ASKSTREAM: WHEN AND WHAT TO ASK PROACTIVELY UNDER UNCERTAINTY IN STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in multimodal large language models have increased interest in interaction paradigms for video understanding. The evolution from offline to streaming and proactive interaction has gradually shifted models from passive responders toward more proactive agents that can decide when to answer. However, existing proactive paradigms typically assume that the information required to answer a query is complete, leaving models to only *wait* or *answer*. In the real world, however, information is often incomplete due to uncertainty from either the **video** or the **user**. In proactive paradigms, such uncertainty can cause the model to wait forever or generate highly hallucinatory responses. We therefore argue that proactive interaction should be redefined with a third action: *ask*, allowing models to actively ask for missing information. To study when and what models should ask before answering, we introduce **AskStream**, the first multi-turn benchmark spanning video and user uncertainty for streaming video understanding, with four task categories and eight subtasks. It contains 438 videos with 2,714 manually annotated frame-level time windows across three types: wait, ask, and answer. Across both multiple-choice and open-ended settings, even the strongest model attains only 52.18% on our proposed **Ask-F1** metric, highlighting substantial limitations in proactive information seeking under uncertainty.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made significant progress in video understanding and reasoning in recent years (Kimi Team, 2026; Qwen Team, 2026; Google DeepMind, 2026; OpenAI, 2026). Beyond advancements in model architecture and training paradigms, research on **interaction paradigms** is equally important for advancing models toward more adaptive and generalizable intelligence in real-world environments. As shown at the top of Figure 1, traditional video interaction primarily follows an **offline paradigm**, where the model observes the complete video before answering user queries. In the real world, videos often arrive continuously as streams. This has driven the emergence of a **streaming paradigm**, where models must continuously understand the incoming stream, maintain memory, and interact with users in real time. Under both paradigms, the core task remains to determine *how to answer*. However, because the streaming paradigm primarily targets real-time responses, it is less suited to scenarios requiring delayed responses, such as anomaly alerts, procedural guidance, and continuous task monitoring. This motivates the **proactive paradigm**, where models continuously monitor the video and decide not only *how to answer* but also *when to answer*. Existing benchmarks (Lin et al., 2026b; Wang et al., 2025c; Azad et al., 2026; Li et al., 2026) progressively extend this paradigm by requiring models to wait for sufficient evidence, supporting multi-turn responses, modeling answer readiness, and adapting response timing under dynamic tasks, with a common focus on increasingly fine-grained response timing.

However, **the paradigm implicitly assumes that the information for answering is fully contained in the stream and will eventually become available, leaving the model only to *wait* or *answer*.** This assumption overlooks a key challenge: *information in the real world is often incomplete*. Such incompleteness may cause the model to wait forever or produce hallucinatory responses. As shown at the bottom of Figure 1, the model should proactively ask for missing video information or clarify which person is being referred to before answering. To further quantify the effect of acquiring

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

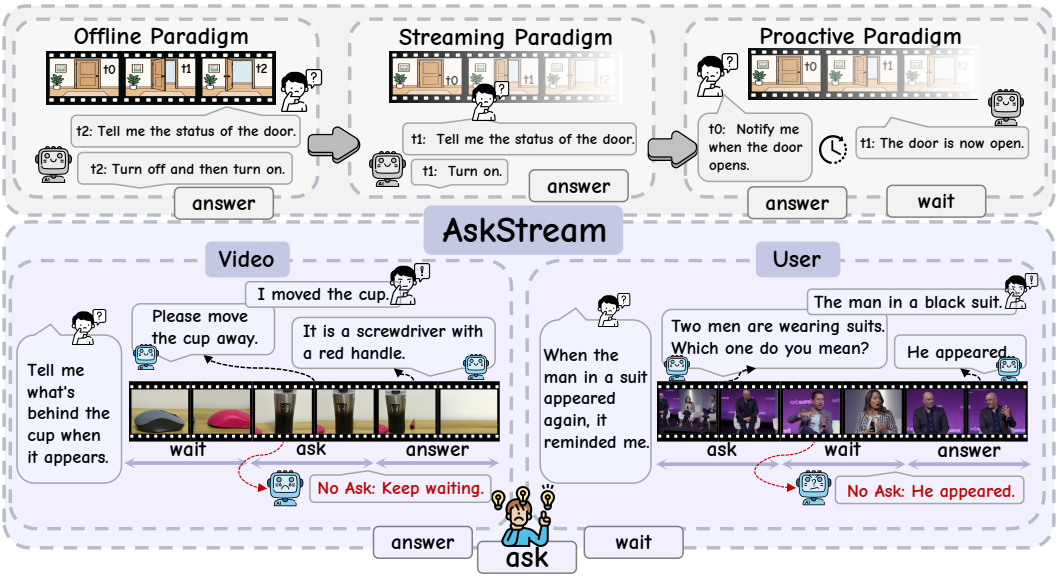


Figure 1: Evolution of interaction paradigms. AskStream extends the conventional action space of *wait* and *answer* by introducing *ask*, enabling models to proactively seek missing information from the video or user before answering.

missing information, we analyze the model's uncertainty. [1] As shown in Figure 2, correct evidence reduces uncertainty and shifts predictions toward the correct answer, raising accuracy from 42.3% to 62.3%, while mismatched evidence provides little benefit. These observations show that acquiring key information improves model's performance. We therefore argue that the proactive paradigm should be redefined beyond *wait&answer* to also proactively *ask* missing information when needed.

Based on the above analysis, we introduce **AskStream**, which is the first multi-turn benchmark to incorporate an *ask* action into the proactive paradigm in streaming video understanding. This paradigm is suitable for applications, such as everyday visual assistance and remote operational guidance. We categorize information uncertainty into **video uncertainty** and **user uncertainty**. Video uncertainty arises from missing visual evidence or missing personalized visual concepts, whereas user uncertainty arises from ambiguous user intent or insufficient knowledge of user preferences. We further divide these four uncertainty categories into eight subtasks, as illustrated in Figure 3. Compared with prior proactive benchmarks that annotate only timestamps, we provide fine-grained manual annotations of time windows for *wait*, *ask*, and *answer* and introduce **Ask-F1** to jointly evaluate the timing and content correctness of both asking and answering. To enable more realistic evaluation, we develop **StreamHarness**, a hierarchical memory framework that retains recent frames while compressing earlier observations and evaluate models under both multiple-choice and open-ended settings. Experimental results reveal a substantial gap between offline and streaming proactive models: the strongest offline model achieves 52.18% Ask-F1, while the best streaming proactive model reaches only 7.22%. Early ASK and wrong ASK are the two common error types, while providing oracle information about both *when* and *what* to ask improves Ask-F1 by 43.83%↑, highlighting proactive asking as a major bottleneck in current models. **Our contributions** are as follows:

- **A new interaction paradigm.** Quantitative and qualitative analyses show that proactively acquiring missing information reduces model uncertainty, motivating a new proactive interaction paradigm with three actions: *wait*, *ask*, and *answer*.
- **AskStream benchmark.** We introduce AskStream, a multi-turn benchmark that jointly evaluates when and what to ask and when to answer across eight subtasks spanning video and user uncertainty, with fine-grained manually annotated action windows, **Ask-F1** metric and the **StreamHarness** hybrid framework.
- **Experiments.** Experiments across diverse models under both settings reveal substantial limitations in proactive information seeking, with the best offline and streaming proactive models achieving only 52.18% and 7.22% Ask-F1, respectively.

[1] See Appendix A.3 for experimental details.

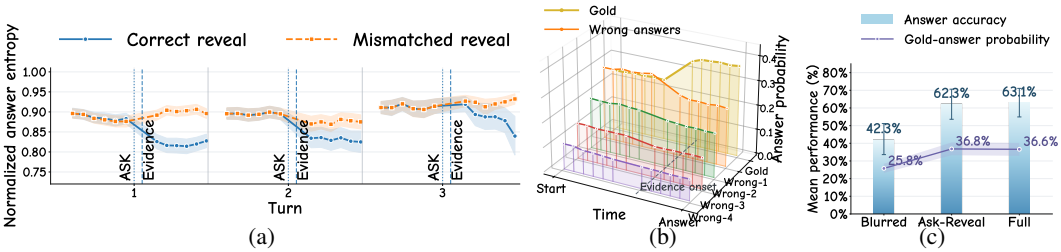


Figure 2: **Analysis of model uncertainty.** (a) Across interaction turns, correct evidence consistently lowers normalized answer entropy, unlike mismatched evidence. (b) Relevant evidence shifts probability toward the correct answer. (c) Accuracy improves from 42.3% (blurred) to 62.3% after asking for key evidence, nearing 63.1% with fully clear input.

2 Related Work

2.1 Offline Paradigm

Traditional video understanding follows an offline paradigm, in which models answer questions after observing the entire video. Benchmarks such as Video-MME (Fu et al., 2025a), LVBench (Wang et al., 2025a), LongVideoBench (Wu et al., 2024) and other benchmarks (Zhou et al., 2025; Li et al., 2024) focus on visual perception, temporal reasoning, and long-horizon video understanding.

2.2 Streaming Paradigm

Streaming video understanding requires models to reason only over video content observed so far. VStream-QA (Zhang et al., 2024) and StreamingBench (Lin et al., 2026b) established the standards for evaluating early streaming videos. OVO-Bench (Niu et al., 2025), RTV-Bench (Xun et al., 2026), and StreamArena (Zhang et al., 2026c), along with other benchmarks (Lin et al., 2026a; Huang et al., 2025; Yang et al., 2025), further extend evaluation to dynamic temporal reasoning and long-horizon streaming interaction. Meanwhile, existing methods mainly improve model performance under the streaming paradigm from three aspects: computational efficiency (Chen et al., 2024; Xu et al., 2026; Wang et al., 2026c), GPU memory usage (Chen et al., 2026; Zhang et al., 2026a; 2024), and long-horizon memory (Wang et al., 2026a; Guan et al., 2026; Zhang et al., 2026b).

2.3 Proactive Paradigm

Proactive video understanding extends evaluation from *how to answer* to *when to answer*. Early benchmarks (Lin et al., 2026b; Niu et al., 2025) require models to wait until sufficient evidence is observed, while ProactiveVideoQA (Wang et al., 2025c) and ProReady-QA (Azad et al., 2026) further introduce metrics that jointly evaluate answer quality and timing. IPIBench (Li et al., 2026) supports dynamic timestamp modification, OmniMMI (Wang et al., 2025d) incorporates audiovisual interaction, and other benchmarks (Lu et al., 2026b; Fu et al., 2025b; Yan et al., 2026) cover procedural guidance, mobile interaction, and game commentary. Methodologically, JoyAI-VL-Interaction (Yao et al., 2026) and MOSS-VL (Wang et al., 2026b) improve proactive response capabilities through architectural and training innovations. For proactive information asking, ClearVQA (Jian et al., 2025) addresses visual ambiguity through visual feedback, while ProactiveBench (De Min et al., 2026) supports actions such as removing occlusions and changing viewpoints. However, these works mainly focus on static images and do not distinguish different sources of information uncertainty, leaving proactive information asking in streaming video underexplored. **AskStream** addresses this gap by extending proactive video understanding toward active information asking and jointly modeling video and user uncertainty.

3 AskStream

3.1 Task Definition

Task Modeling. We model **AskStream** as a partially observable Markov decision process (POMDP), where the latent state captures information that is missing from the video stream or the user stream. For each interaction event, the model makes decisions based only on the video

Table 1: **Comparison of video understanding benchmarks.** MC & OE denote multiple-choice & open-ended evaluation. Opt., Task, & Dur. denote average options, task number, and average video duration. TS & Win. denote timestamps and time windows. FL, MT, & Pro. denote frame-level annotation, multi-turn interaction, and proactive tasks. † denotes values measured by us.

Benchmark	Eval.	Opt.	Task	Dur.	Timing	FL	Ask	MT	Pro.
(A) Offline video benchmarks									
Video-MME (Fu et al., 2025a)	MC	4	12	17.0	–	✗	✗	✗	✗
LongVideoBench (Wu et al., 2024)	MC	4.73†	17	7.9	–	✗	✗	✗	✗
(B) Streaming video benchmarks									
StreamingBench (Lin et al., 2026b)	MC	4	18	4.1	TS	✗	✗	✓	✓
RIVER-Bench (Shi et al., 2026)	ALL	4	3	24.1†	TS	✗	✗	✗	✓
RTV-Bench (Xun et al., 2026)	MC	3.53†	8	18.2	TS	✗	✗	✗	✗
StreamArena (Zhang et al., 2026c)	OE	–	4	88.8	TS	✗	✗	✓	✓
(C) Proactive streaming benchmarks									
OmniMMI (Wang et al., 2025d)	ALL	–	6	5.4	TS	✗	✗	✓	✓
ProactiveVideoQA (Wang et al., 2025c)	OE	–	4	2.1	Win.	✗	✗	✓	✓
ProReady-QA (Azad et al., 2026)	OE	–	5	40.0	Win.	✗	✗	✓	✓
IPIBench (Li et al., 2026)	OE	–	9	1.0†	TS	✗	✗	✓	✓
AskStream (Ours)	**ALL**	**9.44**	**8**	**11.0**	**Win.**	✓	✓	✓	✓

observed so far, the user query, and previous interactions. At each decision point, the model chooses one of three actions: `WAIT`, `ASK`, or `ANSWER`.

Time Window Annotation. We divide a video stream into multiple events. Event i begins when the current query arrives at q_i and ends when the next query arrives at q_{i+1}, defining the event interval as $[q_i, q_{i+1})$. Because the information uncertainty may be recognized at the beginning or as the video unfolds, we define two types of temporal windows:

- **Type I: wait before asking.** In this type of event, the initial observations are insufficient for the model to identify the uncertainty. The model must therefore wait until relevant evidence emerges before asking. Let a_i denote the time at which the model autonomously issues `ASK`, and let s_i and e_i denote the annotated start and end of the `ANSWER` window, respectively. For this type, we annotate only the `WAIT` and `ANSWER` windows:

$$W_i^{\text{wait}} = [q_i, s_i), \qquad W_i^{\text{answer}} = [s_i, e_i). \tag{1}$$

  A valid `ASK` occurs at $a_i \in W_i^{\text{answer}}$. It ends the model's actual waiting interval $[q_i, a_i)$ and initiates the answering phase. The model then should answer after receiving the requested information.

- **Type II: ask before waiting.** In the other type of event, the initial observations are sufficient for the model to identify the uncertainty. The model should issue `ASK` within the annotated `ASK` window and then wait until relevant evidence for answering emerges. For this type, we annotate only the `ASK` and `ANSWER` windows:

$$W_i^{\text{ask}} \subseteq [q_i, s_i), \qquad W_i^{\text{answer}} = [s_i, e_i). \tag{2}$$

  For a valid `ASK` at $a_i \in W_i^{\text{ask}}$, the model's actual waiting interval is $[a_i, s_i)$, whose start is determined dynamically by the `ASK` action rather than manually annotated.

Action Evaluation. Let $\omega_{i,k}$ denote the time of the k-th WAIT action, and let a_i and ρ_i denote the times when the model issues `ASK` and `ANSWER` for event i, respectively. $\omega_{i,k}$ indicates that the model chooses to observe further and ρ_i provides the final response. In our main setting, each turn allows at most one `ASK` and `ANSWER`. The model is allowed to make one action selection per second. A correct `ASK` triggers the user to provide the missing information. A turn is successful only if the model issues a correct ask within W_i^{answer} for Type I and W_i^{ask} for Type II, and subsequently submits the correct answer within W_i^{answer}. Otherwise, the turn is marked as failed and the model proceeds to the next query.

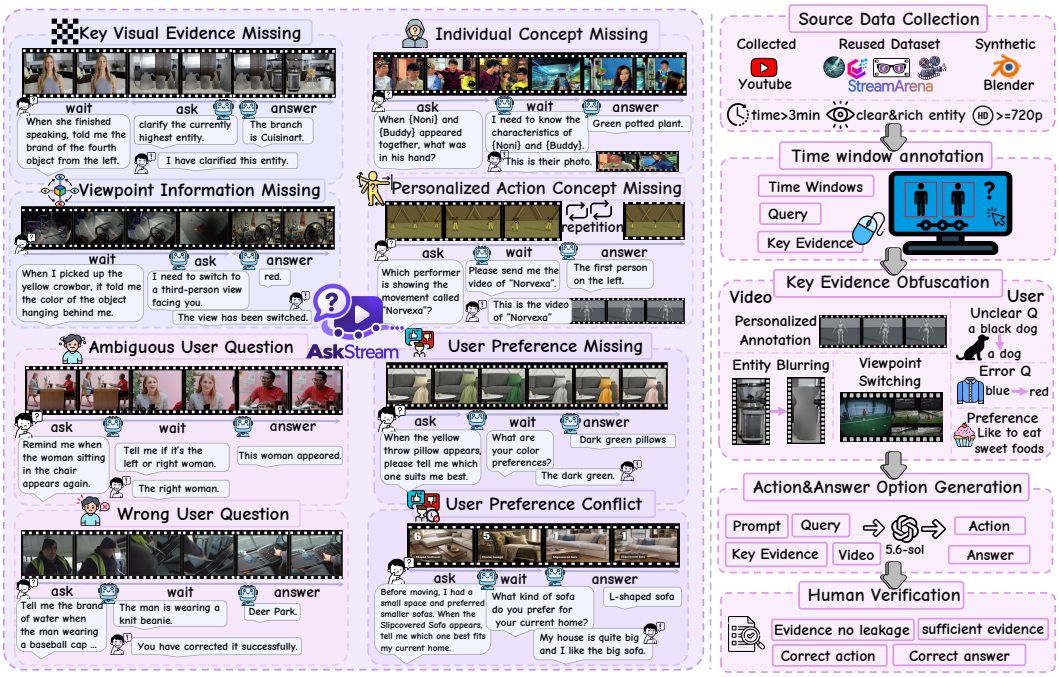


Figure 3: **AskStream task taxonomy and benchmark construction.** Left: eight subtasks spanning video and user uncertainty, with examples of *wait*, *ask*, and *answer* interactions. Right: the five-stage construction pipeline.

3.2 ASK-F1

Inspired by prior timing-aware evaluation (Azad et al., 2026; Wang et al., 2025c), we introduce **Ask-F1** to jointly evaluate the temporal and content correctness of both ASK and ANSWER. For turn i, a_i and ρ_i denote the ASK and ANSWER times of the model, $W_i^a = [s_i^a, e_i^a)$ and $W_i^\rho = [s_i^\rho, e_i^\rho)$ their valid windows, and $\mathbb{I}_i^a$ and $\mathbb{I}_i^\rho$ indicate whether the ASK and ANSWER contents are correct. We further define $\mathbb{I}_i^{\text{ask}} = \mathbb{I}_i^a \mathbb{I}[a_i \in W_i^a]$, which equals 1 only when the ASK content is correct and the ASK is issued within the valid ASK window. Accordingly, $N_{\text{ask}} = \sum_i \mathbb{I}_i^{\text{ask}}$ denotes the total number of such correct ASK turns across all turns. We define

$$T(t;[s,e)) = \mathbb{I}[s \le t < e]\left(1 - \frac{t-s}{e-s}\right), \quad S_{\text{ask}} = \frac{1}{N}\sum_i \mathbb{I}_i^a T(a_i; W_i^a), \quad S_{\text{ans}} = \begin{cases} \frac{1}{N_{\text{ask}}}\sum_i \mathbb{I}_i^{\text{ask}}\mathbb{I}_i^\rho T(\rho_i; W_i^\rho), & N_{\text{ask}} > 0, \\ 0, & N_{\text{ask}} = 0. \end{cases} \tag{3}$$

$$\text{Ask-F1} = \begin{cases} \dfrac{2S_{\text{ask}}S_{\text{ans}}}{S_{\text{ask}} + S_{\text{ans}}}, & S_{\text{ask}} + S_{\text{ans}} > 0, \\ 0, & S_{\text{ask}} = S_{\text{ans}} = 0. \end{cases} \tag{4}$$

S_{ask} and S_{ans} represent the timing-aware scores for asking and answering, respectively, while N denotes the total number of turns. Further analysis of the metric design is provided in Appendix A.2.

3.3 Task Taxonomy

The uncertainties in the real world may arise from videos or users. Accordingly, we divide our benchmark into **video uncertainty** and **user uncertainty**, which include four tasks and eight subtasks, as illustrated in Figure 3.

3.3.1 Video Uncertainty

Video uncertainty arises at two levels: either the visual evidence required by the question is missing, or the visual evidence is complete but the relevant personalized concepts are absent. These correspond to explicit and implicit video information uncertainty, respectively.

Explicit Video Information Uncertainty: (1) *Key Visual Evidence Missing (KVEM)*: When the key entity in the initial video is blurred, the model must recognize that the evidence is insufficient and

proactively request a video showing the entity clearly. It then answers the question using the clear video provided by the user. (2) *Viewpoint Information Missing (VIM)*: When the initial viewpoint lacks evidence needed to answer the question, the model must identify this gap and proactively request an alternative viewpoint that reveals the missing evidence. It then answers after the user switches to the requested view.

Implicit Video Information Uncertainty: (1) *Individual Concept Missing (ICM)*: Given a clear video, the model lacks the personalized concept information about a visible entity needed to answer the question. It must recognize this memory gap and proactively request the relevant information from the user before answering. (2) *Personalized Action Concept Missing (PACM)*: Although a clear video fully captures a user-specific action, the model lacks the personalized conceptual knowledge needed to answer the question. It must recognize this gap and proactively request the action concept's definition from the user before answering.

3.3.2 User Uncertainty

User uncertainty concerns accurately understanding the user's request or obtaining the current preferences, corresponding to user intent uncertainty and user preference uncertainty, respectively.

User Intent Uncertainty: (1) *Ambiguous User Question (AUQ)*: When a question leaves the user's intent unclear, the model must recognize the ambiguity while observing the video, proactively request clarification from the user, and then answer the clarified question. (2) *Wrong User Question (WUQ)*: When a question contains a false premise about the video, the model must detect its inconsistency with the visual evidence while observing the video, correct the premise, and answer the question.

User Preference Uncertainty: (1) *User Preference Missing (UPM)*: When answering depends on unknown user preferences, such as style or interests, the model must proactively ask for the relevant preferences and combine the user's feedback with visual evidence to provide a personalized answer. (2) *User Preference Conflict (UPC)*: Given a video, historical user preferences, and a question requiring updated preferences, the model must recognize that the existing preferences are outdated, proactively ask for the user's current preferences, and answer using the updated information.

3.3.3 Interaction Timing

Explicit video requires observing relevant evidence before uncertainty can be identified, following the recognition timing in equation 1. Implicit video and user preference uncertainty can be identified from the outset, so the model asks first and then waits for relevant evidence before answering in equation 2. User intent uncertainty spans both timing patterns, depending on whether the ambiguity can be recognized from the initial query or only after observing relevant visual evidence.

3.4 Benchmark Construction

We construct **AskStream** through the five-stage pipeline in Figure 3. (1) We collect videos from YouTube, existing datasets and synthesize additional videos using Blender. (2) Using a self-built annotation webpage in Appendix A.7, we then manually annotate user queries, time windows, key entities, and the evidence required to answer each query. Time windows are annotated with frame-level precision to provide timing supervision for *wait*, *ask*, and *answer*. (3) We use Grounded SAM 2 (Ren et al., 2024) to segment and track entities for selective Gaussian blurring and PySceneDetect to assist video splitting. We also change viewpoints, modify queries, and withhold or alter personalized concepts and preferences. (4) We provide the resulting data to GPT-5.6 Sol (OpenAI, 2026) to generate action and answer options. (5) We manually verify each instance, checking that humans cannot uniquely determine the answer without the missing evidence or clarification but can do so once it is provided, and that each action and answer candidate set contains exactly one correct option. The annotation webpage and task-specific screening, annotation, and verification details are provided in Appendix A.7.

3.5 Benchmark Statistics

Figure 4 summarizes the main statistics of AskStream, which comprises 438 videos and 1,401 questions across eight subtasks. The videos span nine domains, with durations ranging from 16.72 seconds to 118 minutes, covering diverse streaming scenarios. In addition, the total durations of the three types of temporal windows, *wait*, *ask*, and *answer*, are 33.92, 30.47, and 12.00 hours, respectively.

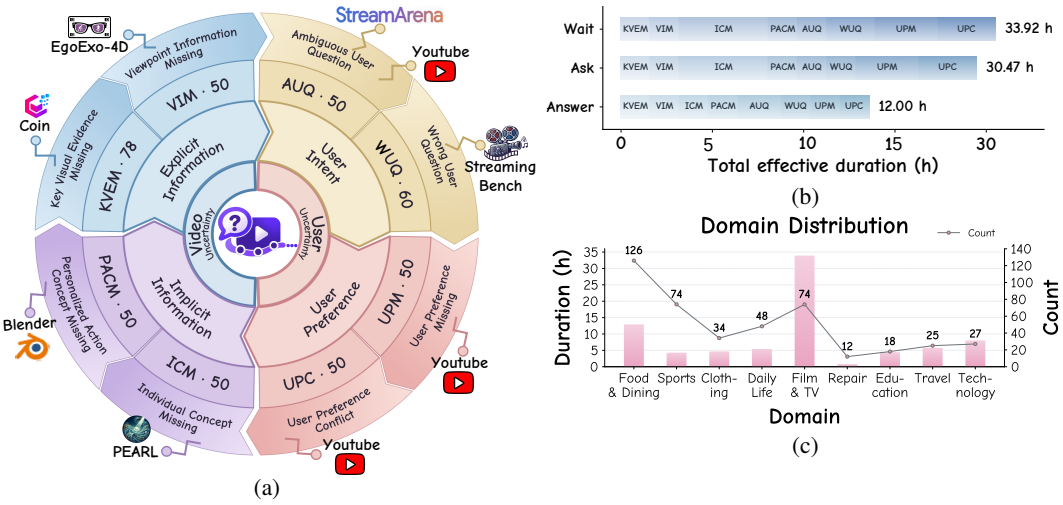


Figure 4: **Statistics of AskStream.** (a) Task taxonomy, data distribution, and sources. (b) Total effective durations of *wait*, *ask*, and *answer* across subtasks. (c) Video durations and sample counts across domains. The benchmark averages 3.2 interaction turns per video and contains 2714 manually annotated timestamps in total.

4 Experiments

4.1 Setup

4.1.1 Baseline

We evaluate three groups of models on **AskStream**: (A) Offline Closed-source Models, including GPT-5.6-Sol (OpenAI, 2026), Gemini-3.8 Flash (Google DeepMind, 2026), and Claude Sonnet 5 (Anthropic, 2026); (B) Offline Open-Source Models, including Qwen3.8-flash (Qiu et al., 2026), MiMo-V2.5 (Team et al., 2026b), Kimi K2.6 (Team et al., 2026a), and DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026); and (C) Streaming Models with Proactive Capabilities, including VideoLLM-online (Chen et al., 2024), TimeChat-Online (Yao et al., 2025), MMDuet2 (Wang et al., 2025b), MiniCPM-o 4.5 (Cui et al., 2026), AURA (Lu et al., 2026a), JoyAI-VL-Interaction (Yao et al., 2026), and MOSS-VL-Realtime (Wang et al., 2026b). We exclude streaming models without proactive capabilities because they are unable to respond to proactive tasks.

4.1.2 Evaluation protocols

Existing streaming benchmarks (Lin et al., 2026b; Niu et al., 2025) often evaluate offline models through sparse uniform frame sampling, which may miss the dense temporal information required in streaming scenarios. Sliding window methods (Shen et al., 2026) preserve recent frames but discard earlier video context. We therefore introduce **StreamHarness**, a hierarchical memory framework. It retains the most recent 10 frames (1 fps) as short memory, summarizes every 10 frames into one mid-term memory, consolidates every four mid-term memories into one long-term memory block, and retains up to eight long-term blocks. Native streaming models instead use their own memory mechanisms. Across multiple turns, only shared video memory is preserved. Questions, options, and action histories are discarded after each turn. The model is evaluated at one action opportunity per second, with a tolerance of ±1 second. To more realistically evaluate models' proactive capabilities, we adopt both **multiple-choice** and **open-ended** settings. In the open-ended setting, we use Gemini-3.8 Flash as the judge model to assess the correctness of both the model's `ASK` and `ANSWER` responses. Please see section 3.1 to check the definition of the correct turn.

4.2 Main Results

Table 2 presents the main results under the multiple-choice setting, while results for the open-ended setting are provided in Appendix A.4. **(1)** Among offline models, Gemini-3.8 Flash achieves relatively strong performance on most subtasks, reaching 37.17%, 43.10%, and 81.36% on KVEM, WUQ, and UPC, with an overall Ask-F1 score of **52.18%**. However, performance differences

Table 2: Main results on AskStream in the multiple-choice setting. Scores reported for individual subtasks are Ask-F1 scores. Bold denotes the best performance on each subtask.

Method	Backbone	Video				User				Metric		
		Explicit		Implicit		Intent		Preference				
		KVEM	VIM	ICM	PACM	AUQ	WUQ	UPM	UPC	S_{ask}	S_{ans}	**Ask-F1**
Offline closed-source models												
GPT-5.6-Sol	-	18.10	2.93	3.05	0	15.69	17.50	18.60	26.55	18.11	23.62	20.50
Claude Sonnet 5	-	21.48	13.01	7.60	9.70	13.28	29.42	7.93	29.53	20.87	24.37	22.48
Gemini-3.8 Flash	-	**37.17**	17.46	**33.26**	**17.73**	**31.13**	**43.10**	**64.03**	**81.36**	**44.63**	**62.80**	**52.18**
Offline open-source models												
MiMo-V2.5	-	28.49	3.73	16.16	0	11.48	11.83	28.66	26.18	10.48	24.38	14.66
Kimi K2.6	-	23.41	**21.56**	9.27	0	16.74	34.40	0	30.69	19.35	14.41	16.52
Qwen-3.8 Flash	-	21.03	15.81	1.29	9.94	6.28	23.20	4.64	0	31.75	12.17	17.60
DeepSeek-V4.1-Flash	-	17.68	2.33	6.27	0	17.97	19.88	6.32	15.27	18.64	20.87	19.69
Streaming proactive models												
MOSS-VL-Realtime	Qwen3-8B	0	0	0	0	0	0	0	0	0.74	0	0
MMDuet2	Qwen2.5-VL-3B	0	0	0	0	0	0	0	0	0	0	0
VideoLLM-online	Llama-3-8B	0	0	0	0	0	0	0	0	0	0	0
MiniCPM-o 4.5	Qwen3-8B	6.76	0	0	0	0	0	0	0	7.17	0.06	0.12
TimeChat-Online	Qwen2.5-VL-7B	9.29	0	0	0	0	4.15	0	0	12.58	2.51	4.18
JoyAI-VL-Interaction	JoyAI-VL-1.0	5.73	0	0	0	0	4.03	0	0	7.54	3.64	4.91
AURA	Qwen3-VL-8B	8.31	4.35	0	0	5.26	2.93	0	0	10.85	5.41	7.22

across different types of uncertainty, and for stronger models, S_{ask} is significantly lower than S_{ans}, indicating that `ASK` is the bottleneck. **(2)** In contrast, streaming proactive models perform significantly worse: they score zero on most subtasks, and most models fail to complete the task. Even AURA, which performs relatively better, achieves a final score of only 7.22%. This indicates that the current proactive model lacks the ability to `ASK`. **(3)** We further analyze different types of model errors. `E-ASK/E-ANS`, `W-ASK/W-ANS`, and `M-ASK/M-ANS` denote early, wrong, and missed `ASK/ANSWER` actions, respectively. Figure 5 shows that **early ASK** is the main error types across offline models. This indicates that models often recognize the need for additional information, but tend to either ask too early, suggesting that *when to ask* remain major bottlenecks. Overall, while current MLLMs demonstrate emerging proactive information seeking capabilities, their performance remains constrained by unreliable `ASK` decisions.

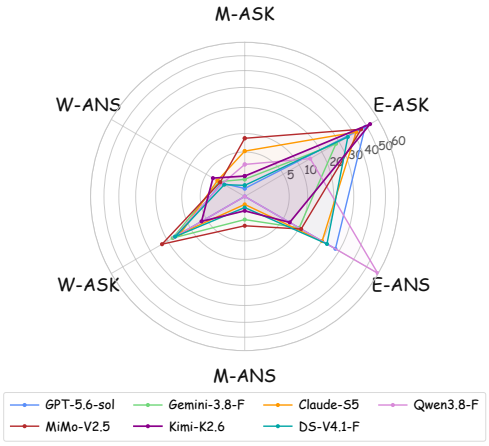


Figure 5: **Error-type distributions** of offline models in the multiple-choice setting. Each vertex shows the percentage of the corresponding error type.

4.3 Analysis Experiment

4.3.1 Where Do Models Fail?

To understand how decisions at different stages affect the model's final performance, we design an ablation study that gradually adds oracle information. We remove uncertainty about when to `ASK`, what to `ASK`, and when to `ANSWER`, as well as both `ASK` factors and all three factors, and measure how much performance can be recovered. Figure 6(a) shows that providing oracle information about when to `ASK` leads to the largest performance improvement, raising Ask-F1 from 37.17% to 58.30% (+21.13 points), further supporting the finding in Section 4.2 that *when to ask* is the main bottleneck. Figure 6(b) compares different amounts and combinations of oracle information, showing that performance generally improves as more decision-stage uncertainty is removed.

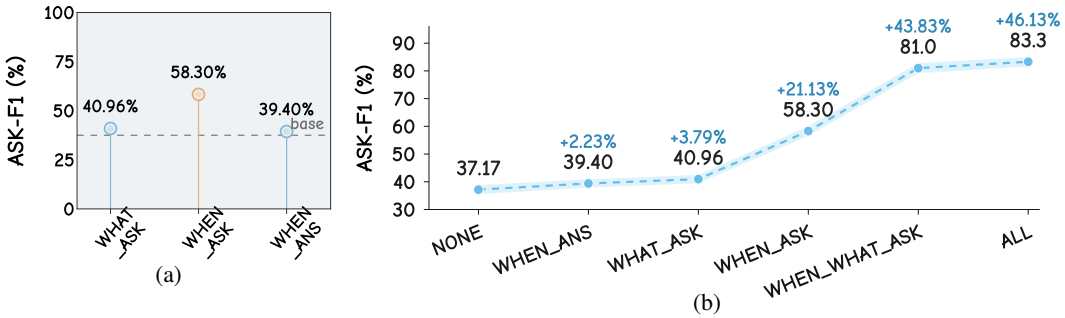


Figure 6: **Oracle ablation of decision stages.** Results on KVEM using Gemini-3.8 Flash in the multiple-choice setting. (a) Effect of different oracle signals. (b) ASK-F1 under different amounts and combinations of oracle information.

4.3.2 The Impact of Contextual Memory

In the main setting, we do not keep interaction history across turns so that each turn can be evaluated independently. We further study whether keeping historical context helps with different types of uncertainty. As shown in Figure 7(a), contextual memory has little effect on **video uncertainty**, because the model mainly relies on the visual evidence in the current turn. In contrast, it greatly improves performance on **user uncertainty**, with gains of 4.40 and 5.38 points on AUQ and UPM, respectively, because user intent and preferences often continue across turns, and previous interactions provide useful information for understanding the current request. These results show that the benefit of contextual memory strongly depends on the source of uncertainty.

4.3.3 The Effect of Ask Budget

In real-world interactions, a model may struggle to obtain sufficient information through a single ask. Therefore, we further investigate whether increasing the ASK budget can improve the model's proactive information seeking capability. As shown in Figure 7(b), performance improves as the ASK budget increases from 1 to 16 but gradually saturates at around 52.35%. This suggests that more ASK opportunities are helpful, but the improvement is limited.

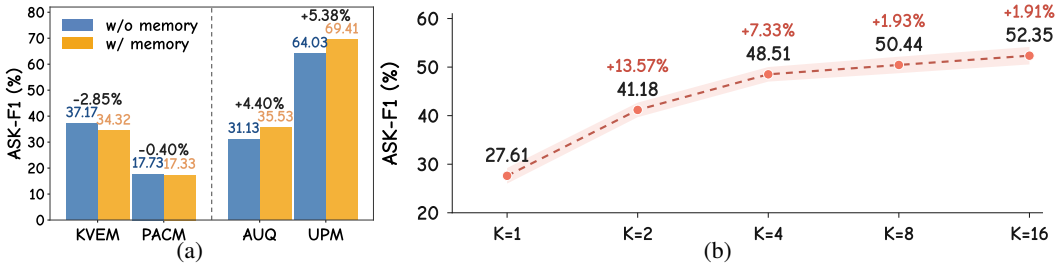


Figure 7: **Analysis Experiments.** (a) Impact of contextual memory. We use Gemini-3.8 Flash under the multiple-choice setting on four subtasks, KVEM, PACM, AUQ, and UPM, to analyze the effect of cross-turn contextual memory. (b) Impact of ASK budget. We evaluate Gemini-3.8 Flash on KVEM under the open-ended setting to analyze the effect of different ASK budgets.

5 Conclusion

We introduce **AskStream**, a benchmark for proactive information seeking under uncertainty in streaming video understanding. By extending the proactive paradigm with an explicit *ask* action, AskStream evaluates when and what models should ask before answering across eight subtasks covering video and user uncertainty. We further introduce frame-level action windows, Ask-F1, and StreamHarness to support timing-aware evaluation under both settings. Experiments show that current MLLMs remain limited in proactive information seeking, with the main bottleneck lying in determining *when and what to ask*.

AI USE STATEMENT

We used generative AI models as part of the benchmark construction, evaluation, and manuscript preparation. Specifically, GPT-5.6 Sol was used to generate candidate ASK actions and ANSWER options, which were subsequently manually verified, and Gemini-3.8 Flash was used as the judge for open-ended evaluation, with its judgments further validated against human annotations. Generative AI tools were also used for limited language editing and drafting assistance. All AI-assisted outputs were reviewed and revised by the authors, who take full responsibility for the experimental results, scientific claims, and final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the task formulation, Ask-F1 metric, task taxonomy, benchmark construction, StreamHarness, and evaluation protocols in Section 3 and Section 4. Additional reproducibility details are provided in the appendix, including task-specific data construction and human verification procedures A.7 and prompts A.8. Together, these materials document the data construction, annotation, and evaluation procedures required to reproduce the reported results.

REFERENCES

Anthropic. Claude sonnet 5 system card. Technical report, Anthropic, 2026. URL https://www.anthropic.com/news/claude-sonnet-5.

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. Streamready: Learning what to answer and when in long streaming videos, 2026. URL https://arxiv.org/abs/2603.08620.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 18407–18418. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01742. URL https://doi.org/10.1109/CVPR52733.2024.01742.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *CoRR*, abs/2604.27393, 2026. doi: 10.48550/ARXIV.2604.27393. URL https://doi.org/10.48550/arXiv.2604.27393.

Thomas De Min, Subhankar Roy, Stéphane Lathuilière, Elisa Ricci, and Massimiliano Mancini. Proactivebench: Benchmarking proactiveness in multimodal large language models. *arXiv preprint arXiv:2603.19466*, 2026.

DeepSeek-AI, :, Anyi Xu, B. Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, C. C. Yu, Chao Jin, Chaofan Lin, Chen Dong, Chenbing Wang, Chenfan Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyuan Zhang, Chenhao Xu, Chenqi Zhao, Chenze Shao, Chuhao Wang, Chuqi Zhang, Damai Dai, Dejian Yang, Deli Chen, Di Huang, Di Wu, Donghao Li, Erhang Li, Eric Fu, F. Zhou, Fangwei Zhou, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanglin Li, Guanting Chen, Guoai Cao, Guofan Fan, Guolai Meng, Guowei Li, Haichuan Zhang, Haiyang Ma, Haiyang Shen, Han Li, Han Yu, Han Zhang, Hangyuan Deng, Hanwei Xu, Hanxiang Xu, Hanxun Zhong, Hao Guo, Hao Jiang, Hao Li, Hao Qin, Haodong Wen, Haofen Liang, Haofeng Huang, Haohua Liu, Haoling Zhang, Haoming Luo, Haoran Yang, Haotian Xu, Haotian Yuan, Haoting Huang, Haowen Luo,

Haoyang Cai, Haoyu Chen, Haozhe Ji, Hengran Zhang, Hengrui Wang, Hengxu Wu, Honghui Ding, Hongxuan Tang, Huadong Wang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J. Yang, J. H. Jin, J. H. Zhang, J. X. Zou, Jia Yu, Jiahui Zhou, Jiajun Chen, Jialiang Huang, Jialin Zhao, Jiamin Tang, Jian Zhou, Jianan Tong, Jianwen Li, Jiaqi Zhu, Jiarui Wang, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiaying Ding, Jibai Lu, Jiewen Hu, Jin Yan, Jincheng Zhai, Jingchang Chen, Jingcheng Hu, Jingli Zhou, Jingsheng Xu, Jingting Xiang, Jingyan Yun, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jinpeng Wang, Jinyi Chen, Jinyi Hu, Jiping Yu, Jueliang Guo, Junbo Pei, Junbo Sun, Junguang Jiang, Junjie Qiu, Junkang Zhou, Junqi Liu, Junren Li, Junxian Li, Junxiao Song, Junyi Guo, Kai Dong, Kaifeng Chen, Kaige Gao, Kang Guan, Kangdong Yuan, Ke Hong, Ke Xu, Kefan Zhao, Kexin Ji, Kexin Zhang, Kexing Zhou, Kuai Yu, Lan Zhang, Lean Wang, Lecong Zhang, Lei Wang, Letian Gao, Liang Zhao, Liansheng Xu, Lihua Guo, Lingxiao Luo, Lingyue Fu, Litao Deng, Litong Wang, Liyue Zhang, Longhao Chen, Lu Chen, Luotian Huang, Luyao Ma, Luyao Wang, M. S. Di, Max Mei, Menghao Ye, Miao Cui, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingjing Zhang, Mingqi Wei, Mingshu Chen, Mingxing Liu, Mingxu Zhou, Mingyu Xu, Mingyu Yang, Mingze Wang, Muyang Chen, Ni Shentu, Ning Wang, Niufang Ning, Panpan Huang, Peixin Cong, Peiyi Wang, Peiyuan Xin, Pengfei Ren, Pengfei Yan, Pengle Zhang, Qi Kang, Qi Tang, Qiancheng Wang, Qiang Li, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qizhou Guo, Rongxian Xu, Rui Ding, Rui Hu, Rui Tian, Rui Yu, Ruidong Zhu, Ruifan Xu, Ruihan Yang, Ruihang Xia, Ruijie Lu, Ruilin Geng, Ruipeng Hong, Ruiqi Ge, Ruisong Zhang, Ruize Sun, Ruizhe Pan, Runji Wang, Runqian Chen, Runxin Xu, Ruohong Tian, Ruomeng Shen, Ruoyu Zhang, Ryan X., S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoyuan Chen, Shengding Hu, Shengkai Lin, Shengwen Ran, Shengyu Liu, Shengyuan Jia, Shi Bai, Shi Feng, Shicheng Xu, Shichun Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shiyuan Feng, Shufan Gong, Shuhan Lin, Shuiping Yu, Shunfeng Zhou, Shuo Yang, Shuomeng Wang, Shuting Guo, Shuting Pan, Shuying Yu, Sinuo Cao, Siyi Lin, Sizhe Chen, Songyang Chen, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tongrui Xiong, Wangding Zeng, Wei Liu, Wei Zhang, Weibin Xu, Weihao Zeng, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Shao, Wenkai Yang, Wenli Zhang, Wenlu Wang, Wenlve Huang, Wenqian Yan, Wentao Zhang, Xi Gao, Xiang He, Xiang Li, Xiangli Li, Xiangwen Wang, Xiangying Zhang, Xiankui Wei, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojian Qu, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaoyao Zou, Xiaoyuan Li, Xicheng Guo, Xieting Chu, Xin Cheng, Xin Liu, Xin Xie, Xinbo Xu, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xintong Yao, Xinyang Chen, Xinyong Jiang, Xinyu Yang, Xinyu Yang, Xu Chen, Xuanyu Wang, Xubei Zhong, Xuecheng Su, Xuejie Liu, Xuheng Lin, Xujie Fan, Xuncheng Zhao, Xuwei Fu, Y. C. Yan, Y. H. Jiang, Y. T. Wu, Y. W. M., Y. Z. Wang, Yafei Gao, Yang Yang, Yang Zhang, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Meng, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yaoyang Ye, Yehang Yin, Yexinrui Wu, Yi Qian, Yi Tao, Yi Yu, Yichao Zhang, Yichen Jiang, Yicheng Wang, Yifan Ding, Yifan Shi, Yifeng Peng, Yifeng Zhai, Yijia Wu, Yiliang Xiong, Yilun Wang, Ying He, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yiping Wang, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyao Yang, Yiyuan Liu, Yizai Cai, Yizhen Wei, Yizhi Wang, Yonglun Yang, Yongqi Zhuo, Yongqiang Guo, Yongtong Wu, Yu Wu, Yu Zhang, Yuan Bian, Yuan Cheng, Yuan Ou, Yuan Sun, Yuanfan Xu, Yuanhang Sun, Yuanhao Li, Yuchen Liu, Yuchen Yao, Yudong Han, Yuduan Wang, Yuhan Wu, Yuhao Meng, Yuheng Zou, YuKun Li, Yunchuan Wang, Yunfan Xiao, Yunfan Xiong, Yupeng Chen, Yuqian Cao, Yuqian Wang, Yuqing Chen, Yushun Zhang, Yutong Lin, Yuwei Xiao, Yuxian Gu, Yuxiang Chen, Yuxiang Huang, Yuxiang Luo, Yuxiang You, Yuxin Chen, Yuxin Xiang, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhe Guo, Yuzhen Huang, Yuzhuo Bai, Z. Y. Z., Zanlin Ni, Zehao Wang, Zehua Zhao, Zehui Ren, Zejun Zhao, Zhangli Sha, Zhanying Wang, Zhaochen Zhang, Zhaoshuai Du, Zhe Fu, Zhean Xu, Zhenda Xie, Zheng Liu, Zhengyan Zhang, Zhenhua Dong, Zhewen Hao, Zhibang Wang, Zhibin Gou, Zhicheng Ma, Zhihao Li, Zhihong Shao, Zhihuan Huang, Zhijie Li, Zhirui Lu, Zhixian Huang, Zhixuan Chen, Zhixuan Chen, Zhixuan Pan, Zhiyu Wu, Zhizhou Ren, Zhu He, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zili Zhang, Zilin Li, Zilong Hou, Zilong Lyu, Ziqiao Wang, Ziwei Xie, Ziya Zhang, Ziyi Gao, Zizheng Pan, Zonglin Li, Zongqing Yao, Zui Chen, Zuofan Wu, Chenchen Ling, Chengyu Hou, Chong Chen, D. Li, Di Qi, Dongjie Ji, Fang Wei, Fanyi Xia, Fei Xie, Feiyi Tan, Hailong Guo, Haiyan Zhai, Hui Zhou, Huihui Tan, Huijie Li, Jia Luo, Jia Song, Jialu Cai, Jian Liang, Jiangting Zhou, Jiaqi Gao, Jiayi Shao, Jie Chen, Jieyu Yang, Jin Chen, Jingde Zhang, Jingzi Zhou, Jinqian Wang, Jinyang Liu, JinZhao Sun, Junhua Ling, Junmin Zheng, Kaicheng Yang, Ke Xu, Le Su, Leyi Xia,

Liangfeng Ding, Lin Zhuo, Linwang Ma, Linyan Zhu, Liyu Cai, Luqi Yao, M. K. Zhang, Meng Li, Miao Lin, Miaojun Wang, Min Zhang, Mingming Li, Mingming Wang, Mingze Yin, Minmin Han, Nan Cao, Ning Wang, Ningxin Ma, Panpan Wang, Peihan Lin, Peng Sun, Peng Zhang, Qian Ying, Qiang Xiang, Qiao Wang, Qingmiao Mao, Qiwei Jiang, Rongli Jin, Ruyi Chen, Sha Tao, Shangmian Sun, Shaoqing Wu, Shichao Zou, Si Lei, Tianyang Zhang, Tianyu Sun, Tingting Yin, W. L. Xiao, Wei An, Wei Li, Wei Wang, Weiwei Lin, Wenqing Hou, X. Lin, Xiangfei Meng, Xianzhu Huang, Xiao Peng, Xiaoqian Li, Xiaoting Zhang, Xiaowen Sun, Xiaoxiang Wang, Xiaoyu Ye, Xinrou Zhang, Xinyu Zhang, Xue Cao, Xueyin Chen, Yanan Zhou, Yanhong Xu, Yao Xia, Yao Xu, Yi Shao, Yihong Zhang, Yiling Ma, Ying Tang, Yining Lou, Yiru Chen, Yishi Piao, Yixuan Chen, Yong Xiong, Yuchen Xuan, Yuehan Yang, Yuer Xu, Yukun Zha, Yunxian Ma, Yuping Lin, Yuting Yan, Yutong Xie, Yuwen Sheng, Yuxuan Zhu, Zekai Zhang, Zhe Ju, Zhenzhen Lin, Zheren Gao, Zheyang Sun, Zhigang Yan, Zhongyu Wu, Zi Wang, Zihua Qu, Ziling Yan, and Ziyi Wan. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026. URL https://arxiv.org/abs/2609.19969.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 24108–24118. Computer Vision Foundation / IEEE, 2025a. doi: 10.1109/CVPR52734.2025.02245. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21778–21788. IEEE, 2025b.

Google DeepMind. Gemini 3.8 flash. Model Card, September 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. URL https://arxiv.org/abs/2311.18259.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. *arXiv preprint arXiv:2603.12262*, 2026.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338. IEEE, 2025.

Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. Teaching vision-language models to ask: Resolving ambiguity in visual questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3619–3638, 2025.

Kimi Team. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Jinzhao Li, Yinuo Chen, Wenxuan Song, Yijia Lei, Yichi Zhang, Honglei Yan, Panwang Pan, and Miao Liu. Ipibench: Evaluating interactive proactive intelligence of mllms under continuous streams, 2026. URL https://arxiv.org/abs/2605.27074.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024.

Jingli Lin, Chenming Zhu, Runsen Xu, Xiaohan Mao, Xihui Liu, Tai Wang, and Jiangmiao Pang. Ost-bench: Evaluating the capabilities of mllms in online spatio-temporal scene understanding. *Advances in Neural Information Processing Systems*, 38, 2026a.

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151. IEEE, 2026b.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. AURA: always-on understanding and real-time assistance via video streams. *CoRR*, abs/2604.04184, 2026a. doi: 10.48550/ARXIV.2604.04184. URL https://doi.org/10.48550/arXiv.2604.04184.

Xudong Lu, Huankang Guan, Yang Bo, Jinpeng Chen, Xintong Guo, Shuhan Li, Fang Liu, Peiwen Sun, Xueying Li, Wei Zhang, et al. Phostream: Benchmarking real-world streaming for omnimodal assistants in mobile scenarios. *arXiv preprint arXiv:2601.22575*, 2026b.

Junbo Niu, Yifei Li, 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? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913. IEEE, 2025.

OpenAI. GPT-5.6 System Card. System Card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, Fan Zhou, Hao Luo, Haofeng Huang, Haoran Lian, Haoyan Huang, Hongqing Chen, Jianwei Zhang, Jing Xu, Junjie Wang, Langshi Chen, Liangyu Wang, Linlang Jiang, Man Yuan, Minmin Sun, Peng Jin, Siqi Zhang, Siyu Wang, Xingzhang Ren, Yakai Wang, Yi Zhang, Yiming Dong, Yizhong Cao, Yubo Ma, Yunfei Mao, Bo Zheng, and Dayiheng Liu. On the design of qwen3.8-next architecture: Evaluation, efficiency, and training stability, 2026. URL https://arxiv.org/abs/2608.30320.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026.

Yansong Shi, Qingsong Zhao, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. RIVER: A real-time interaction benchmark for video llms. *CoRR*, abs/2603.03985, 2026. doi: 10.48550/ARXIV.2603.03985. URL https://doi.org/10.48550/arXiv.2603.03985.

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, 2019. URL https://arxiv.org/abs/1903.02874.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Yean Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Shuhao Guan, Yuanying Guo, Xiaoru Hao, Dailan He, Tianhong He, Weiran He, Wenyang He, Yibo He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Chaobo Jia, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, G. Luo, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zifan Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Xiaofei Yang, Xinlong Yang, Xinyu Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhuorui Ye, Peng Yebo, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026a. URL https://arxiv.org/abs/2602.02276.

Xiaomi MiMo Team, Anqi Liu, Aoxin Ma, Bo Chen, Bo Yang, Chen Wang, Chen Zhang, Chengda Tang, Chengwei Wang, Chiheng Lou, Depeng Yan, Fuli Luo, Gang Wang, Hailin Zhang, Jiale Sun, Kang Zhou, Rui Huang, Shaohui Liu, Shen Huang, Shijie Cao, Shuaishuai Fan, Tianling Zhou, Xiangwei Deng, Xueyang Xie, Xuli Wang, Yingchun Lai, Yu Yang, Yuan Zhang, Zhen Tang, Zhonghua Deng, and Zihan Jiang. Full-pipeline inference optimization for mimo-v2.5 series: Pushing hybrid swa efficiency to the limit, 2026b. URL https://arxiv.org/abs/2607.13095.

Lu Wang, Zhuoran Jin, Yupu Hao, Yubo Chen, Kang Liu, Yulong Ao, and Jun Zhao. Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models. *arXiv preprint arXiv:2603.11896*, 2026a.

Pengyu Wang, Chenkun Tan, Shaojun Zhou, Qirui Zhou, Yanxin Chen, Xingyang He, Huazheng Zeng, Jijun Cheng, Chenghao Wang, Xiaomeng Qian, Pengfei Wang, Zhan Huang, Shanqing Gao, Wei Huang, Longjun Cao, Wu Ran, Jie Liu, Changtai Zhu, Hongkai Wang, Yixian Tian, Chenghao Liu, Zhen Ye, Xinghao Wang, Botian Jiang, Guoguo Feng, Zhaoye Fei, Ruixiao Li, Mingshu Chen, Yang Gao, Qinyuan Cheng, Shimin Li, and Xipeng Qiu. MOSS-VL technical report. *CoRR*, abs/2608.15045, 2026b. doi: 10.48550/ARXIV.2608.15045. URL https://doi.org/10.48550/arXiv.2608.15045.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. Lvbench: An extreme long video understanding benchmark. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 22958–22967. IEEE, 2025a. doi: 10.1109/ICCV51701.2025.02131. URL https://doi.org/10.1109/ICCV51701.2025.02131.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18523–18533, 2026c.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. Mmduet2: Enhancing proactive interaction of video mllms with multi-turn reinforcement learning. *CoRR*, abs/2512.06810, 2025b. doi: 10.48550/ARXIV.2512.06810. URL https://doi.org/10.48550/arXiv.2512.06810.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models, 2025c. URL https://arxiv.org/abs/2507.09313.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. Omni-immi: A comprehensive multi-modal interaction benchmark in streaming video contexts. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18925–18935. IEEE, 2025d.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In *International Conference on Learning Representations*, volume 2026, pp. 61463–61475, 2026.

Shuhang Xun, Sicheng Tao, Jungang Li, Yibo Shi, Zhixin Lin, Zhanhui Zhu, Yibo Yan, Hanqian Li, Linghao Zhang, Shikang Wang, et al. Rtv-bench: Benchmarking mllm continuous perception, understanding and reasoning through real-time video. *Advances in Neural Information Processing Systems*, 38, 2026.

Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Tao Jin, Xing Xie, Hao Liao, and Jianxun Lian. Proact-vl: A proactive videollm for real-time ai companions. *arXiv preprint arXiv:2603.03447*, 2026.

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. In *International Conference on Learning Representations*, volume 2025, pp. 42522–42556, 2025.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Haowen Hou, Zheming Liang, Congcong Wang, Yuhang Cao, Shenglong Ye, Shuai Xie, Shuhuan Gu, Haoyang Huang, Qingyi Si, Nan Duan, and Jiaqi Wang. Joyai-vl-interaction: Real-time vision-language interaction intelligence. *CoRR*, abs/2606.14777, 2026. doi: 10.48550/ARXIV.2606.14777. URL https://doi.org/10.48550/arXiv.2606.14777.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 10807–10816. ACM, 2025. doi: 10.1145/3746027.3754839. URL https://doi.org/10.1145/3746027.3754839.

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 preprint arXiv:2406.08085*, 2024.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026a.

Jialiang Zhang, Junlong Tong, Junyan Lin, Hao Wu, Yirong Sun, Yunpu Ma, and Xiaoyu Shen. Think-as-you-see: Streaming chain-of-thought reasoning for large vision-language models. *arXiv preprint arXiv:2603.02872*, 2026b.

Xichen Zhang, Guankai Li, Yinghao Zhu, Shijian Wang, Sitong Wu, Shaozuo Yu, Meng Chu, Yuan Lu, and Jiaya Jia. Streamarena: Toward continuous, interactive, and long-horizon agentic streaming video understanding. *arXiv preprint arXiv:2608.05703*, 2026c.

Yuanhong Zheng, Ruichuan An, Xiaopeng Lin, Yuxing Liu, Sihan Yang, Huanyu Zhang, Haodong Li, Qintong Zhang, Renrui Zhang, Guopeng Li, Yifan Zhang, Yuheng Li, and Wentao Zhang. Pearl: Personalized streaming video understanding model, 2026. URL https://arxiv.org/abs/2603.20422.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

A Appendix

A.1 Limitations

AskStream still simplifies several aspects of real-world proactive interaction. First, the current setting allows at most one *ASK* action per turn, while practical scenarios may require repeated clarification, especially when multiple entities are simultaneously ambiguous or when one ASK reveals further missing information. Second, our benchmark focuses on streaming video understanding and does not consider embodied agents that must jointly coordinate perception, information seeking, and physical actions, such as vision-language-action (VLA) models. Third, each interaction is initiated by an explicit user query, whereas more autonomous agents should be able to proactively detect information insufficiency even without being directly asked and decide whether clarification is needed. Finally, although AskStream introduces frame-level temporal windows, real-world uncertainty may evolve over longer and more complex horizons, with overlapping, discontinuous, or dynamically changing opportunities for ASK and ANSWER. Extending the benchmark along these directions would enable more realistic evaluation of proactive information seeking.

A.2 METRIC DESIGN

In this section, we further discuss the design of **Ask-F1** and analyze its sensitivity to different temporal weighting functions.

Why harmonic aggregation? AskStream evaluates two coupled stages of proactive information seeking: (1) whether a model can correctly and timely acquire missing information through an `ASK` action, and (2) whether it can subsequently provide a correct and timely `ANSWER` after the requested information becomes available. We measure these two capabilities using S_{ask} and S_{ans}, respectively, and aggregate them using

$$\text{Ask-F1} = \frac{2S_{\text{ask}}S_{\text{ans}}}{S_{\text{ask}} + S_{\text{ans}}}. \tag{5}$$

We use the term *F1* to denote an **F1-style harmonic aggregation**, rather than the conventional classification F1 defined from precision and recall. The harmonic mean is suitable for our setting because successful proactive interaction requires both information acquisition and subsequent answering to perform well. A model should not obtain a high overall score if it answers accurately only after the missing information has been provided but rarely identifies when and what to ask. Conversely, issuing appropriate ASK actions is insufficient if the model cannot subsequently answer the user's query. Compared with an arithmetic mean, the harmonic mean more strongly penalizes imbalance between these two stages.

Why is S_{ans} conditional on a correct ASK? We define S_{ans} over turns in which the model has successfully issued a correct ASK. This conditioning is intentional because it separates failures in *information acquisition* from failures in *reasoning after information acquisition*. Without this conditioning, an incorrect or missed ASK prevents the model from obtaining the required information and therefore necessarily affects the subsequent ANSWER. Counting such a turn again as an ANSWER failure would conflate the two decision stages and make it difficult to identify the actual source of failure. Accordingly, S_{ask} measures how reliably the model acquires the required information, while S_{ans} measures whether the model can correctly use that information after successful acquisition. When $N_{\text{ask}} = 0$, we define $S_{\text{ans}} = 0$ and consequently $\text{Ask-F1} = 0$. Importantly, S_{ans} is interpreted jointly with S_{ask}: even if the conditional answer score is high, a small S_{ask} directly limits the final harmonic score.

Temporal weighting. Besides content correctness, proactive interaction is inherently time-sensitive. For an action with valid temporal window $W = [s, e)$, our default metric uses the linear temporal weighting function

$$T_{\text{lin}}(t; [s, e)) = \mathbb{I}[s \le t < e] \left(1 - \frac{t - s}{e - s}\right). \tag{6}$$

The valid window first determines whether an action is temporally acceptable, while the decay term further distinguishes different latencies within that window. Once the beginning of a valid window is reached, sufficient information has become available for the corresponding action. Therefore, unnecessarily delaying an otherwise correct action provides no additional utility in a streaming interaction. We choose linear decay because it is bounded, parameter-free, and easy to interpret. An action at the beginning of the valid window receives maximum temporal credit, while the reward decreases smoothly toward the end of the window. However, linear decay is not the only reasonable temporal utility function. We therefore examine whether the conclusions of AskStream remain stable under alternative temporal weighting schemes.

Alternative temporal weighting functions. We examine whether Ask-F1 is sensitive to the particular temporal weighting function used within a valid action window. Let $u = (t - s)/(e - s) \in [0, 1)$ denote the normalized position of an action within the window. In addition to the default linear weighting, we consider two nonlinear variants. The first is a logarithmic weighting,

$$T_{\log}(t;[s,e)) = \mathbb{I}[s \le t < e] \frac{\log(2-u)}{\log 2}, \tag{7}$$

which decays more slowly than the linear function and therefore assigns relatively more credit to later but still valid actions. The second is an exponential weighting,

$$T_{\exp}(t;[s,e)) = \mathbb{I}[s \le t < e] \exp(-2u), \tag{8}$$

which places a stronger nonlinear preference on earlier actions. We use a fixed decay coefficient of 2 as a simple stress-test setting rather than tuning it to individual models. In particular, the reward decreases to $e^{-2} \approx 0.135$ near the end of the valid window, providing a substantially stronger timing penalty while still retaining nonzero credit for late but valid actions. Table 3 reports the results for the seven API-evaluated models in the multiple-choice setting. Although the absolute scores vary with the weighting function, the model ordering remains unchanged across all three variants, indicating that the comparative results are not sensitive to the precise form of the within-window temporal decay.

Table 3: **Sensitivity of Ask-F1 to alternative temporal weighting functions.** Results are reported for the seven models under the multiple-choice setting. Linear is the default metric, while logarithmic and exponential replace its within-window temporal weighting function. Spearman ρ w.r.t. Linear measures the rank correlation of model rankings relative to the default Linear metric, where $\rho = 1.00$ indicates identical rankings.

Method	Logarithmic	Linear	Exponential
Offline closed-source			
GPT-5.6-Sol	22.04	20.50	16.65
Claude Sonnet 5	24.16	22.48	18.27
Gemini-3.8 Flash	**56.11**	**52.18**	**42.34**
Offline open-source			
MiMo-V2.5	15.81	14.66	11.82
Kimi K2.6	17.70	16.52	13.51
Qwen3.8-Flash	18.78	17.60	14.52
DeepSeek-V4.1-Flash	21.16	19.69	16.01
Spearman ρ w.r.t. Linear	**1.00**	**1.00**	**1.00**

Multiple-ASK protocol. The main evaluation allows at most one ASK action per turn. To study whether additional opportunities for information seeking improve performance, we further consider a multiple-ASK protocol with an ASK budget $B \in \{1, 2, 4, 8, 16\}$. Under a budget of B, the model may issue up to B ASK actions before producing its final ANSWER. Each ASK is evaluated at the time it is issued using the same temporal and content criteria as in the main setting. A valid ASK triggers the corresponding user or visual feedback, after which the model continues the interaction with the newly acquired information. An invalid or redundant ASK consumes one unit of the ASK budget but does not immediately terminate the turn, allowing the model to recover through subsequent interactions. Once the ASK budget is exhausted, the model may continue to WAIT or produce its final ANSWER, while the ANSWER evaluation remains unchanged. This protocol isolates the effect of repeated information-seeking opportunities without modifying the underlying task or answer criterion.

A.3 Uncertainty Analysis

Experimental configuration. We evaluate Qwen3.5 9B on 130 five choice events from 78 videos to examine whether obtaining the information requested by an *ask* action changes the model's belief over possible answers. At each checkpoint, the model receives ten chronological frames sampled at 1 Hz from the causal history ending at that checkpoint, with six checkpoints before asking and six

after the reveal. The correct evidence is revealed one second after asking. The **Mismatched Reveal** condition follows the same temporal alignment but reveals an irrelevant entity from another event in the same video, while **Blurred** masks the task relevant entity and **Full** uses the original clear video. Let x denote the ten frame input and let $\{a_1, \ldots, a_5\}$ denote the five complete answer texts. For $a_j = (y_{j,1}, \ldots, y_{j,L_j})$, we compute its mean token log probability as

$$s_j(x) = \frac{1}{L_j} \sum_{t=1}^{L_j} \log p_\theta(y_{j,t} \mid x, y_{j,<t}).$$

We then normalize the five scores into an answer distribution

$$q_j(x) = \frac{\exp(s_j)}{\sum_{k=1}^{5} \exp(s_k)}.$$

Model uncertainty is measured using normalized entropy

$$H(q) = \frac{1}{\log 5} \sum_{j=1}^{5} q_j \log \frac{1}{q_j}.$$

Entropy across interaction turns. Figure 2(a) shows the evolution of normalized answer entropy across three interaction turns. Before the requested evidence appears, the model remains relatively uncertain. Once the correct evidence becomes available, normalized entropy consistently decreases across all three turns. In contrast, the **Mismatched Reveal** condition remains close to its previous uncertainty level. This comparison shows that the reduction in uncertainty is associated with the relevance of the revealed evidence rather than with additional visual observations or elapsed time alone.

Answer probability evolution. Figure 2(b) visualizes how the answer distribution changes as visual evidence accumulates over time. Increasing support for the gold entity raises the probability assigned to the gold answer while reducing the probabilities assigned to the four incorrect alternatives. When the revealed evidence is irrelevant to the current question, the same separation does not emerge. The probability evolution therefore shows that relevant evidence specifically shifts the model's belief toward the correct answer.

Final answer quality. Figure 2(c) summarizes the final answer performance. With blurred input, the model achieves 42.3% accuracy and assigns an average probability of 25.8% to the gold answer. After asking for and receiving the key evidence, accuracy increases to 62.3% and the mean gold answer probability rises to 36.8%. These results approach the fully visible condition, which achieves 63.1% accuracy and a mean gold answer probability of 36.6%. Together, the three panels show that obtaining the requested evidence reduces model uncertainty and improves answer quality by shifting belief toward the correct answer.

A.4 Open-Ended Evaluation Results

Table 6 reports the results under the open-ended setting. Overall, performance drops substantially compared with the multiple-choice setting, indicating that generating appropriate ASK and ANSWER responses is considerably more challenging than selecting from predefined candidates. Gemini-3.8 Flash achieves the strongest overall performance with an Ask-F1 of **29.19%**, while its S_{ask} (23.34%) remains notably lower than S_{ans} (38.97%), again suggesting that proactive information acquisition is a major bottleneck. Open-source offline models generally achieve lower scores, and streaming proactive models struggle even more severely: most obtain zero on nearly all subtasks, with AURA reaching only 4.01% Ask-F1. These results further demonstrate that current models have limited ability to reliably formulate and time proactive information-seeking actions in unconstrained open-ended interactions.

A.5 Validation of the Open-Ended Judge

To validate the reliability of Gemini-3.8 Flash as the judge in the open-ended evaluation, we randomly sample 300 turns and ask two human annotators to independently assess the correctness

of both ASK and ANSWER responses, with disagreements resolved by an additional annotator. We compare inter-annotator agreement and the agreement between the Gemini judge and the human consensus, and separately report results on **Gemini generated** and **non Gemini generated** responses to examine potential self-preference. As shown in Table 4, the Gemini judge exhibits high agreement with human judgments for both ASK and ANSWER. Moreover, its agreement on Gemini generated responses is comparable to that on non Gemini responses, providing no clear evidence of self-preference within the sampled data.

Table 4: Validation of Gemini-3.8 Flash as the open-ended judge on 300 sampled turns. **Human&Human** reports the agreement between two independent human annotators and serves as a reference for annotation consistency. **ASK Agreement** and **ANSWER Agreement** denote the percentage of identical correctness judgments for ASK and ANSWER responses, respectively. **Overall Agreement** is computed over both ASK and ANSWER judgments. For the Gemini Judge vs. Human Consensus rows, human consensus is used as the reference judgment. Results are further separated into Gemini-3.8 Flash and non Gemini outputs to examine potential self-preference.

Comparison	**ASK Agreement (%)**	**ANSWER Agreement (%)**	**Overall Agreement (%)**
Human&Human	95.33	97.00	96.17
Gemini Judge vs. Human Consensus			
All outputs	**92.67**	**94.33**	**93.50**
Gemini-3.8 Flash outputs	93.06	94.44	93.75
Non Gemini outputs	92.31	94.23	93.27

A.6 StreamHarness Ablation

We compare StreamHarness with three simpler context strategies using Gemini-3.8 Flash under the multiple-choice setting. The sliding-window baseline keeps only the most recent 10 frames and removes the mid- and long-term memory, while the other two baselines uniformly sample 64 or 128 frames from the observed video. For uniform sampling, we ensure that each valid temporal window in the observed video contains at least one sampled frame, while the temporal window boundaries are not provided to the model. As shown in Table 5, removing the hierarchical memory reduces Ask-F1 from 52.18% to 41.70%. Uniform sampling leads to a larger performance drop, reaching 26.72% with 128 frames and 16.99% with 64 frames. These results show that preserving dense recent observations together with compressed historical context is important for proactive ASK and ANSWER decisions in streaming video.

Table 5: Ablation of StreamHarness using Gemini-3.8 Flash under the multiple-choice setting. Uniform sampling ensures that each valid temporal window contains at least one sampled frame.

Context Strategy	S_{ask}	S_{ans}	Ask-F1
StreamHarness	**44.63**	**62.80**	**52.18**
Sliding Window	35.92	49.71	41.70
Uniform Sampling (128)	22.41	33.08	26.72
Uniform Sampling (64)	13.86	21.94	16.99

A.7 Data Construction

A.7.1 Annotation and Verification Interface.

To facilitate efficient and consistent data construction, we develop task-specific web interfaces for all eight subtasks, covering both annotation and human verification. We use KVEM as a representative example. As shown in Figure 8, the annotation interface allows annotators to review event-level video segments, inspect and refine tracked entities at keyframes, select the gold entity, and annotate the temporal decision windows associated with WAIT, ASK, and ANSWER actions. The interface also supports the annotation of proactive queries and other event-level metadata required for

Table 6: Main results on AskStream in the Open-ended setting.

Method	Backbone	Video				User				Metric		
		Explicit		Implicit		Intent		Preference				
		KVEM	VIM	ICM	PACM	AUQ	WUQ	UPM	UPC	S_{ask}	S_{ans}	**Ask-F1**
Offline closed-source models												
Claude Sonnet 5	-	13.59	17.62	0	**4.78**	12.69	12.28	0	**19.73**	15.88	10.99	12.99
GPT-5.6-Sol	-	6.30	3.90	2.80	0	16.20	9.96	7.63	16.52	12.05	18.80	14.69
Gemini 3.8 Flash	-	**27.61**	**29.40**	**35.79**	0	35.87	**32.12**	14.41	18.43	**23.34**	**38.97**	**29.19**
Offline open-source models												
MiMo-V2.5	-	10.15	9.29	2.57	0	7.86	8.90	6.63	0	4.73	25.72	7.99
Kimi K2.6	-	15.54	0	7.82	0	**41.13**	15.93	**19.31**	0	6.12	17.77	9.10
DeepSeek-V4.1-Flash	-	6.28	5.74	0	0	26.54	7.50	10.16	11.49	10.09	17.94	12.92
Qwen3.8-flash	-	6.03	10.13	1.29	0	7.30	5.52	0	0	12.54	16.03	14.07
Streaming proactive models												
MOSS-VL-Realtime	Qwen3-8B	0	0	0	0	0	0	0	0	0	0	0
MiniCPM-o 4.5	Qwen3-8B	0	0	0	0	0	0	0	0	0	0	0
MMDuet2	Qwen2.5-VL-3B	0	0	0	0	0	0	0	0	0	0	0
TimeChat-Online	Qwen2.5-VL-7B	0	0	0	0	0	0	0	0	0	0	0
VideoLLM-online	Llama-3-8B	0	0	0	0	0	0	0	0	0	0	0
JoyAI-VL-Interaction	JoyAI-VL-1.0	0	0	2.89	0	0	0	0	0	0.43	7.75	0.81
AURA	Qwen3-VL-8B	4.19	5.76	0	0	6.50	0	0	0	2.77	7.25	4.01

constructing each instance. After data construction, we conduct an additional human verification stage using the interface shown in Figure 9. For each instance, annotators jointly inspect the fully masked video and the corresponding revealed video in which only the gold entity is visible, together with the query and candidate ASK options. This dual-video design enables annotators to verify that the masked version does not leak sufficient information to uniquely answer the question, while the revealed version provides the necessary evidence to identify the correct answer. The verification interface further allows annotators to check whether the correct ASK requests the intended missing visual evidence and whether the incorrect ASK candidates correspond to irrelevant or insufficient evidence.

A.7.2 Task-specific Construction Details

KVEM. For the KVEM task, we first select videos from the COIN (Tang et al., 2019) dataset that contain diverse entities and whose key visual evidence can be reliably masked. We determine the number of interaction turns according to the step segmentation provided by the source dataset, and identify multiple key entities in each turn, which are subsequently used to construct correct and incorrect ASK candidates. To ensure reliable tracking, we manually annotate bounding boxes for the key entities in each turn and re-annotate the corresponding entities whenever a scene transition occurs, thereby reducing tracking drift or failure. Based on these annotations, we use Grounded SAM 2 to segment and track the entities across frames, and apply Gaussian blur to mask them. For each instance, we generate two video variants: one in which all key entities are masked, and another in which only the gold entity remains unmasked. The latter is returned as the visual evidence provided after a correct ASK. We then manually annotate the temporal interval during which the gold entity appears and disappears, and use these annotations to define the corresponding temporal windows. Based on the annotated windows, we manually construct proactive queries, typically using formulations such as "when ...tell me ...", as well as the correct and incorrect ANSWER candidates. Finally, each instance undergoes manual verification using both video variants: the fully masked version is inspected to ensure that no information leakage allows the question to be uniquely answered without the requested evidence, while the version with only the gold entity visible is used to verify that the revealed evidence is sufficient to uniquely determine the correct answer.

VIM. For the VIM task, we build upon the Ego-Exo4D (Grauman et al., 2024) dataset, which provides temporally synchronized egocentric and multiple exocentric video streams of the same activity. We reuse the key-step annotations provided by the source dataset to define the start and end bound-

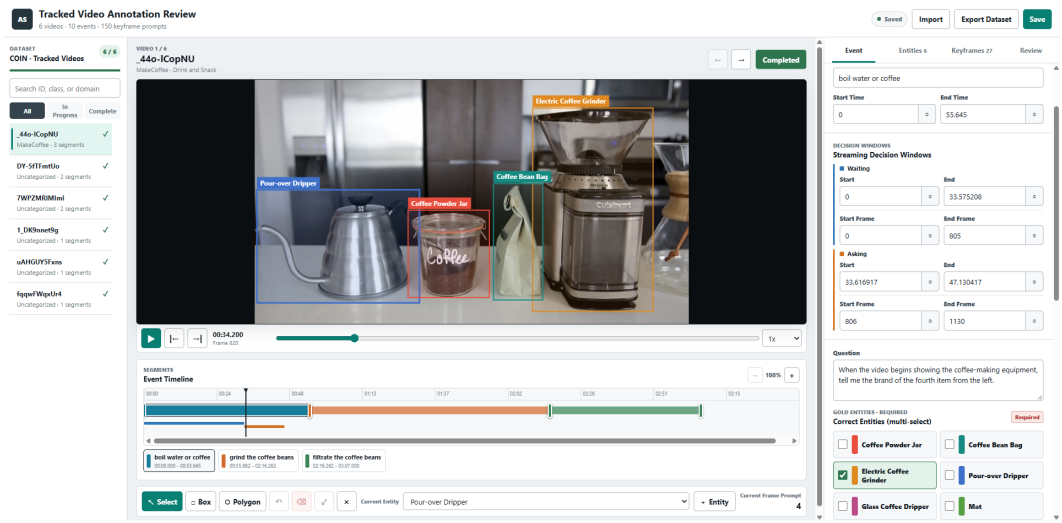


Figure 8: **Web interface for KVEM annotation.** Annotators review each event together with its tracked entities and keyframes, refine entity bounding boxes when necessary, select the gold entity, and annotate the temporal decision windows for streaming interaction. The interface also supports event-level query annotation and visualization of the corresponding timeline, facilitating consistent construction of KVEM instances.

aries of interaction turns. For each turn, annotators jointly inspect all available viewpoints using the multi-view annotation interface shown in Figure 10, and construct a query that cannot be answered from the designated initial viewpoint but can be uniquely answered from one or more alternative viewpoints. We then manually annotate the corresponding temporal decision windows according to the onset and disappearance of the query-relevant visual evidence. The ASK candidates are formulated as precise textual descriptions of different available viewpoints, such that a correct ASK requests a viewpoint that exposes the missing evidence required to answer the query, whereas incorrect ASK candidates request viewpoints that remain insufficient for resolving the information gap. After a correct ASK, the video stream is switched to the requested viewpoint, allowing the model to observe the previously missing evidence before producing its answer. Finally, each instance undergoes manual verification. Annotators first inspect the designated initial viewpoint together with the query to confirm that the answer cannot be uniquely inferred and that no relevant information is leaked. They then inspect the candidate alternative viewpoints to verify that the viewpoint associated with the correct ASK reveals sufficient evidence to uniquely determine the answer, while the viewpoints associated with incorrect ASK candidates remain insufficient or irrelevant. This procedure ensures that successful answering requires an appropriate viewpoint-switching request rather than relying on information already available from the initial stream.

ICM. For the ICM task, we build upon PEARL (Zheng et al., 2026), a streaming-video dataset with personalized concepts. We remove the original personalized-concept annotations while preserving the multi-turn interaction structure provided by the dataset and manually annotate the queries, answers, and temporal decision windows for each turn. The ASK candidates are designed to evaluate whether the model can correctly identify and request clarification for all personalized entity concepts referenced in an ambiguous query. A correct ASK must accurately describe and localize all relevant entities whose personalized concepts are required to disambiguate the user's intent, whereas incorrect ASK candidates either contain incorrect entity descriptions or provide only partial clarification that is insufficient to resolve the ambiguity. After a correct ASK, we return images of the corresponding entities as additional evidence, allowing the model to ground the previously missing personalized concepts before answering. Finally, each instance undergoes manual verification. Annotators first inspect the original interaction without the additional entity evidence to confirm that the query cannot be uniquely resolved from the available video and dialogue context alone. They then verify that the correct ASK covers all personalized entity concepts necessary for disambiguation, while each incorrect ASK is either factually inconsistent with the referenced entities or incomplete and therefore insufficient to uniquely determine the user's intended referents. After the correspond-

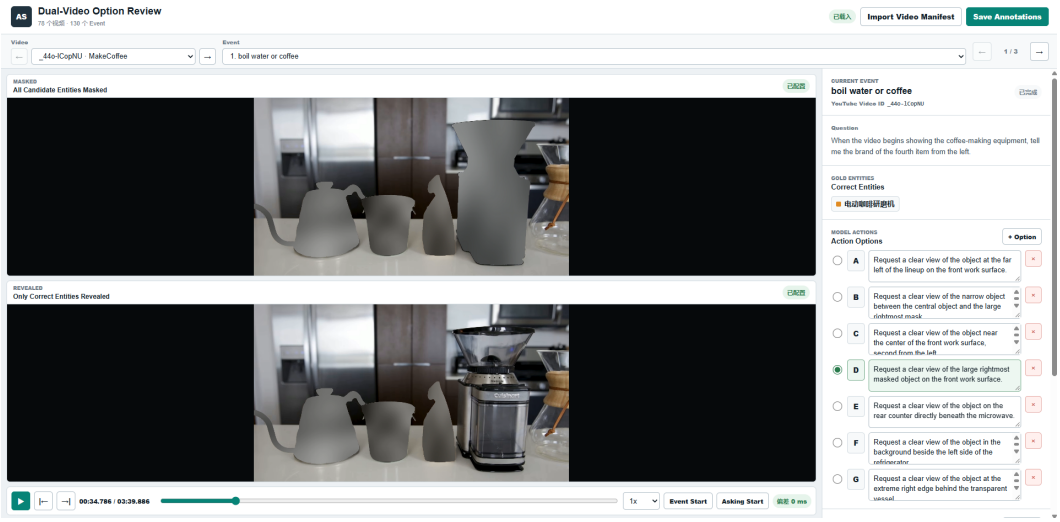


Figure 9: **Web interface for KVEM human verification.** The fully masked video and the revealed video, in which only the gold entity is visible, are presented side by side for comparison. Annotators jointly inspect the query, gold entity, and ASK candidates to verify that the masked video contains no information leakage, that revealing the requested gold entity is sufficient to uniquely answer the query, and that the correct and incorrect ASK options correspond to appropriate and inappropriate information requests, respectively.

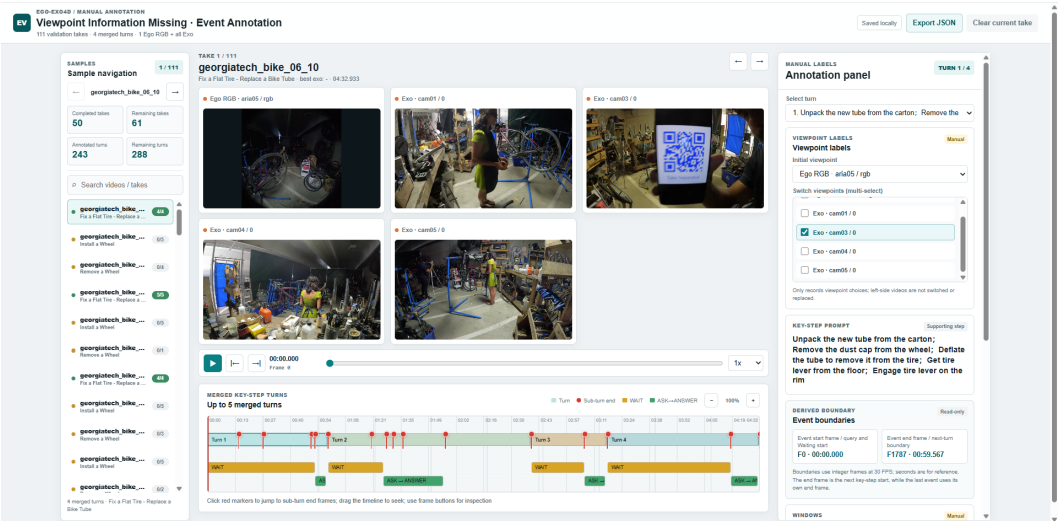


Figure 10: **Multi-view annotation interface for the VIM task.** The interface presents temporally synchronized egocentric and multiple exocentric video streams from Ego-Exo4D, together with the key-step-based turn structure and temporal decision windows. Annotators jointly inspect all viewpoints, designate the initial viewpoint, identify alternative viewpoints that reveal the missing visual evidence, and construct viewpoint-dependent queries and ASK candidates.

ing entity images are revealed, annotators further confirm that the ambiguity is resolved and that the query admits a unique and unambiguous correct answer.

PACM. For the PACM task, since personalized actions in the real world can be highly diverse and may overlap with action concepts already encoded in a model's prior knowledge, we use Blender to synthesize personalized action videos with newly defined action concepts. Specifically, we adopt existing human character models and manually design 25 atomic actions, which are then compositionally combined into diverse personalized actions and assigned novel names to minimize the

influence of model priors. We synthesize two types of videos: (1) multiple characters repeatedly perform different personalized actions in parallel, and (2) multiple characters perform personalized actions sequentially. For each target personalized action, we additionally construct several hard negative actions by modifying one of its constituent atomic actions while keeping the remaining components unchanged. For the first video type, the query follows the form "after observing the first complete execution of all characters, tell me XXX," and the valid ANSWER window starts once all characters have completed their first full execution and extends to the end of the turn. For the second video type, the query follows the form "tell me when you can determine which character is performing the correct personalized action." In this case, the ANSWER window starts when the character performing the target action has completed the action and extends to the end of the turn. ASK candidates are constructed according to the discriminative characteristics of the personalized actions: a correct ASK requests an action characteristic that can uniquely distinguish the target from the other candidates, whereas incorrect ASK candidates request characteristics that are shared by at least two characters and therefore cannot uniquely identify the target. To provide the missing personalized action concept after a correct ASK, we additionally render the corresponding action using a textureless white humanoid character and return this action video as the requested visual demonstration, thereby avoiding additional appearance-related cues. Finally, the correct and incorrect ANSWER candidates indicate which character performs the target personalized action. Finally, each instance undergoes manual verification to ensure both the validity of the ASK candidates and the answerability of the task. Annotators first verify that the correct ASK requests an action characteristic that uniquely distinguishes the target personalized action from all other candidates, while each incorrect ASK remains ambiguous and is compatible with at least two candidate actions. They then inspect the returned white-humanoid demonstration to confirm that it provides sufficient information to identify the target action without introducing additional appearance or identity cues. Finally, annotators verify that, after the requested action concept is revealed, the correct character can be uniquely identified and the corresponding ANSWER is unambiguous.

AUQ. For the AUQ task, we construct the data using videos from StreamArena (Zhang et al., 2026c), supplemented with additional videos collected from YouTube. We first use PySceneDetect to assist manual segmentation of each video into interaction turns, with all detected boundaries subsequently reviewed and adjusted by annotators. For each turn, we manually construct a video-grounded ambiguous query whose intended referent or interpretation cannot be uniquely determined from the query alone. We annotate the start of the valid ASK window as the earliest time at which the ambiguity becomes identifiable from the observed video context. The ANSWER window begins when the visual event or entity required to answer the clarified query first becomes available and ends when the corresponding evidence disappears from the stream. We construct ASK candidates to cover different possible interpretations of the ambiguity, where the correct ASK explicitly requests the clarification necessary to resolve the uncertainty, while incorrect ASK candidates either target an irrelevant source of ambiguity or fail to sufficiently disambiguate the query. After a correct ASK, the model receives the corresponding user clarification, which resolves the ambiguity and enables subsequent answering. Finally, each instance undergoes manual verification in two stages. Annotators are first presented with the original ambiguous query and the observed video context to confirm that the intended answer cannot be uniquely determined without clarification. They are then provided with the clarified query and verify that the ambiguity is fully resolved and that the available visual evidence supports a unique and unambiguous correct answer within the annotated ANSWER window.

WUQ. For the WUQ task, we construct the data using videos from StreamingBench (Lin et al., 2026b) and additional videos collected from YouTube. Similar to AUQ, we first use PySceneDetect to assist the manual segmentation of each video into interaction turns, with all automatically detected boundaries subsequently reviewed and adjusted by annotators. For each turn, we first annotate a standard proactive query associated with a visual trigger and determine the corresponding ANSWER window based on the temporal extent during which the required visual evidence is available. We then manually select an appropriate entity whose appearance makes it possible to detect an inconsistency in the user's query and annotate the onset time of this entity as the beginning of the valid ASK window. Based on the selected entity, we construct a corrupted version of the query by injecting an incorrect factual statement about the entity, such that the query contains a false premise that becomes identifiable once the relevant visual evidence appears. The model is expected to recognize and explicitly correct the erroneous premise rather than directly answering the corrupted query. After the model correctly identifies the inconsistency and provides the appropriate correction, the

user returns a confirmation acknowledging the correction, after which the interaction proceeds using the aligned information. Finally, each instance undergoes manual verification. Annotators first inspect the corrupted query together with the video to confirm that the injected statement is indeed inconsistent with the observed evidence and that the inconsistency becomes detectable no earlier than the annotated ASK window. They then verify that the proposed correction accurately resolves the false premise and that, after the correction is confirmed, the original information needed can be uniquely and unambiguously answered within the annotated ANSWER window.

UPM and UPC. For the UPM and UPC tasks, we collect preference-oriented videos from YouTube, covering domains such as clothing, furniture, and other everyday scenarios in which human preferences naturally influence the desired answer. We manually segment each video into interaction turns according to scene transitions. For each turn, annotators construct a proactive query whose answer depends on the user's preference, and identify multiple candidate entities appearing in the relevant frames These entities are used to construct the ANSWER candidates. We then randomly select one candidate as the ground-truth preferred entity and manually annotate the corresponding ASK and ANSWER windows according to the temporal availability of the query-relevant evidence. For **UPM**, the constructed instance is used to generate a set of user-preference descriptions along different preference dimensions. The correct preference is designed to uniquely identify the ground-truth entity among the available candidates, whereas the incorrect preference options are intentionally insufficient to uniquely determine that entity. After a correct ASK, the user provides the missing preference information, allowing the model to combine the revealed preference with the observed visual candidates and produce the final answer. During human verification, annotators first confirm that the query cannot be uniquely answered without additional preference information. They then verify that the preference associated with the correct ASK uniquely selects the ground-truth entity, while each incorrect preference remains compatible with multiple candidates and therefore cannot uniquely determine the answer. For **UPC**, we additionally construct scenarios in which a previously known user preference has become outdated. To make preference changes objective and interpretable rather than arbitrary, we select contexts in which preferences may plausibly change due to an observable change in user circumstances, such as moving to a new home, losing weight, or other changes that alter the suitability of previously preferred items. We therefore provide both a historical preference and an updated preference context, while constructing ASK candidates in the same manner as in UPM: the correct ASK requests the preference dimension necessary to identify the user's current preference, whereas incorrect ASK candidates fail to resolve the uncertainty. After a correct ASK, the user provides the updated preference, and the model is expected to answer according to the current rather than historical preference. Human verification follows the same uniqueness checks as in UPM, with an additional consistency check that annotators can recognize from the provided context that the historical preference is no longer sufficient or applicable, and that obtaining the updated preference is necessary to uniquely determine the correct answer.

A.7.3 Temporal Window Annotation Consistency.

Because Ask-F1 directly relies on the annotated temporal decision windows, we further evaluate the consistency of our temporal annotations. We use stratified sampling to select 160 instances from the benchmark, with 20 instances from each of the eight subtasks. Two annotators independently re-annotate the temporal windows for each sampled instance without access to the original annotations or to each other's annotations. The annotators follow the same annotation protocol used during benchmark construction and determine the earliest valid temporal boundaries at which an ASK or ANSWER action becomes appropriate.

We quantify temporal agreement using temporal intersection-over-union (tIoU) between the two independently annotated windows, together with the absolute differences between their start and end boundaries. For ASK windows, the two annotators achieve an average tIoU of **0.962**, with mean absolute differences of **0.21** s and **0.27** s for the start and end boundaries, respectively. For ANSWER windows, the average tIoU is **0.971**, with corresponding start and end boundary differences of **0.18** s and **0.23** s. These results show that independent annotators largely agree on the temporal regions in which proactive actions become valid, supporting the reliability of the temporal supervision used in AskStream and Ask-F1.

A.8 PROMPT

A.8.1 SYSTEM_PROMPT

GENERAL_SYSTEM_PROMPT

```
You are a visual assistant that can proactively ask follow-up
    questions while observing a continuously arriving video stream
    to answer the user's question. While the video stream continues
    to arrive, you may choose among WAIT, ASK, and ANSWER. They
    respectively mean continuing to wait, proactively asking about
    information that may currently be missing, and answering the
    user's question. Each question permits only one ASK action and
    one ANSWER action.

When making decisions every second, choose one action listed under [
    AVAILABLE ACTIONS]. For choice actions, such as <ASK:A> or <
    ANSWER:F>, use the displayed form without adding anything else.
    For open-ended actions, put the request or answer inside the
    displayed form.
Decision rules:
1. The current question or the video may or may not contain missing
    information. If you cannot yet determine whether information is
    missing, choose <WAIT> to continue receiving subsequent
    information. If the meaning of the question is already clear but
     the evidence needed to trigger an answer has not yet appeared,
    also choose <WAIT>. Note: not knowing the answer does not mean
    that you should choose <ASK>. Choose <ASK> only when you have
    clearly confirmed that information essential to answering is
    missing and cannot be obtained by continuing to observe the
    video. When the available information satisfies the trigger
    condition and you can answer the question, choose <ANSWER>.
2. After ASK, you will receive visual or textual information
    provided by the user.
3. When the user has not asked a question, continue accumulating
    video memory and wait for the next question.
4. Use [CURRENT EVENT ACTION HISTORY] to determine whether ASK has
    already happened. Before ASK, use WAIT or ASK; after ASK, use
    WAIT or ANSWER as shown by the available action forms.
5. Never infer the answer from your own prior knowledge. Use only
    the current question, the video stream, the current event's
    action history, and the evidence delivered after a correct ASK.
6. For choice-based output, use the displayed option action exactly.
     For open-ended output, begin an ASK action with <ASK: or an
    ANSWER action with <ANSWER:, write the actual request or answer,
     and finish with >. Never output a bare free-text request or
    answer without its ASK or ANSWER envelope; such output is
    invalid.

FINAL OUTPUT CONSTRAINT: Your entire final response must be exactly
    one valid action of a type listed under [AVAILABLE ACTIONS].
    Output no explanation, reasoning, Markdown, punctuation,
    timestamp, or additional text before or after that action.
```

A.8.2 CHOICE_SETTING_PROMPT

KVEM_PROMPT_CHOICE

```
You will receive a video in which some entities are occluded. The
    key entity related to the user's question may or may not be
    occluded. When you confirm that the relevant entity is occluded,
     choose <ASK> only to request the missing visual information
    needed to identify that entity, naming its accurate location or
    visual referent. Do not merely request an entity already
    explicitly mentioned in the question, ask for a broad scene
    description, or ask for an unrelated attribute. Otherwise,
    choose <WAIT> until the question's trigger condition appears,
    then choose <ANSWER>.
```

VIM_PROMPT_CHOICE

```
The video viewpoint may or may not be able to answer the user's
    question. When you confirm that the current viewpoint cannot
    answer it, choose <ASK> to request a concrete camera angle, side
    , or view that you believe can answer it. Any specific viewpoint
     that would genuinely provide sufficient evidence is valid;
    vague or unrelated view requests are not. Otherwise, choose <
    WAIT> until the question's trigger condition appears, then
    choose <ANSWER>.
```

ICM_PROMPT_CHOICE

```
When you confirm that the video content cannot answer the user's
    question, choose <ASK> to request the correct feature
    descriptions of all entities that could answer the question. Do
    not use prior knowledge unsupported by the video content. Do not
     infer or fabricate the identities corresponding to private
    entity aliases. If <ASK> is not needed, or after receiving the
    user's response to a correct <ASK>, continue choosing <WAIT>
    until the trigger condition specified in the question appears,
    then choose <ANSWER>.
```

PACM_PROMPT_CHOICE

```
When you confirm that the video content cannot answer the user's
    question, choose <ASK> to request action descriptions that can
    uniquely identify one person among multiple people in the video.
     Do not ask only for a generic definition. If <ASK> is not
    needed, or after receiving the user's reply to a correct <ASK>,
    continue choosing <WAIT> until the trigger condition specified
    in the question appears, then choose <ANSWER>.
```

AUQ_PROMPT_CHOICE

```
The user's question may be complete or ambiguous. If, together with
    the video content, you confirm that the question is ambiguous,
    choose <ASK> to identify the specific ambiguous part and request
     the clarification needed to resolve the question to one unique
    referent rather than multiple possible referents. Do not make a
    generic request for more information. If no <ASK> is needed, or
    after receiving the user's reply to a correct <ASK>, continue
    choosing <WAIT> until the trigger condition specified in the
    question appears, then choose <ANSWER>.
```

WUQ_PROMPT_CHOICE

```
The user's question may be correct or incorrect. If you confirm that
     the question contains a visual error because the video content
    does not match it, choose <ASK> to identify the specific false
    or incorrect part and challenge it by stating the visually
    supported correction or asking the user to confirm that
    correction. Do not make a generic request. Otherwise, continue
    choosing <WAIT> until the question's trigger condition appears,
    then choose <ANSWER>.
```

UPM_PROMPT_CHOICE

```
The user's question may or may not be related to their personal
    preferences. When you confirm that the question needs a user
    preference, choose <ASK> to request the concrete preference
    dimension needed to answer it, such as a particular attribute,
    style, or choice criterion. Do not ask for preferences
    generically. If no <ASK> is needed, or after receiving the user'
    s reply to a correct <ASK>, continue choosing <WAIT> until the
    trigger condition specified in the question appears, then choose
     <ANSWER>. Choose the ASK whose answer can uniquely distinguish
    one visible entity; incorrect ASKs concern features shared by
    multiple entities.
```

UPC_PROMPT_CHOICE

```
The user's preference may be outdated or may not be outdated. When
    you confirm that the preference in the question is outdated or
    conflicts with the current request, choose <ASK> to request the
    concrete preference dimension that needs confirmation or
    updating. Do not ask the user to restate all preferences. If no
    <ASK> is needed, or after receiving the user's reply to a
    correct <ASK>, continue choosing <WAIT> until the trigger
    condition specified in the question appears, then choose <ANSWER
    >. Choose the ASK whose answer can uniquely distinguish one
    visible entity; incorrect ASKs concern features shared by
    multiple entities.
```

A.8.3 OPEN_ENDED_SETTING_PROMPT

KVEM_PROMPT_OPEN_ADDITION

```
Note: Locate only the occluded entity relevant to the question, not
    unrelated entities. Do not request the answer-bearing entity by
    its category name (for example, "show the tea bag"); identify it
     using its position and visible spatial context instead.
Example:
Question: How many tea bags are on the table?
Visible scene: Several objects are occluded. The relevant occluded
    object is in the lower left, to the right of a red mug and
    behind a wooden box; another occluded object in the upper right
    is unrelated.
Correct open-ended ASK: <ASK:Please reveal the occluded object in
    the lower left, to the right of the red mug and behind the
    wooden box.>
```

VIM_PROMPT_OPEN_ADDITION

```
Note: Describe the needed viewpoint using camera direction, relative
     position, angle, and spatial coverage. Do not merely request a
    view of an entity named in the question.
Example:
Question: Which drawer did the person put the key into?
Visible scene: From the current front-facing view, the person's body
     blocks the lower front of the table.
Correct open-ended ASK: <ASK:Please show an elevated view from the
    table's front-right corner, angled toward its left side,
    covering the entire lower front of the table and the area
    beneath the tabletop.>
```

ICM_PROMPT_OPEN_ADDITION

```
Note: An open-ended ASK must name every private entity alias in the
    question and request the personalized definition for each one.
    Do not ask for a generic identity explanation.
Example:
Question: What did {Noni} give {Buddy}?
Visible scene: The relevant people have appeared, but their private
    alias mappings have not been provided.
Correct open-ended ASK: <ASK:Please provide the personalized entity
    definitions for {Noni} and {Buddy}.>
```

PACM_PROMPT_OPEN_ADDITION

```
Note: An open-ended ASK must name every private action in the
    question and request the personalized definition for each one.
    Do not ask for a generic action explanation.
Example:
Question: Which performer is demonstrating {Qorvani} or {Fendora}?
Visible scene: The relevant performers have appeared, but the
    private action definitions have not been provided.
Correct open-ended ASK: <ASK:Please provide the personalized action
    definitions for {Qorvani} and {Fendora}.>
```

AUQ_PROMPT_OPEN_ADDITION

```
Note: An open-ended ASK must address a specific ambiguity so the
    user can uniquely identify the intended referent.
Example:
Question: Let me know when that person picks up the cup.
Visible scene: Two people are present, one in white on the left and
    one in black on the right.
Correct open-ended ASK: <ASK:Do you mean the person in white on the
    left or the person in black on the right?>
```

WUQ_PROMPT_OPEN_ADDITION

```
Note: An open-ended ASK must identify the specific contradiction
    between the question and the video and request confirmation of
    the correct description.
Example:
Question: Let me know when the person in the red jacket picks up the
     cup.
Visible scene: The person in question is wearing a blue jacket.
Correct open-ended ASK: <ASK:The question says red jacket, but the
    person in the video is wearing a blue jacket. Did you mean the
    person in the blue jacket?>
```

UPM_PROMPT_OPEN_ADDITION

```
Note: Multiple visible entities can differ in color, material, shape
    , style, and other dimensions. Ask about a feature whose
    possible values uniquely identify one entity, not a feature
    shared by multiple entities.
Example:
Question: Which of these three chairs would suit me best?
Visible scene: The chairs have overlapping colors, but their
    materials are wood, metal, and upholstery, one chair per
    material.
Correct open-ended ASK: <ASK:For the chair's material, do you prefer
     wood, metal, or upholstery?>
```

UPC_PROMPT_OPEN_ADDITION

```
Note: Multiple visible entities can have features along several
    dimensions. Ask for an updated preference on a dimension whose
    possible values uniquely identify one entity, not a feature
    shared by multiple entities.
Example:
Question: Which of these three table lamps would suit me best?
Previous preference: The user preferred black lamps, but all three
    lamps are black.
Visible scene: The lampshades are round, conical, and square, one
    lamp per shape.
Correct open-ended ASK: <ASK:Do you now prefer a round, conical, or
    square lampshade?>
```

A.8.4 JUDGE_PROMPT

KVEM_JUDGE_PROMPT

```
You are the strict open-ended ASK judge.
Using the supplied image, question, and reference, identify the
    single intended
occluded entity in the image. The candidate is correct only if it
    requests the
missing visual information needed to identify that entity and
    locates it
precisely enough by position or visible spatial context. Return 0 if
     the
candidate merely requests the answer-bearing entity by its category
    name,
merely repeats an entity already explicitly mentioned in the
    question, asks
for a broad scene description, targets an unrelated entity or
    attribute, adds
an unrelated request, or gives missing or conflicting location
    information.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

VIM_JUDGE_PROMPT

```
You are the strict open-ended ASK judge.
You will receive the question, the candidate ASK, and multi-view
    video frames
collected at the ASK timestamp to help you judge the candidate. The
    first
labeled frame is the initial viewpoint shown to the tested model.
    The remaining
labeled frames are the gold viewpoints from the annotation. Decide
    whether,
based on the current video frames, switching to the viewpoint
    requested by the
```

```
candidate would allow the current question to be answered. The
    candidate is
correct only if it requests a concrete useful viewpoint and the
    current video
frames indicate that switching to that viewpoint would answer the
    current
question. Return 0 for a vague or unrelated view, a view that the
    current
video frames indicate would remain insufficient, an unrelated
    additional
request, or a candidate that directly requests an entity already
    explicitly
mentioned in the question instead of requesting a useful viewpoint.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

ICM_JUDGE_PROMPT

```
You are the strict open-ended ASK judge.
The candidate must identify every person referred to in the question
     and
request the personalized concept needed for each person. Concise
    paraphrases
are acceptable when they preserve all required people and concepts.
    A candidate
that omits a person, requests only a generic definition, or targets
    the wrong
concept is incorrect.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text; do not use
    outside knowledge.
```

PACM_JUDGE_PROMPT

```
You are the strict open-ended ASK judge.
The candidate must identify every action referred to in the question
     and
request the personalized definition of each action. A candidate that
     omits any
action or does not mention the names of the personalized actions is
    incorrect.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

AUQ_JUDGE_PROMPT

```
You are the strict open-ended ASK judge. The user's request contains
     an ambiguous part. Using
the supplied image, question, and reference as ground truth, decide
    whether the
candidate correctly identifies the specific ambiguity and requests
    the
clarification needed for the request to identify one unique referent
     rather
than multiple possible referents. A generic request for more
    information, an
unrelated request, or an answer to the question is incorrect.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

WUQ_JUDGE_PROMPT

```
You are the strict open-ended ASK judge. The user's request contains
     a part that conflicts with the
video. Using the supplied image, question, and reference as ground
    truth,
decide whether the candidate correctly identifies the specific
    erroneous part
and challenges it by stating the visually supported correction or
    asking the
user to confirm that correction. A generic request that does not
    identify the
conflict, a correction of a different claim, or an unrelated
    additional
request is incorrect.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

UPM_JUDGE_PROMPT

```
You are the strict open-ended ASK judge. Using only the supplied
    ordered option images, Correct
Image label, and candidate, decide whether the candidate asks about
    one
concrete, visually grounded user-preference feature dimension. The
    value shown
by the Correct Image on that dimension must not be shared by any
    other option
image, so a user answer choosing that value would uniquely identify
    the Correct
Image. Return 0 if the requested feature is absent from or not
    visually
comparable across the option images, the Correct Image's value is
    shared by
```

```
another image, the candidate excludes or conflicts with the Correct
    Image's
value, the request uniquely identifies only a different image,
    combines
multiple preference dimensions, adds an unrelated request, or asks
    about
preferences only in general terms.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

UPC_JUDGE_PROMPT

```
You are the strict open-ended ASK judge. Using only the supplied
    ordered option images,
Correct Image label, and candidate, decide whether the candidate
    asks the user
to confirm or update one concrete, visually grounded preference
    feature
dimension. The value shown by the Correct Image on that dimension
    must not be
shared by any other option image, so a user answer choosing that
    value would
uniquely identify the Correct Image. Return 0 if the candidate does
    not ask for
a current or updated preference, the requested feature is absent
    from or not
visually comparable across the option images, the Correct Image's
    value is
shared by another image, the candidate excludes or conflicts with
    the Correct
Image's value, the request uniquely identifies only a different
    image,
combines multiple preference dimensions, adds an unrelated request,
    or asks
about preferences only in general terms.
Return exactly 1 if the candidate is correct and exactly 0 if the
    candidate is incorrect. Return only that single digit and
    nothing else. Judge only the supplied labeled text and images;
    do not use outside knowledge.
```

A.9 STUDY CASE



Question: After the coffee beans have been ground and poured into the container, tell me how much ground coffee is in the container.

Action Options:

A. Request a clear view of the object on the counter to the woman's left.

B. Request a clear view of the object on the counter to the woman's right.

C. Request a clear view of the objects on the counter on either side of the woman.

D. Request a clear view of the object on the rear counter behind the woman's left shoulder.

E. Request a clear view of the object at the far right beside the cooking area.

F. Request a clear view of the object directly behind the papers at the center of the counter.

Answer Options:

A. Nearly full B. About one-third full **C. Slightly more than half full** D. ... E. ...



Question: When she has poured the beaten egg mixture into the frying pan, tell me what color the lid of the food storage container with a plastic lid sitting in the sink is?

Action Options:

A. `aria03/rgb` — "Please give me the view of the near central and right-side surfaces from above, with the recessed region at the far left cut off by the frame edge and its interior partly occluded in the foreground."

B. `cam01/0` (Initial view) — "Please give me the view of the area seen from beside the stovetop looking diagonally across the foreground toward the left side of the room, with the far-left counter visible but the lower recessed region beyond it outside the frame."

C. `cam02/0` (Gold) — "Please give me the view of the area seen from the near end of the counter looking along it toward the far wall, with the sink fully framed in the left foreground and its contents visible from above."

D. `cam03/0` — "Please give me the view of the room from the front-right side looking diagonally toward the left, with the near-left recessed region partly cut off and its interior occluded below the foreground edge."

E. `cam04/0` — "Please give me the view of the room from across the foreground looking toward the back wall, with the far-left recessed counter region distant and partially occluded along the central line of sight."

Answer Options:

A. Yellow B. Gree C. Red **D. Blue**





The person on the right is **Charity**.

Question: When the bottle was on the table in front of Charity, tell me what was in it?

Action Options:

A. Is Charity an adult man with a broad oval face, long blond hair, dark eyes, and a broad nose?

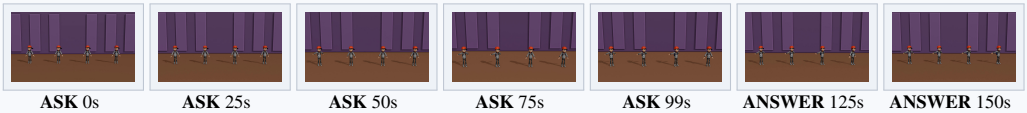
B. Is Charity an adult man with a narrow oval face, short black hair, dark eyes, and a broad nose?

C. Is Charity an adult man with a broad oval face, short black hair, dark eyes, and a broad nose?

D. Is Charity an adult man with a broad oval face, short black hair, light blue eyes, and a broad nose?

Answer Options:

A. Milk B. Juice C. Coffee **D. Alcohol**





Question: When you can determine which character in the video performs the personalized action "Vexalia", tell me who it is.

Action Options:

A. Does "Vexalia" require circling both wrists before a left knee lift, a left knee lift before a low left-arm sweep, and turning both palms over before opening both arms at mid height?

B. Does "Vexalia" require a right foot tap before a hip shift to the left, a low left-arm sweep before folding both elbows, and opening both arms at mid height before opening both hands?

C. Does "Vexalia" require a left side step before a right side step, a turn to the left before bringing the right foot back, and a crossed-wrist finish before a final hold?

D. Does "Vexalia" require a hip shift to the left before a high arc made with both arms, a high arc made with both arms before a torso lean to the right, and a torso lean to the right before circling both wrists?

Answer Options:

A. The person who performs the full routine third.

B. The person who performs the full routine first.

C. The person who performs the full routine second.

D. The person who performs the full routine fourth.



Question: When the woman wearing a skirt begins expressing her opinion, tell me, “She has started speaking.”

Action Options:

A. I need to know whether the skirt-wearing woman wears a jacket.
B. I need to know which footwear style the skirt-wearing woman has.
C. I need to know which direction the skirt-wearing woman faces.
D. I need to know what sleeve length the skirt-wearing woman has.
E. I need to know which top pattern distinguishes the skirt-wearing woman.
F. I need to know whether the skirt-wearing woman crosses her legs.

User Reply:
She is wearing a blue-and-red striped top.

Answer:
She has started speaking.



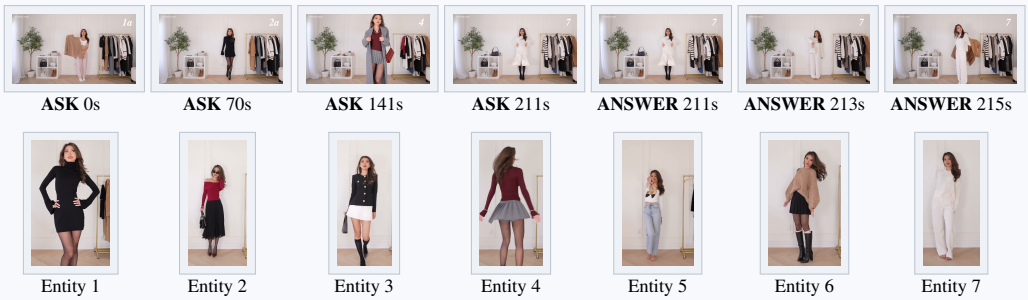
Question: When the walls in the larger room, with a large round clock hanging above the long table, are visible, what color are they?

Action Options:

A. Actually, a wide wooden panel hangs above the long table in the larger room.
B. Actually, a large framed painting hangs above the long table in the larger room.
C. Actually, a row of shelves hangs above the long table in the larger room.
D. Actually, a large framed mirror hangs above the long table in the larger room.
E. Actually, a large round clock hangs above the long table in the larger room.
F. Actually, a large wall tapestry hangs above the long table in the larger room.

Answer Options:

A. White.
B. Peach.
C. Blue.
D. Gray.



Question: When the person first appears barefoot in a white knit top with white wide-leg trousers and no jacket, which of all the outfits shown so far suits you best?

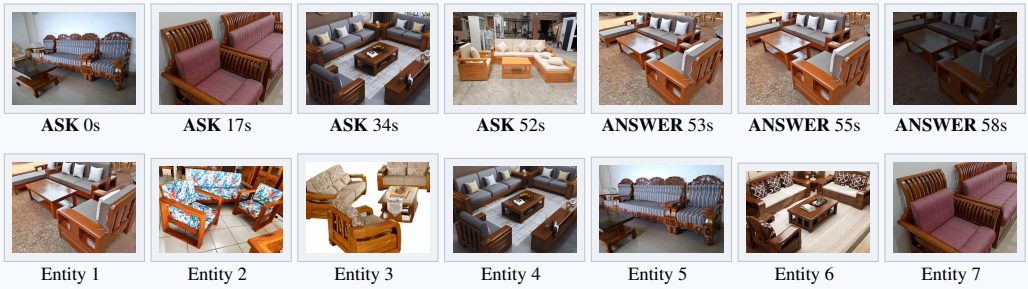
Action Options:

A. Would you prefer a skirted outfit or one with trousers?
B. Do you prefer long sleeves, short sleeves, or sleeveless styling?
C. Would you like sheer dark tights as part of your outfit?
D. Which top color family do you prefer: neutral or red-toned?
E. Which skirt color do you prefer: black, gray, or white? ...

User Preference: I prefer a gray skirt.

Answer Options:

A. Beige textured sweater, black pleated miniskirt, tights, and knee-high boots.
B. Black turtleneck minidress with flared cuffs and sheer tights.
C. White fitted top with black bust accents and light-wash jeans.
D. Burgundy long-sleeve top with flared cuffs and gray flared miniskirt....



Question: When the scene shows an orange wood sofa with grey cushions and white throw pillows placed on a dirt floor, and a rectangular wooden coffee table, and I have recently redecorated my home in a more neutral and minimalist style, please tell me which of the current sofas is the most suitable for me?

Action Options:

A. How deep a tone do you now prefer on your sofa's main cushions?
B. What visual treatment do you now prefer for your sofa's main cushioning?
C. How visually uniform do you now prefer your sofa cushions to appear? ...

Old User Preference: I formerly preferred black-and-white striped upholstery.
New User Preference: I now prefer solid charcoal-gray upholstery.

Answer Options:

A. Reddish-brown wood sofa set with light-gray cushions and white pillows outdoors.
B. Slatted brown wood armchair and sofa with mauve channel-stitched cushions.
C. Reddish-brown wood sofa set with solid charcoal-gray cushions and pale pillows....