al., 2025). However, most existing research and evaluation benchmarks adopt an offline, single-round question-and-answer paradigm (Zhou et al., 2025; Shu et al., 2025; Lin et al., 2024b; Niu et al., 2025). In this setting, the model observes the complete video V and question Q and generates the final answer A in a single pass. This differs substantially from real online interactions (Ma et al., 2025; Zhang et al., 2025b), where users may issue new instructions or interruption signals while the model is generating answers, such as requests to stop or correct the ongoing response. In such cases, the model must not only continue generation but also infer the interruption intent and adjust its response strategy accordingly. The lack of a systematic evaluation framework for this dynamic setting makes it difficult to objectively characterize the capabilities of current MLLMs in interruptible interactions.

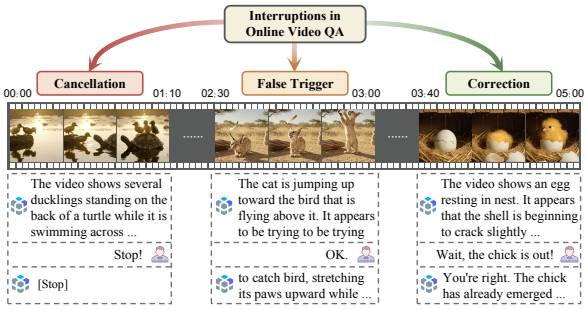


Figure 1: Illustration of three interruption types in online video question answering: Cancellation, False Trigger, and Correction. In Online interactions, users may terminate the response, provide irrelevant signals, or introduce semantic corrections during generation.

To address this gap, we introduce **OVIBench**, the first standardized benchmark for online streaming video QA with interruptions. We formalize the task setting and categorize interruptions into three types: **Cancellation**, **False Trigger**, and **Correction** (Fig. 1). To evaluate interruption handling from both realistic and controllable perspectives, we build a complete data construction pipeline supporting two evaluation formats: **open-ended** and **multiple-choice**. The open-ended

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 16427–16446
October 24-29, 2026 ©2026 Association for Computational Linguistics

setting better reflects free-form real interactions, while the multiple-choice setting enables low-noise and reproducible evaluation for stable comparisons. Moreover, we develop an offline simulation protocol that reproduces "interruption during generation" under a unified temporal setup, enabling large-scale batch evaluation with high reproducibility.

For evaluation, we design a multi-dimensional metric suite spanning two aspects: **interruption understanding** and **response generation**. It measures key capabilities, including interruption-intent recognition, post-interruption intent fulfillment, pre/post-interruption fluency and content consistency, and consistency between the response and the observed video evidence. Experiments on OVIBench show that, while mainstream MLLMs perform well in standard offline video QA, they exhibit substantial weaknesses under online interruptions, particularly in reliably following user correction requests.

Furthermore, to investigate the root causes and explore effective improvements, we construct **OVI-Train**, a multiple-choice training dataset for interruption-aware fine-tuning, and apply targeted fine-tuning to a base model. Compared with Qwen2.5-VL-7B, our 7B model achieves substantial gains, improving multiple-choice accuracy by 16.88% and interruption-type accuracy by 13.94%, and even surpassing the 72B model in the same series. These results validate our data and task design and provide a reusable foundation for future training and evaluation. As the first benchmark dedicated to interruption scenarios in online video question answering, OVIBench pushes large video models beyond offline, static inference toward more realistic online interaction. The following contributions are made in this paper.

- We introduce **OVIBench**, the first benchmark formalizing interruptions in online video QA, supporting both open-ended and multiple-choice evaluation.
- We propose an **offline simulation** protocol for interruptions to enable large-scale, low-noise evaluation, and design a **multi-dimensional metric suite** to comprehensively assess model behavior under dynamic interactions.
- We present **OVI-Train**, a multiple-choice training dataset for interruption scenarios. Experiments validate the effectiveness of our data and task design.

2 Related Work

Video-LLMs and Online Video Understanding. Recently, the capabilities of MLLMs have been extended from static images (Chen et al., 2026) to action reasoning and cross-modal modeling in complex video scenes (Qian et al., 2024; Jiang et al., 2026). This progress has been driven by more sophisticated visual perception modules and staged training strategies (Yu et al., 2026), which have substantially improved models' offline video understanding capabilities (Wang et al., 2026; Maaz et al., 2024; Team et al., 2024; Wu et al., 2026). For example, Video-LLaVA (Lin et al., 2024a) focuses on cross-modal spatio-temporal modeling, Qwen-VL (Wang et al., 2024a) performs well in multi-modal modeling, GPT-4o (Hurst et al., 2024) demonstrates strong long-form text generation capability, and VideoChat (Li et al., 2025) and VideoLLaMA (Zhang et al., 2025a) achieve strong performance in spatio-temporal reasoning. However, in mainstream frameworks, video understanding is usually based on complete observation-holistic generation offline reasoning, where the model generates a single response after receiving the complete video (Wang et al., 2024b). This limits the model's ability to cope with dynamic interruptions and context updates in real interaction environments.

Video Benchmarks. Existing video understanding benchmarks can be broadly categorized into offline and online evaluation settings (Jiang et al., 2025). Video-MME (Fu et al., 2025) evaluates the comprehensive video understanding ability of multimodal models through multiple-choice questions across diverse domains and durations. LVBench (Wang et al., 2025b) focuses on long-video reasoning and assesses whether models can retrieve relevant evidence from extended video contexts. More recently, online benchmarks moved toward streaming inputs and multi-round interactions. StreamingBench (Lin et al., 2024b) evaluates models under streaming video inputs and measures their ability to maintain consistent reasoning as new frames arrive. OVO-Bench (Niu et al., 2025) studies online video understanding with progressive observations and emphasized adaptive reasoning under evolving visual contexts. OVBench (Huang et al., 2025b) further introduces interaction-oriented evaluation protocols to assess model behaviors in online video question answering scenarios. Although these benchmarks covered multiple aspects of video understanding, existing evaluation frameworks ignore the issue of

interruption handling in dynamic interactions.

Interactive Modeling and Turn-Taking Benchmarks. In the research of human-computer speech dialogue, turn-taking is usually regarded as the core mechanism of natural interaction (Lin et al., 2022, 2025; Arora et al., 2025; Zheng et al., 2023). Recent benchmarks such as Talking Turns builds a conversation event prediction and evaluation framework for audio basic models (Arora et al., 2025). Full-Duplex-Bench systematically evaluates full-duplex conversation models from multiple dimensions such as pause processing, backchannel behavior, and interrupt response (Lin et al., 2025). AudioBench extends to a wider range of speech and paralinguistic ability assessments (Wang et al., 2025a). These works emphasize the temporal behavioral evaluation of full-duplex speech dialogue models, but their research objects mainly focus on audio or text modalities. In the video stream question and answer scenario, interruption is not only a problem of judging the timing of speaking, but also a multi-modal decision-making problem based on visual context. So far, there is no benchmark to evaluate this scenario under a unified framework.

3 OVIBench

3.1 Task Formulation

We define Online Video Question Answering under Interruption to evaluate how a model handles user interruptions while answering questions about online video. Online video stream is denoted as $V = \{v_1, v_2, \dots, v_T\}$, where v_t represents the video frame at time step t. At time t_q, a user issues an initial question Q_0, upon which the model begins generating an answer in an autoregressive manner, denoted as $A = \{a_1, a_2, \dots\}$. During generation, at time $t_i > t_q$, the user may provide an interruption signal I. At this moment, the model has produced a partial response $A_{\leq t_i} = \{a_1, \dots, a_{t_i}\}$. Therefore, the observable context at interruption time is the video frames $V_{\leq t_i}$, the initial question Q_0, the partially generated answer $A_{\leq t_i}$, and the interruption signal I. Under this setting, the model is required to perform intent-response coupled modeling. Specifically, it first identify interrupt type $c \in \mathcal{C}$, where $\mathcal{C} = \{\text{False Trigger}, \text{Correction}, \text{Cancellation}\}$. Conditioned on the observable context and interrupt type c, the model then executes the corresponding strategy (e.g., continue, revise, or terminate), producing the post-interruption response $A' = f(V_{\leq t_i}, Q_0, A_{\leq t_i}, I, c)$. Unlike conventional offline VideoQA (Wu et al., 2024; Chen et al., 2024), which can be formulated as a one-shot mapping $A = f(V, Q)$, OVI introduces answer-time external intervention. The model must handle partially generated answers and an evolving interaction state. This makes the task online and state-aware, rather than a static one-shot inference problem.

3.2 Interruption Taxonomy

In online video question answering, interruptions are not merely temporal disturbances (e.g., meaningless insertions), but semantic inputs that may alter the ongoing question-answering process. To characterize model behavior under different interruption conditions, we construct a structured interruption taxonomy. This taxonomy is defined based on two criteria: (1) whether the interruption changes the current semantic context, i.e., whether it introduces new facts, constraints, or objectives; and (2) whether it requires the model to adjust its generation strategy, such as continuing the original response, revising existing content, or terminating the response process. Formally, we define an interruption as an external input occurring during response generation, whose role is to impose constraints on or modify the current interaction state. Based on the degree to which an interruption affects the contextual state and strategy space, we categorize interruptions into three types:

False Trigger. This type does not change the semantic context or the generation objective. The model is expected to recognize the interruption as ineffective and continue the original response. This type evaluates the model‘s ability to resist spurious interventions and maintain generation stability.

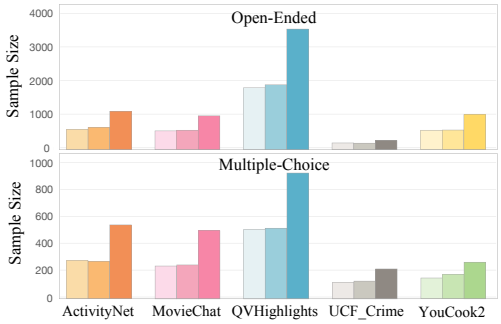
Correction. This type introduces new semantic constraints that modify previously generated content or the question context. The model must update its interaction state and revise the response coherently under the new conditions. This type assesses contextual reconstruction and response-consistent revision.

Cancellation. This type explicitly terminates the current query and invalidates the ongoing response. The model should stop or reset generation upon recognition. This type evaluates responsiveness at the interaction-control level.

3.3 Data Collection

To construct an online video QA dataset with interruptions during generation, we curate 3,200 videos

(a) Data Source Distribution



(b) Benchmark Sample Size Distribution

Figure 2: (a) Distribution of video sources across five public datasets during data collection. (b) Sample size distribution of the benchmark under open-ended and multiple-choice evaluation settings. For each dataset, three bars are shown from left to right, corresponding to the False Trigger, Cancellation, and Correction interruption types.

from five public datasets: ActivityNet (Caba Heilbron et al., 2015), MovieChat (Song et al., 2024), QVHighlights (Lei et al., 2021), UCF-Crime (Qian et al., 2025), and YouCook2 (Ohkawa et al., 2025). These sources cover diverse video understanding scenarios, including open-domain activities, narrative dialogue, highlight localization, anomalous events, and instructional procedures. For each source dataset, videos are randomly sampled without replacement to ensure unbiased coverage of content diversity. We apply basic quality filtering before inclusion: videos with corrupted files, extremely short duration (less than 10 seconds), missing visual content, or severe frame loss are excluded. This collection provides diverse temporal and semantic contexts for constructing False Trigger, Correction, and Cancellation interruptions. The detailed source distribution is shown in Fig. 2(a).

3.4 QA Generation and Data format

As shown in Fig. 3(a), we first use a vision language model (VLM) to generate a complex question Q for each video sample V. By increasing the complexity of the question, we ensure that the model can generate a raw answer A of sufficient length, thus reserving enough processing space for the interruption insert. The video timestamp at which the question is triggered is denoted as T_0. Next, based on the preset interruption type, we use the VLM to generate the corresponding interruption signal I. The interruption time T_1 is randomly generated within the interval (T_0, T_{end}) (T_{end} is the video end time). Finally, each open-ended sample is encapsulated in a standard format: $Data_{open} = \{Q, T_0, I, T_1, V\}$.

In addition, we construct an evaluation set in a multiple-choice format. Based on the data format described above, we introduce candidate options O and the ground truth label GT. The data format is $Data_{choice} = \{Q, T_0, I, T_1, V, O, GT\}$. Options O contain four categories: three categories are correct answers generated by three different interruption types (Cancellation, False Trigger, and Correction), and the remaining one is an incorrect distractor generated by a VLM. The ground truth GT is the candidate that matches the inserted interruption type. To improve robustness, we randomly shuffle the option order for all questions.

Ultimately, we use Qwen2.5-VL-32B (Yao et al., 2024) to generate 13145 samples for $Data_{\text{open}}$ and 4710 samples for $Data_{\text{choice}}$. Since the open-ended format better reflects real-world usage, $Data_{\text{open}}$ more closely matches practical interrupted interactions and is scored by a judge model. In contrast, the multiple-choice set reduces evaluation noise and bias, enabling more accurate and fair comparisons across models. The distributions of the two evaluation sets are shown in Fig. 2(b). Details and examples of the data generation process are provided in the appendix.

3.5 Simulate Online Video Interruption

Since offline batch evaluation is not feasible in real-world dynamic interactions, we simulate the online video interruption process offline. As shown in Fig. 3 (b), the simulation process is as follows:

First, we determine the video content that the model can observe based on the timestamp T_0 of question Q in the data. This step simulates the video stream transmission process, ensuring that the model receives the video segment V_Q that is visible at the given time.

Then, we input the observable video V_Q along with question Q into the model to generate the answer A for question Q. At this point, the model infers based on the currently visible video content

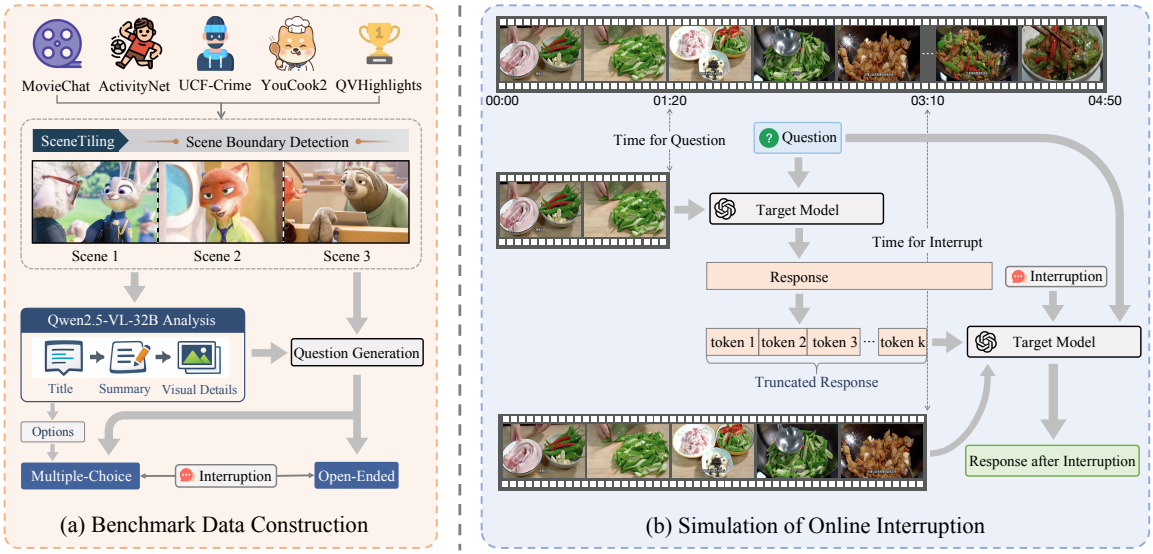


Figure 3: Overview of OVIBench. (a) Benchmark data construction pipeline, including scene segmentation, VLM-based question generation and interruption insertion for both open-ended and multiple-choice formats. (b) Offline simulation of online video interruption, where the model generates responses autoregressively and reacts to interruption signals during generation.

and generates a preliminary answer.

Next, we simulate the interruption process. We set a fixed generation rate for answer A, which allows us to calculate the time at which each token is generated and simulate the time progression of answer A during its generation. Using the interruption timestamp T_1 in the data, we determine the exact time point when the interruption occurs. With T_1, we infer the video content V_I that the model has already observed at this moment, and the position where the generated answer A was truncated.

Finally, we input the interruption signal I, the observable video V_I, and the truncated answer A_I into the model. Based on this information, the model determines the intent of the interruption signal (e.g., whether it's a false trigger, cancellation, or correction) and generates a new response. The model reacts to the interruption, and we evaluate the model's response, measuring its performance and adaptability in handling interruptions.

Through this offline simulation, we can assess the model's ability to answer questions, its responsiveness to interruption signals, and its adaptability under various interruption scenarios.

4 Model-Judged Evaluation of OVIBench

4.1 Comparison Methods

To evaluate the performance of existing models in the online video question answering under interruption scenario, we select several state-of-the-art VLMs for comparison, including Doubao-Seed-1.6 (Huang et al., 2025a), Gemini-2.5-Flash (Comanici et al., 2025), Qwen2.5-VL-7B (Yao et al., 2024), Qwen3-VL-8B (Bai et al., 2025), VideoChat-R1-7B (Li et al., 2025), VideoLLaMA3-7B (Zhang et al., 2025a), and Qwen3-VL-30B-A3B (Bai et al., 2025). These models represent different architectures and application domains of current vision-language models. By comparing them, we can comprehensively assess the strengths and weaknesses of existing models in interruption scenarios and reveal the performance gaps in real-time online video question answering.

4.2 Metrics Design

In online video QA, interrupt signals may occur at any time during answer generation. Handling such interruptions requires models not only to produce correct content but also to recognize user intent. In addition, the model needs to adjust its response logic to be consistent with visual and contextual information. The above capabilities cannot be measured by a single indicator. Therefore, beyond intent recognition accuracy, we introduce six complementary metrics to assess the model from multiple perspectives. All metrics are automatically scored by a judge model based on predefined criteria.

Intent-Action Consistency (IAC). IAC evaluates whether the model's post-interruption response strategy aligns with the interruption intent it has identified in online video interaction. For example,

if the model interprets the user‘s intent as Cancellation but continues generating content or instead performs a correction, the response is considered inconsistent.

Textual Fluency Score (TFS). TFS is used to evaluate whether the model‘s post-interruption responses can smoothly continue from the pre-interruption responses at the textual dimension.

Visual Evidence Score (VES). VES is used to evaluate whether the visual content mentioned in the model's post-interruption response is consistent with the observed video evidence.

Intent Fulfillment Accuracy (IFA). IFA is used to evaluate whether the model‘s post-interruption response fulfills the user‘s intent. Even if the model misidentifies the intent, the response is still considered correct as long as it successfully satisfies the user‘s actual intent.

Correction Following Score (CFS). CFS measures the extent to which, after a user issues a correction-type interruption, the model‘s post-interruption response incorporates and executes the requested corrections. For example, replacing incorrect information, filling in missing elements, or removing content that should not appear.

Contextual Consistency Score (CCS). CCS measures whether the post-interruption responses generated by the model are consistent with the existing context at the content dimension.

4.3 Judge Model

In the open-ended benchmark, we use a model-judged approach to automatically score the model's response (Liu et al., 2023; Chehbouni et al., 2025). For each metric, we designed a corresponding Judge Prompt (See appendix for details). IAC and IFA are computed using a binary satisfaction criterion (0/1), while TFS, VES, CFS, and CCS use a 1–10 scoring, with higher values indicating better performance. To select a stable Judge model with higher consistency with human reviewers, 415 samples are selected from the open-ended subset of OVIBench to construct a validation set. Qwen3-VL-8B, Qwen3-VL-30B-A3B, Qwen3-VL-235B-A22B (Bai et al., 2025), Doubao-Seed-1.6, Gemini-2.5-Flash, and GPT-4.1-mini (Achiam et al., 2023) are evaluated as Judge model. We measure judge performance using the mean absolute error (MAE) between the judge scores and human annotations (Table 1). Experiments reveal that Qwen3-VL-235B-A22B demonstrated the best consistency with human reviewers for metrics IAC, TFS, IFA, CFS, and CCS; while Qwen3-VL-30B-A3B performed best for VES. Therefore, we ultimately adopt Qwen3-VL-235B-A22B as the Judge Model for all metrics except VES, and specifically use Qwen3-VL-30B-A3B as the Judge Model for VES.

4.4 Results and Analysis

Table 2 reports the results on the open-ended subset of OVIBench. Overall, existing models show limited capability in handling interruptions in online video question answering, and their performance varies substantially across metrics. Most models obtain low scores on CFS and CCS, indicating persistent challenges in incorporating and executing user-requested corrections and maintaining content-level consistency with the existing context. Doubao-Seed-1.6 achieves the best overall results, ranking highest in DAC, TFS, CFS, CCS, and overall accuracy. Qwen3-VL-8B achieves the best VES and IFA, suggesting strong video–response consistency and effective intent fulfillment. However, all models still exhibit substantial limitations in following user correction instructions and revising their responses accordingly. These results indicate that interruption handling remains challenging for current MLLMs.

Fig. 4 shows the interruption-type classification accuracy (CA) for different interruption categories. For Cancellation, all models achieve nearly perfect accuracy, reaching close to 100%. This result indicates that explicit termination signals are easy to recognize. In contrast, False Trigger exhibits the largest performance variation across models. Smaller models show noticeably lower accuracy, indicating that distinguishing irrelevant interruptions from meaningful ones remains challenging. Correction lies between Cancellation and False Trigger in difficulty. Most models achieve accuracy around 87%–92%, suggesting that corrective intent is recognizable but still more ambiguous than explicit termination signals.

5 Multiple-Choice Evaluation of OVIBench

5.1 Interruption-aware Fine-tuning

Training Dataset. Overall, the suboptimal performance of MLLMs can be largely attributed to the lack of explicit interruption-type supervision in instruction-tuning data. To address this issue, we construct 20765 training instances (OVI-Train) following the data generation pipeline in

Table 1: MAE between different Judge models and human annotations on the validation set, and lower is better. The best value in each column is highlighted in bold.

Model	MAE_{IAC}	MAE_{TFS}	MAE_{IFA}	MAE_{CCS}	MAE_{CFS}	MAE_{VES}
Qwen3-VL-8B (Bai et al., 2025)	0.2070	1.1445	0.1641	3.1289	2.0703	1.6016
Qwen3-VL-30B-A3B (Bai et al., 2025)	0.1406	0.8984	0.1250	3.1602	2.3359	**0.8477**
Qwen3-VL-235B-A22B (Bai et al., 2025)	**0.0508**	**0.3672**	**0.0625**	**0.9453**	**0.5547**	1.0273
Doubao-Seed-1.6 (Huang et al., 2025a)	0.0859	1.0859	0.1484	2.6016	1.9766	2.1445
Gemini-2.5-Flash (Comanici et al., 2025)	0.0938	1.2656	0.1406	2.8984	1.8711	-
GPT-4.1-mini (Achiam et al., 2023)	0.0938	0.7500	0.1445	3.0938	1.6484	-

Table 2: Experimental results on the open-ended subset of OVIBench. CA represents the accuracy of interruption-type classification. IAC and IFA are binary classification accuracy. TFS, VES, CFS, and CCS are scored on a 1-10 scale. Higher values indicate better performance (↑). The best results in each column are highlighted in bold.

Model/Metric	IAC ↑	TFS ↑	VES ↑	IFA ↑	CFS ↑	CCS ↑	CA ↑
Doubao-Seed-1.6 (Huang et al., 2025a)	**95.73%**	**9.61**	6.23	79.42%	**5.24**	**7.26**	**92.85%**
Gemini-2.5-Flash (Comanici et al., 2025)	76.45%	8.41	6.28	64.58%	4.42	6.97	73.39%
Qwen2.5-VL-7B (Yao et al., 2024)	64.14%	7.55	5.34	50.11%	3.73	5.06	83.64%
Qwen3-VL-8B (Bai et al., 2025)	93.63%	9.32	**6.68**	**81.07%**	4.91	6.75	88.61%
VideoChat-R1-7B (Li et al., 2025)	66.42%	7.69	5.36	51.03%	3.86	4.98	87.27%
VideoLLaMA3-7B (Zhang et al., 2025a)	72.33%	7.50	5.25	57.92%	4.06	5.63	89.61%
Qwen3-VL-30B-A3B (Bai et al., 2025)	88.29%	9.05	6.50	80.23%	5.10	6.47	91.10%

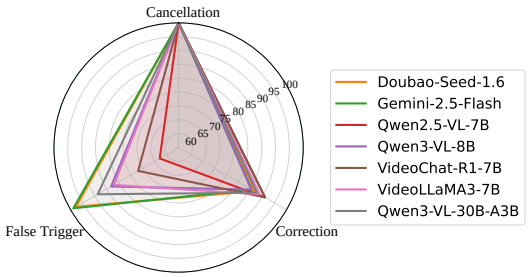


Figure 4: Comparison of model performance across different interruption types on OVIBench. Each axis represents the recognition accuracy (%) for one interruption type, and larger areas indicate better overall interruption understanding capability.

Table 3: Statistics of OVI-Train instances by video source and interruption type.

	Cancellation	Correction	False Trigger	Total
ActivityNet	840	1616	799	3255
MovieChat	475	913	477	1865
QVHighlights	2736	5365	2744	10845
UCF_Crime	173	297	140	610
YouCook2	1041	2050	1099	4190

Sec. 3.4. Specifically, we first derive a target answer A for each question Q in the dataset, where an interruption is intended to occur during generation. Then, based on the offline simulation procedure in Sec. 3.5, we truncate A at the simulated interruption point and inject an interruption instruction, yielding training samples of the form $Data_{train} = \{Q, A_I, I, V_I, O, GT\}$. Here, A_I is the truncated answer segment shown before the interruption, I is the interruption signal/instruction, V_I denotes the video segment at the interruption moment, and O is the set of candidate options. During training, we feed $\{Q, A_I, I, V_I, O\}$ as model inputs and use GT as the supervision label, enabling the model to accurately recognize and select the correct interruption type in online video interactions. The data distribution is shown in Table 3.

Training Details. We perform LoRA-based fine-tuning (Dettmers et al., 2023) on the Qwen2.5-VL-7B base model using the OVI-Train dataset. The LoRA rank r and scaling factor α are set to 16 and 32. The training process uses the AdamW optimizer with an initial learning rate of 1×10^{-4}, together with a cosine learning rate schedule and a 3% linear warmup phase. The fine-tuning is conducted on 8 NVIDIA H20 GPUs with a global batch size of 128, trained for 10 epochs.

Table 4: Multiple-choice evaluation results on OVIBench. MA denotes answer selection accuracy, and CA denotes interruption-type classification accuracy. Δ indicates the performance improvement of our model over the baseline. The best results in each column are highlighted in bold.

Model	All		False Trigger		Cancellation		Correction	
	MA	CA	MA	CA	MA	CA	MA	CA
Open-Source Models								
Qwen2.5-VL-7B (Yao et al., 2024)	65.54	81.39	43.46	98.69	45.69	42.74	86.99	92.79
Qwen2.5-VL-72B (Yao et al., 2024)	71.46	93.36	95.37	**99.16**	30.35	**98.49**	65.93	77.24
VideoChat-R1-7B (Li et al., 2025)	71.92	83.67	56.54	98.60	52.87	53.55	89.58	91.82
Qwen3-VL-30B-A3B (Bai et al., 2025)	67.75	84.47	15.62	66.93	65.03	73.65	95.43	98.95
Closed-Source Models								
Doubao-Seed-1.6 (Huang et al., 2025a)	66.34	93.01	30.28	98.25	43.50	77.62	96.40	98.37
Gemini-2.5-Flash (Comanici et al., 2025)	52.94	63.71	56.63	63.61	17.40	50.76	69.58	70.51
Ours (7B)	**82.42**	**95.33**	**96.40**	98.81	**77.24**	91.82	**98.69**	**98.83**
Δ (vs Qwen2.5-VL-7B)	+16.88	+13.94	+52.94	+0.12	+31.55	+49.08	+11.70	+6.04

5.2 Metrics

To measure a model‘s interruption handling ability during generation under reproducible, low-noise conditions, we evaluate the multiple-choice subset of OVIBench using two core metrics. We report (1) Multiple-Choice Question Accuracy (MA) and (2) Interruption-Type Classification Accuracy (CA). This multiple-choice setting complements the open-ended evaluation by reducing subjectivity via discrete options while explicitly measuring intent recognition under interruption. MA measures whether the model selects the correct answer option from the candidates under online video interruption. CA measures whether model correctly identifies the interruption intent (i.e., the interruption type).

5.3 Results and Analysis

To evaluate the effectiveness of the proposed fine-tuning under interruption scenarios, we conduct a multiple-choice evaluation on Qwen2.5-VL-7B. We compare it with several vision-language models, including Doubao-Seed-1.6, Gemini-2.5-Flash, Qwen2.5-VL-72B, Qwen3-VL-30B-A3B, and VideoChat-R1-7B.

As shown in Table 4, the Qwen2.5-VL-7B baseline shows limited performance, especially on MA under False Trigger and Cancellation. Its overall MA and CA reach only 65.54% and 81.39%, indicating limited capability in handling interruption scenarios. After fine-tuning, our 7B model achieves the best overall results, reaching 82.42% MA and 95.33% CA on the full set. The largest gains appear in False Trigger MA (+52.94%) and Cancellation CA (+49.08%). These improvements demonstrate the effectiveness of the OVI-Train dataset for interruption-aware training. Notably, the fine-tuned 7B model surpasses larger models such as Qwen2.5-VL-72B and also outperforms video-specialized models such as VideoChat-R1-7B on several metrics. This result indicates that targeted interruption-aware fine-tuning can significantly enhance online video question answering capability even with a smaller model.

6 Conclusion

This paper studies online video question answering, where users may interrupt the model during response generation. To analyze this scenario, we introduce the Online Video Question Answering under the Interruption task and construct the benchmark OVIBench, which models three interruption types and supports both open-ended and multiple-choice evaluation. We simulate interruptions offline to enable large-scale, reproducible evaluation, and design six complementary metrics to comprehensively assess model performance. Experiments show that existing MLLMs still struggle under interruptions. To address this, we introduce the OVI-Train dataset for interruption-aware fine-tuning and demonstrate that specialized training significantly improves model responsiveness. Overall, this study provides a practical framework to evaluate and improve video QA models in real-world scenarios where user interruptions may occur.

7 Limitations

Although OVIBench provides a systematic and reproducible framework for evaluating interruption handling in online video question answering, several directions remain for further extension. We simulate user interruptions during model generation through an offline protocol, which enables fair comparison across models, but real online systems may involve more dynamic generation speed, network latency, and user input timing. In addition, this work focuses on three representative interruption types, namely Cancellation, False Trigger, and Correction, while more complex scenarios such as multi-turn follow-up questions, task switching, and mixed intentions can be explored in future work. Moreover, OVIBench relies on VLMs to generate questions, interruption signals, and candidate answers, which supports scalable and consistent data construction, while incorporating more human annotations or real user interaction data could further improve its naturalness and coverage. Finally, although we evaluate multiple mainstream VLMs and validate the effectiveness of OVI-Train, its generalization to broader model architectures, languages, and real-world interactive settings involving speech or multi-user inputs remains an important direction for future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Siddhant Arora, Zhiyun Lu, Chung-Cheng Chiu, Ruoming Pang, and Shinji Watanabe. 2025. Talking turns: Benchmarking audio foundation models on turn-taking dynamics. *arXiv preprint arXiv:2503.01174*.

Kumar Ashutosh, Zihui Xue, Tushar Nagarajan, and Kristen Grauman. 2024. Detours for navigating instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18804–18815.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. 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.

Chen Cai, Zheng Wang, Jianjun Gao, Wenyang Liu, Ye Lu, Runzhong Zhang, and Kim-Hui Yap. 2024. Empowering large language model for continual video question answering with collaborative prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3921–3932.

Khaoula Chehbouni, Mohammed Haddou, Jackie Chi Kit Cheung, and Golnoosh Farnadi. 2025. Neither valid nor reliable? investigating the use of llms as judges. *arXiv preprint arXiv:2508.18076*.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024. 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.

Zhuohong Chen, Zhenxian Wu, Yunyao Yu, Hangrui Xu, Zirui Liao, Zhifang Liu, Xiangwen Deng, Pen Jiao, and Haoqian Wang. 2026. Learning to search: A decision-based agent for knowledge-based visual question answering. *arXiv preprint arXiv:2604.07146*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118.

Yizhe Huang, Yang Liu, Ruiyu Zhao, Xiaolong Zhong, Xingming Yue, and Ling Jiang. 2025a. Memorb: A plug-and-play verbal-reinforcement memory layer for e-commerce customer service. *arXiv preprint arXiv:2509.18713*.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. 2025b. Online video understanding: Ovbench and videochat-online. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3328–3338.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Tianxiang Jiang, Linquan Wu, Sheng Xia, Songze Li, Ziang Yan, Haoyu Yang, Yu Qiao, and Yi Wang. 2026. Imagine before you predict: Interleaved latent visual reasoning for video event prediction. *arXiv preprint arXiv:2606.05769*.

Tianxiang Jiang, Sheng Xia, Yicheng Xu, Linquan Wu, Xiangyu Zeng, Limin Wang, Yu Qiao, and Yi Wang. 2025. Vknowu: Evaluating visual knowledge understanding in multimodal llms. *arXiv preprint arXiv:2511.20272*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2025. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024a. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 5971–5984.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. 2025. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024b. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*.

Ting-En Lin, Yuchuan Wu, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3299–3308.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2025. Language model can listen while speaking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24831–24839.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602.

Tushar Nagarajan and Lorenzo Torresani. 2024. Step differences in instructional video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18740–18750.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, and 1 others. 2025. Ovobench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913.

Takehiko Ohkawa, Takuma Yagi, Taichi Nishimura, Ryosuke Furuta, Atsushi Hashimoto, Yoshitaka Ushiku, and Yoichi Sato. 2025. Exo2egodvc: Dense video captioning of egocentric procedural activities using web instructional videos. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8324–8335. IEEE.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. 2024. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360.

Yuanbin Qian, Shuhan Ye, Chong Wang, Xiaojie Cai, Jiangbo Qian, and Jiafei Wu. 2025. Ucf-crime-dvs: A novel event-based dataset for video anomaly detection with spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6577–6585.

Sourjyadip Ray, Shubham Sharma, Somak Aditya, and Pawan Goyal. 2025. Eduvidqa: Generating and evaluating long-form answers to student questions based on lecture videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34689–34715.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. 2025. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. 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.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy Chen. 2025a. Audiobench: A universal benchmark for audio large language models. 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 4297–4316.

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

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

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, and 1 others. 2025b. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

Y Wang, K Li, X Li, J Yu, Y He, G Chen, B Pei, R Zheng, J Xu, Z Wang, and 1 others. 2024b. Internvideo2: Scaling video foundation models for multimodal video understanding. arxiv 2024. *arXiv preprint arXiv:2403.15377*, 2.

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

Zhiheng Wu, Tong Wang, Shuning Wang, Naiming Liu, and Yumeng Zhang. 2026. See further, think deeper: Advancing vlm's reasoning ability with low-level visual cues and reflection. *arXiv preprint arXiv:2604.24339*.

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

Song Yao, Chunli Lv, Kun Zhu, and Xiaobin Qiu. 2024. Fine-tuning the qwen2. 5-vl model for intelligent applications in the electrical domain. *EAI Endorsed Transactions on Energy Web*, 12.

Yunyao Yu, Zhengxian Wu, Zhuohong Chen, Hangrui Xu, Zirui Liao, Xiangwen Deng, Zhifang Liu, Senyuan Shi, and Haoqian Wang. 2026. Stabilizing unsupervised self-evolution of mllms via continuous softened retracing resampling. *arXiv preprint arXiv:2604.03647*.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, and 1 others. 2025a. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2024. A simple llm framework for long-range video question-answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21715–21737.

Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chao-Hong Tan, Zhihao Du, and 1 others. 2025b. Omniflatten: An end-to-end gpt model for seamless voice conversation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14570–14580.

Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N Metaxas, and Licheng Yu. 2025. Accelerating multimodal large language models by searching optimal vision token reduction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29869–29879.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, and 1 others. 2025. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13691–13701.

A Prompt Templates

Intent-Action Consistency asks the judge model to determine whether the model's post-interruption reaction is logically consistent with its predicted interruption type. All placeholders are filled with sample-specific information during inference.

IAC Evaluation Prompt

Role. You are a professional dialogue system interaction evaluation expert. You are assessing a vision-language model's **Intent-Action Consistency** in real-time video stream interaction.

Interaction Background.
Original question: `<question>` (timestamp: `<q_ts>`s)
Response before interruption: `<truncated_text>`
User interruption signal: `<interruption_signal>` (timestamp: `<i_ts>`s)

Model Behavior.
Predicted interruption type: `<pred_type>`
Model's interruption reaction: `<predicted_reaction>`

Interruption Type Definitions.
False Trigger = mistaken interruption;
Cancellation = terminate;
Correction = revise.

Evaluation Objective.
Determine whether the model's interruption reaction is consistent with its predicted interruption type.

Consistency Criteria.

- **False Trigger**: The model should continue the original response path and should *not* revise or terminate.
- **Correction**: The model should revise or reconstruct the original semantics, and should *not* simply continue or directly terminate.
- **Cancellation**: The model should immediately terminate the task and should *not* continue answering.

Output Requirement.

1. **Logical Analysis**: Briefly explain whether the reaction is logically consistent.
2. **Final Score**: If logically consistent, output `Score: [1]`; otherwise, output `Score: [0]`.

Textual Fluency Score asks the judge model to assess whether the model performs a natural and stable strategy transition after being interrupted, rather than exhibiting abrupt shifts, semantic discontinuities, or unnecessary repetition. All placeholders are filled with sample-specific information during inference.

TFS Evaluation Prompt

Role. You are a professional dialogue generation trajectory evaluation expert. You are assessing the **Textual Fluency Score** of a vision-language model when it performs a strategy shift after being interrupted by a user during a real-time video stream question-answering task.

TFS Evaluation Prompt (continued)

Metric Definition.
The **Textual Fluency Score** metric evaluates whether the model achieves a natural and stable transition before and after the interruption, rather than exhibiting abrupt shifts, semantic discontinuities, or unnecessary repetition.

Interaction Background.
Original question: `<question>` (timestamp: `<q_ts>`s)
Response before interruption: `<truncated_text>`
User interruption signal: `<interruption_signal>` (timestamp: `<i_ts>`s)

Model Behavior.
Predicted interruption type: `<pred_type>`
Model's interruption reaction: `<predicted_reaction>`

Interruption Type Definitions.
False Trigger = mistaken interruption;
Cancellation = terminate;
Correction = revise.

Evaluation Dimensions.

1. **Boundary Continuity**: Is the transition from the last sentence before interruption to the first sentence after interruption natural? Is there any sudden topic shift, semantic jump, or syntactic break?
2. **Controlled Strategy Shift**: If False Trigger, does the model naturally continue along the original path? If Correction, does the model smoothly shift toward revision rather than abruptly overturning prior content? If Cancellation, does the model terminate cleanly rather than continuing with redundant output?
3. **Unnecessary Discontinuity Detection**: Is there repetition, disordered syntax, or logical collapse? Is there obvious distributional collapse, such as suddenly generating completely irrelevant content?

Scoring Criteria (1–10). Please assign a score from 1 to 10 based on transition smoothness, boundary continuity, strategy correctness, and the degree of unnecessary discontinuity. The detailed score-to-criterion mapping is summarized in Table 5.

Output Requirement.

- **Logical Analysis**: Specify which scoring rule(s) were triggered and how they apply.
- **Final Score**: `Score: [number]`

Visual Evidence Score (VES) asks the judge model to verify whether the model's post-interruption reaction is visually grounded in the corresponding video segment, with particular attention to visual authenticity, temporal synchronization, and hallucination detection. All placeholders are filled with sample-specific information during inference.

VES Evaluation Prompt

Role. You are a senior video content auditing expert. You are evaluating the **Visual Evidence Score** of a vision-language model during real-time video stream interaction.

VES Evaluation Prompt (continued)

Interaction Background.
Original question: <question> (timestamp: <q_ts>s)
Response before interruption: <truncated_text>
User interruption signal: <interruption_signal> (timestamp: <i_ts>s)

Model Behavior.
Predicted interruption type: <pred_type>
Model's interruption reaction: <predicted_reaction>

Core Evaluation Dimensions.
You must not only examine the text, but also trace back to the visual frames between <q_ts>s and <i_ts>s to verify the following aspects:

1. **Visual Authenticity**: Do the objects, actions, colors, positions, and other details mentioned by the model actually appear in the video?
2. **Timestamp Synchronization**: The interruption occurs at <i_ts>s. Does the model's reaction at that moment accurately reflect the visual state at that second, or within the preceding one or two seconds?
3. **Hallucination Detection**: Did the model fabricate events or visual details that do not exist in the video in order to forcibly match the user's interruption signal?

Scoring Criteria (1–10).
Please assign a score from 1 to 10 based on visual grounding accuracy, timestamp synchronization, and the degree of hallucination. The detailed score-to-criterion mapping is summarized in Table 6.

Score Cap Conditions.
Certain severe visual grounding errors impose upper bounds on the final score. The detailed cap conditions are summarized in Table 9.

Evaluation Task.

- **Video Detail Verification**: What specific visual elements did the model mention? Which are genuinely present? Which are not? Which scoring rule(s) were triggered and how?
- **Final Score**: `Score: [number]`

Intent Fulfillment Accuracy (IFA) asks the judge model to infer which implicit strategy the model adopted after being interrupted, and then determine whether this inferred strategy aligns with the target strategy implied by the ground-truth interruption type. All placeholders are filled with sample-specific information during inference.

IFA Evaluation Prompt

Role. You are an interruption strategy identification evaluation expert. Your task is to determine which implicit strategy the model adopted after being interrupted, and compare it with the target strategy.

Interaction Background.
Original question: <question> (timestamp: <q_ts>s)
Response before interruption: <truncated_text>
User interruption signal: <interruption_signal> (timestamp: <i_ts>s)
GT interruption type: <gt_type>

IFA Evaluation Prompt (continued)

Model Reaction After Interruption.
<predicted_reaction>

Definitions of the Three Strategy Behaviors.

1. **Continuation (continue original path)**
   - Continues directly along the original response logic.
   - Does not retract previous content.
2. **Revision (semantic correction)**
   - Explicitly retracts or negates the previous conclusion.
   - Rewrites or corrects content under new constraints.
3. **Stop (termination)**
   - Explicitly stops the task.
   - Outputs `[STOP]` or an equivalent termination expression.

Evaluation Procedure.

1. Determine which category the model's actual behavior belongs to.
2. Compare it with the target strategy corresponding to the GT interruption type:
   - **False Trigger** ⇒ **Continuation** (continue the original response)
   - **Correction** ⇒ **Revision** (rewrite under new constraints)
   - **Cancellation** ⇒ **Stop** (terminate the task, output `[STOP]`)
3. Assign the final score.

Special Rules.

- If the model first continues the old conclusion and then performs correction, count it as incorrect.
- If the model does not explicitly stop and continues generating content, it does *not* qualify as **Stop**.
- If the behavior is ambiguous or mixes two strategies, count it as incorrect.

Output Requirement.

- **Logical Analysis**: Clearly explain the inferred strategy and the reasoning, and explicitly state whether it aligns with the target strategy.
- **Final Score**: Output `Score: [1]` if consistent; otherwise output `Score: [0]`.

Correction Following Score (CFS) asks the judge model to assess whether, under a **Correction** scenario, the model truly performs semantic retraction and structural reconstruction, rather than merely applying superficial paraphrasing. All placeholders are filled with sample-specific information during inference.

CFS Evaluation Prompt

Role. You are a professional semantic revision quality evaluation expert. You only evaluate **Correction** scenarios, assessing whether the model truly performs semantic retraction and structural reconstruction, rather than superficial paraphrasing.

Scoring Objective.
If the GT interruption type is *not* **Correction**, directly output `Score: [0]`. If the GT interruption type is **Correction**, determine whether the model genuinely retracts the old conflicting semantics and reconstructs the response under the new constraints.

Interaction Background.
Original question: `<question>` (timestamp: `<q_ts>`s)
Response before interruption: `<truncated_text>`
User interruption signal: `<interruption_signal>` (timestamp: `<i_ts>`s)

Model Behavior.
Predicted interruption type: `<pred_type>`
Model's interruption reaction: `<predicted_reaction>`

Evaluation Procedure (must follow in order).

1. **Identify the conflict points in the old semantics.**
   - Which statements contradict the user's correction signal?
   - Which conclusions must be retracted?
2. **Check whether the model:**
   - Explicitly retracts or negates the conflicting content
   - Reconstructs the semantics under the new constraints
   - Avoids continuing to use the old conclusions

Scoring Criteria (1–10).
Please assign a score from 1 to 10 according to the detailed semantic revision rubric summarized in Table 7.

Score Cap Rules.
Certain severe semantic revision failures impose upper bounds on the final score. The detailed cap rules are summarized in Table 10.

Output Requirement.

- **Logical Analysis**: Identify the old semantic conflict points, evaluate whether the new constraints are satisfied, determine whether old conflicts remain, and specify which scoring rule(s) were triggered and how.
- **Final Score**: `Score: [number]`

Contextual Consistency Score (CCS) asks the judge model to assess whether the model preserves valid and unaffected context after an interruption, while still performing any necessary modification or termination. All placeholders are filled with sample-specific information during inference.

CCS Evaluation Prompt

Role. You are a context preservation capability evaluation expert. Your task is to assess whether the model preserves unaffected valid context after an interruption, while also performing necessary modification or termination.

Interaction Background.
Original question: `<question>` (timestamp: `<q_ts>`s)
Response before interruption: `<truncated_text>`
User interruption signal: `<interruption_signal>` (timestamp: `<i_ts>`s)

Model Behavior.
Predicted interruption type: `<pred_type>`
Model's interruption reaction: `<predicted_reaction>`

Scoring Focus.

1. **Preservation**: Are key information or context units that should remain unaffected by the interruption preserved, such as characters, events or goals?
2. **Precise Modification**: Were the necessary parts correctly modified, especially in **Correction**, while unaffected parts were not unnecessarily altered?

Evaluation Procedure (must follow in order).

1. **Identify the semantic units that should be preserved.**
   - Established characters or objects
   - Previously described facts
   - Context information unrelated to the correction
2. **Examine the post-interruption reaction.**
   - Which semantic units were preserved?
   - Which were unnecessarily removed?
   - Is there evidence of complete overturning and rewriting?
3. **Determine whether any of the following occur.**
   - Unnecessary large-scale rewriting
   - Complete replacement of the original context

Scoring Criteria (1–10).
Please assign a score from 1 to 10 according to the detailed context preservation rubric summarized in Table 8.

Score Cap Rules.
Certain severe context preservation failures impose upper bounds on the final score. The detailed cap rules are summarized in Table 11.

Output Requirement.

- **Logical Analysis**: List the semantic units that should be preserved, specify which were preserved or removed, and indicate which scoring rule(s) were triggered and how.
- **Final Score**: `Score: [number]`

Table 5: Detailed scoring rubric for Textual Fluency Score (TFS).

Score	Criterion
10	Immediately enters the correct strategy; no repetition, filler transition, or semantic jump; reads like natural human real-time dialogue.
9	Correct strategy executed immediately; may contain a very short transition word; no repetition or semantic discontinuity.
8	Strategy is correct; may contain one explicit transition sentence; no semantic confusion; no repetition exceeding one sentence.
7	Strategy is correct, with only minor repetition or slightly rigid structure.
6	Strategy is correct, but repeats original content for more than one sentence or briefly becomes illogical before recovering.
5	Eventually shifts to the correct strategy, but contains noticeable discontinuity or delayed revision.
4	Strategy execution is unstable, with large repetition.
3	Weak connection to prior content; semantic jump or topic drift occurs.
2	Post-interruption content is almost disconnected from prior context; logical confusion is obvious.
1	Completely ignores the interruption or generates content severely detached from context.

Table 6: Detailed scoring rubric for Visual Evidence Score (VES).

Score	Criterion
10	All mentioned objects, actions, and attributes (e.g., color, position, state) are genuinely present; no hallucination; the description accurately reflects the visual state at the interruption moment; no temporal misalignment.
9	All visual elements genuinely exist; only extremely minor detail imprecision is present; timing is essentially synchronized.
8	Major visual elements are correct; one minor detail may be vague or slightly inaccurate; no obvious hallucination.
7	Core objects and actions are correct, with only one minor visual error or slight temporal lag.
6	Most visual content is correct, but there is one obvious but non-core visual error, or a noticeable temporal synchronization deviation.
5	Core visual elements are correct, but multiple minor visual errors are present, or there is partial over-generalization.
4	One core visual error is present, or there is severe temporal misalignment, such as referring to frames several seconds earlier.
3	Multiple core visual errors are present.
2	Major visual information is severely inconsistent with the actual video.
1	The content is unrelated to the video, describes a completely fabricated scene, or exhibits severe visual hallucination.

Table 7: Detailed scoring rubric for Correction Following Score (CFS).

Score	Criterion
10	Explicitly states that the old conclusion was incorrect; fully retracts the conflicting semantics; completely reconstructs the response under the new constraints; no residual old semantics remain.
9	Fully follows the new constraints; does not explicitly apologize or deny the old conclusion, but contains no actual residual semantic error.
8	Major semantic conflicts are corrected; only extremely minor and irrelevant old phrasing remains.
7	The core conflict is corrected, but some secondary old semantics remain uncleared.
6	The model attempts correction, but still retains noticeable fragments of the old conclusion.
5	The model performs partial rewriting, but the core constraints are not fully satisfied.
4	The model only superficially modifies wording, while substantively continuing the old semantics.
3	The response clearly mixes old and new conclusions, and logical conflict remains.
2	The model largely fails to retract the old semantics and remains clearly inconsistent with the new constraints.
1	The model completely ignores the correction signal or generates content unrelated to the task.

Table 8: Detailed scoring rubric for Contextual Consistency Score (CCS).

Score	Criterion
10	All semantic units that should be preserved are fully retained; only minimal necessary modifications are made; no irrelevant expansion or structural disruption occurs.
9	Almost all preservable context is retained; only a very small amount of non-core information is lost.
8	All core semantics are preserved, but some secondary details are missing.
7	The main semantics are preserved, but some core information is slightly weakened.
6	Core semantics are only partially preserved, and some unrelated content is clearly deleted or rewritten.
5	Only part of the core semantics is preserved, and the structure is noticeably reorganized.
4	A large amount of originally valid semantics is deleted, with clear overturn-and-rewrite behavior.
3	Only very little original semantics is preserved; the response essentially regenerates a new version of the content.
2	The original context is almost erased, and the semantic structure is largely rebuilt.
1	The model completely fails to preserve the original valid context and becomes detached from the original content.

Table 9: Score cap conditions for Visual Evidence Score (VES).

Condition	Maximum Score
One core visual error	≤ 4
Two or more core visual errors	≤ 3
Obvious fabricated scene	≤ 2
Content completely detached from the video	1

Table 10: Score cap rules for Correction Following Score (CFS).

Condition	Maximum Score
Clearly continues the core old conclusion	≤ 4
Old and new conclusions coexist	≤ 3
Completely ignores the correction signal	1

Table 11: Score cap rules for Contextual Consistency Score (CCS).

Condition	Maximum Score
Unjustified deletion of core semantic units	≤ 4
Large-scale rewriting of original context	≤ 3
Complete topic shift	1

B Data Format

Following the task formulation of Online Video Question Answering under Interruption, each benchmark instance is organized around a temporally grounded interrupted interaction. OVIBench provides two complementary data formats: an open-ended format for free-form response evaluation and a multiple-choice format for controlled, low-noise assessment.

B.1 Open-Ended Format

The open-ended format is designed to approximate realistic interrupted interactions, where the model must generate a free-form response after receiving an interruption during answer generation. Each sample contains the original question, the timestamp when the question is issued, the interruption signal, the timestamp when the interruption occurs, and the source video.

For implementation, each sample is stored together with a unique identifier and the interruption type. The question and interruption signal are grouped under an interaction setup field, which explicitly specifies their timestamps. An example is shown below.

```
{
  "case_id": "ActivityNet_120.mp4_s0_Cancellation",
  "video_file": "DATA/my_large_dataset/ActivityNet_120.mp4",
  "interruption_type": "Cancellation",
  "interaction_setup": {
    "question": "Describe the visual elements and possible
      function ...",
    "question_timestamp": 0.97,
    "interruption_signal": "Stop talking",
    "interruption_timestamp": 6.13
  }
}
```

In this example, the model starts answering the original question at timestamp 0.97 and then receives a `Cancellation` interruption at timestamp 6.13. During evaluation, the model is expected to respond according to the interruption signal under this temporal setup.

B.2 Multiple-Choice Format

To reduce evaluation noise and support more reproducible comparisons, OVIBench also provides a multiple-choice version of each interrupted interaction. Instead of requiring the model to freely generate a response, this format presents several candidate answers and asks the model to choose the one that best matches the interruption context. In addition, we avoid constructing overly simple distractors in the multiple-choice setting. Instead of using random irrelevant answers, the incorrect option is designed to be close to plausible responses in surface form but wrong in key semantics.

As shown in the example below, each sample records the interaction state at the moment of interruption, including the dialogue history, the current question, the partial answer generated before interruption, the interruption signal and timestamp, and the corresponding visual stream range. In addition, each sample contains four answer options and a ground-truth label. To avoid positional bias, the option order is randomly shuffled in each sample. An example is given below.

```
{
  "case_id": "ActivityNet_25.mp4_q0_Cancellation",
  "input_state": {
    "history_qa": [],
    "current_question": "Are there any tents or structures
      visible ...",
    "question_timestamp": 5.81,
    "partial_answer_generated": "Yes, in the background ...",

    "interruption_signal": "That's enough",
    "interruption_timestamp": 6.77,
    "visual_stream_range": [5.81, 6.77]
  },
  "options": {
    "A": "...",
    "B": "Understood, I have stopped ...",
    "C": "...",
    "D": "[STOP]"
  },
  "ground_truth": {
    "type": "Cancellation",
    "label": "D",
    "content": "[STOP]",
    "behavior": "stop"
  }
}
```

In this example, the interruption signal "That's enough" indicates a `Cancellation` case. Since the model is expected to stop responding after this interruption, the correct choice is `[STOP]`. In this way, the multiple-choice format complements the open-ended setting by providing a more stable benchmark for interruption understanding and response selection.

C Visualization of Evaluation Metrics

Figure 5 provides intuitive examples of the six evaluation metrics used in OVIBench. Each case contains a short video stream, a user interruption, and representative model responses. Green boxes indicate high-quality responses, while red boxes show typical failure cases.

The figure illustrates how different metrics focus on different aspects of interruption handling, including whether the model follows the interruption intent, maintains textual fluency, grounds its response in visual evidence, satisfies the user's re-

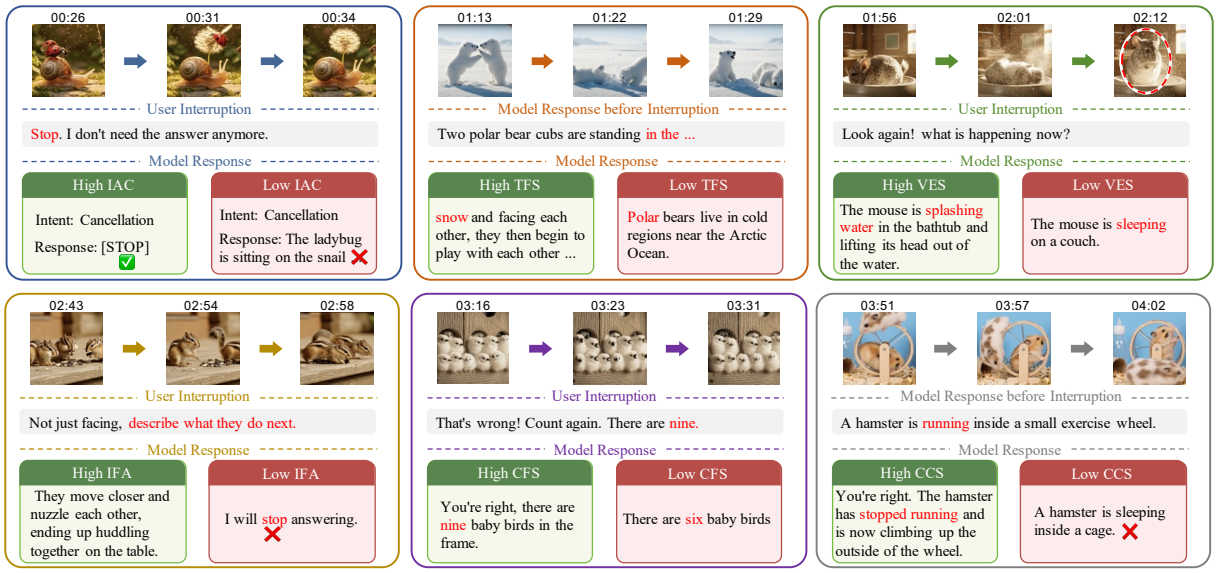


Figure 5: Illustrative examples of the six evaluation metrics in OVIBench. Each case shows a short video stream with a user interruption and the model's responses. Green boxes denote responses that achieve high scores, while red boxes indicate failure cases.

quest, follows correction signals, and preserves valid context. This visualization complements the metric definitions and helps clarify the scoring criteria used in human and model-based evaluation.

D Human Evaluation Subset

To further verify whether the observed model performance and the gains from interruption-aware fine-tuning are stable beyond automatically generated data, we construct a human-written evaluation subset, denoted as OVI-Human. This subset is not used for OVI-Train training, prompt tuning, or model selection. Instead, it is only used as an additional evaluation set to examine whether the main conclusions still hold under manually written questions and interruption expressions.

OVI-Human contains 180 samples, with 60 samples for each interruption type: Cancellation, False Trigger, and Correction. The questions cover diverse user intents, including video description, object counting, action change, visual detail localization, and simple causal reasoning. The interruption signals are also written in more natural forms, such as implicit stop requests, colloquial corrections, and irrelevant short insertions. The answer options are manually written or substantially rewritten to reduce the style bias of automatic generation. We evaluate models on OVI-Human using the same multiple-choice metrics as the main experiment, namely Multiple-Choice Accuracy (MA) and Interruption-Type Classification Accuracy (CA).

As shown in Table 12, the results on OVI-Human show a similar overall trend to the main multiple-choice evaluation. Our fine-tuned model still achieves the best overall performance, improving All MA by 17.78 points and All CA by 15.00 points over Qwen2.5-VL-7B. The improvements are especially clear on False Trigger and Cancellation, indicating that OVI-Train helps the model better distinguish ineffective interruptions and stopping requests. These results suggest that the gains from interruption-aware fine-tuning are not limited to automatically generated samples. Since OVI-Human uses manually written questions and more natural interruption expressions, the consistent improvement provides additional evidence that the model learns useful interruption-handling ability rather than only fitting the style of the generated benchmark.

E Human Evaluation Instructions

To ensure the quality of the OVIBench dataset, we conducted a human evaluation after the automatic data construction process. OVIBench focuses on online video question answering under interruption, where a model is expected to react properly to user interruptions during answer generation. The dataset contains three interruption types: *Cancellation*, *False Trigger*, and *Correction*, corresponding to stopping the response, ignoring an ineffective interruption, and revising the response according to new user-provided information.

Table 12: Multiple-choice evaluation results on the human-written OVI-Human subset. MA denotes answer selection accuracy, and CA denotes interruption-type classification accuracy. Δ indicates the performance improvement of our model over the baseline. The best results in each column are highlighted in bold.

Model	**All**		**False Trigger**		**Cancellation**		**Correction**	
	MA	**CA**	**MA**	**CA**	**MA**	**CA**	**MA**	**CA**
Open-Source Models								
Qwen2.5-VL-7B (Yao et al., 2024)	62.78	79.44	48.33	91.67	42.78	50.00	85.56	94.67
Qwen2.5-VL-72B (Yao et al., 2024)	69.44	90.56	86.67	96.67	38.33	**91.67**	83.33	83.33
VideoChat-R1-7B (Li et al., 2025)	68.33	81.67	52.78	91.67	49.44	58.33	88.89	95.00
Qwen3-VL-30B-A3B (Bai et al., 2025)	70.00	86.11	28.33	73.33	68.89	81.67	96.67	98.33
Closed-Source Models								
Doubao-Seed-1.6 (Huang et al., 2025a)	69.44	92.22	38.89	95.00	51.67	83.33	97.78	96.33
Gemini-2.5-Flash (Comanici et al., 2025)	55.00	66.11	60.00	68.33	21.67	53.33	83.33	76.67
Ours (7B)	**80.56**	**94.44**	**90.00**	**98.33**	**76.67**	88.33	**97.78**	**96.67**
Δ (vs Qwen2.5-VL-7B)	+17.78	+15.00	+41.67	+6.66	+33.89	+38.33	+12.22	+2.00

We invited 10 human participants to evaluate the generated samples. Each participant was asked to inspect the video content, the original question, the interruption signal, the interruption type label, the partial response before interruption, and the post-interruption response. For multiple-choice samples, participants also checked whether the candidate options and the ground-truth answer were reasonable. The final human score of each sample was computed by averaging the scores from the 10 participants, which reduces the influence of individual subjective bias.

The human evaluation mainly considered the following aspects:

1. **Correctness of interruption type.** Participants checked whether the interruption signal was consistent with the annotated interruption type. For example, a stop request should be labeled as *Cancellation*, a corrective signal should be labeled as *Correction*, and an irrelevant or accidental signal should be labeled as *False Trigger*.

2. **Appropriateness of response behavior.** Participants examined whether the post-interruption response matched the expected behavior. A *Cancellation* case should stop the response, a *False Trigger* case should continue the original answer, and a *Correction* case should revise the previous response according to the user correction.

3. **Consistency with visual evidence.** Participants checked whether the objects, actions, scenes, and temporal information mentioned in the response were supported by the video content, avoiding obvious hallucinations or contradictions.

4. **Contextual coherence.** Participants evaluated whether the response before and after the interruption was coherent. For correction cases, the model should revise the incorrect part while preserving valid context that is not affected by the correction.

5. **Quality of multiple-choice options.** For multiple-choice samples, participants checked whether the correct option was unique and consistent with the interruption context, and whether the distractors were distinguishable without being equally correct.

For scoring, binary criteria, such as whether the response behavior is correct, were marked as 0 or 1. Other quality dimensions, including textual fluency, visual consistency, correction quality, and contextual coherence, were scored on a 1–10 scale. For each sample, we collected the scores from all 10 participants and used their average as the final human evaluation score. In this way, the final score reflects the overall judgment of multiple human evaluators rather than the opinion of a single participant. Samples with low scores or large disagreement among participants were manually rechecked. According to the identified problem, these samples were removed, revised, or relabeled. Through

this human evaluation process, we filtered out samples with incorrect interruption labels, inconsistent visual evidence, unreasonable response logic, or ambiguous answer options, thereby improving the overall quality and reliability of the dataset.

F Offline Simulation Pseudocode

To enable large-scale and reproducible evaluation, we simulate the online interruption process offline under a unified temporal protocol. As described in the main paper, the simulation first determines the observable video segment at the question timestamp, then generates a preliminary answer, truncates it according to the interruption time and a fixed generation rate, and finally feeds the interruption signal together with the updated visual context back to the model for post-interruption response generation. In this way, the protocol approximates interruption during answer generation while remaining suitable for batch evaluation and controlled comparison across models.

Offline Simulation of Online Video Interruption

Require: Target model $\mathcal{M}$, benchmark set $\mathcal{B}$, character generation rate R

Ensure: Simulation result set $\mathcal{R}$

Initialize $\mathcal{R} \leftarrow \emptyset$

for each interaction sample $b \in \mathcal{B}$ **do**

Parse b into video V, question Q, interruption signal I, question timestamp T_0, and interruption timestamp T_1

Step 1: Simulate the initial answering stage

Extract the observable video segment before the question:

$V_Q \leftarrow V[0 : T_0]$

Generate the complete preliminary answer:

$A \leftarrow \mathcal{M}(V_Q, Q)$

Step 2: Simulate interruption timing during generation

Compute the elapsed speaking time:

$\Delta T \leftarrow \max(0.5, T_1 - T_0)$

Estimate the generated prefix length:

$k \leftarrow \lfloor \Delta T \cdot R \rfloor$

Truncate the preliminary answer to obtain the interrupted partial response:

$A_I \leftarrow \textsc{Truncate}(A, k)$

Step 3: Simulate post-interruption reasoning

Extract the updated observable video segment up to the interruption time:

$V_I \leftarrow V[0 : T_1]$

Construct the interruption prompt P using interruption signal I and task instructions

Query the model with the updated context, original question, and partial response:

$Y \leftarrow \mathcal{M}(V_I, Q, A_I, P)$

Parse Y into predicted interruption type $\hat{c}$ and post-interruption response A'

Step 4: Record simulation outputs

Construct result entry

$r \leftarrow \{Q, I, T_0, T_1, A, A_I, \hat{c}, A'\}$

Add r to $\mathcal{R}$

end for

return $\mathcal{R}$