BALANCING TIMELINESS AND FIRING PRECISION IN PROACTIVE STREAMING VIDEO AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding is becoming increasingly important for real-world interactive systems. Such systems must be *proactive* for natural and useful human interaction: they should respond as soon as sufficient evidence becomes available while avoiding premature, redundant, or incorrect responses. We present PACE, a framework unifying data, metrics, harness, and training around a shared formulation of proactivity as *effective causal response decisions* balancing the two competing objectives of responding vs silence. First, we construct PACE-780K dataset containing human annotations of proactive response content and timing, which defines this desired behavior. We then propose PAUC-F, a metric to effectively measure our desired proactive behavior. PAUC-F combines existing response timeliness metrics with firing precision to reward early useful responses *without* rewarding excessive firing. With our desired behavior defined and measurable, we finally investigate model-agnostic inference harnesses that draw out proactive streaming capabilities of offline vision language models (VLMs), together with training pipelines that learn to further improve their proactive responsiveness. Across both settings, we find that combinations of sliding visual windows, online interaction memory, and structured response outputs consistently yield strongest proactive performance. Our experiments result in two models, with our PACE-Qwen model achieving state-of-the-art performance across four proactive streaming benchmarks, with gains up to 44% over baselines and outperforming multiple closed API models. Our code and data will be released publicly.

1 INTRODUCTION

Proactive streaming video agents must learn not only what to say, but also *when to say it*, and *when to remain silent* (Wang et al., 2026; 2025b). This differs from offline video question-answering, where queries are immediately followed by responses. Streaming agents observe videos incrementally (causal), and at each frame decide whether to respond or remain silent for a query provided at an earlier time (Chen et al., 2024a; Yu et al., 2025). This causal firing decision presents a key challenge: a correct response is unhelpful if it arrives too late or is repeated too often (Wang et al., 2026). Effective proactivity requires balancing response timeliness with firing precision.

In order to achieve this desired behavior in streaming video agents, we propose aligning the end-to-end pipeline of data, metrics, harness, and training under a shared formulation. While existing works seperately address subsets of these, we present Proactive Agents with Calibrated Expression (PACE), a single framework aligning data, metrics, harness, and training around a shared formulation of proactive responding as effective causal response decisions (illustrated in Figure 1).

We first construct PACE-780K, a large-scale dataset of proactive responses built through automated proactive question-answer generation followed by comprehensive human verification. In contrast to existing proactive streaming datasets (Table 1), PACE-780K provides human verified response content and intervals, providing explicit human supervision on when to remain silent. Next, we propose PAUC-F, an evaluation metric that captures both timeliness and firing precision in proactive streaming model responses. Prior metrics such as PAUC (Wang et al., 2025b) and ESTP-F1 Yu et al. (2025) separately measure individual aspects of the proactive streaming task, leaving room for models to game these metrics with undesirable traits (see Figure 2 and Table 2). Our proposed PAUC-F, retains PAUC's quality-weighted timeliness and balances it with response precision.

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.

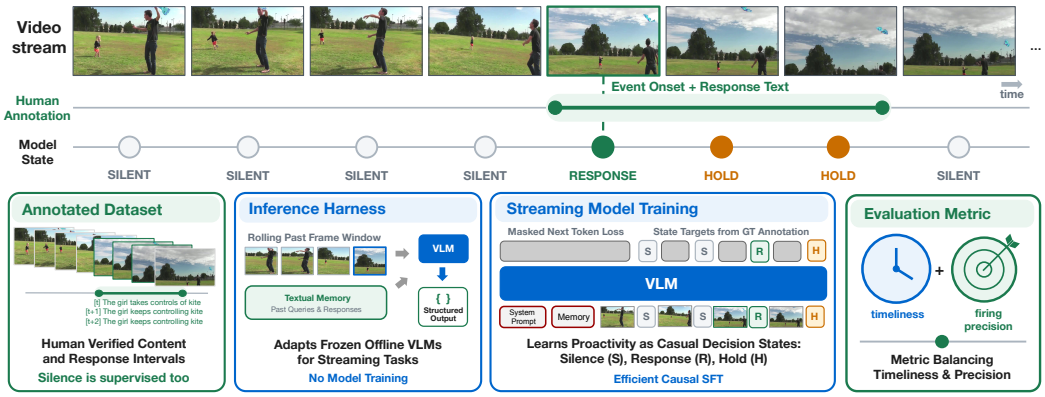

Figure 1: **Overview of proposed PACE framework.** We unify data, metrics, harness, and training around a shared formulation of proactivity as *effective causal response decisions* balancing the two competing objectives of responding vs silence in the proactive streaming task.

With desired behavior defined and measurable, next we investigate harness designs and training pipelines that improve proactive behavior in VLMs. Existing works explore proactivity with learned activation models (Wang et al., 2025a), hand-crafted harness designs (Yao et al., 2026), learned response decision tokens (Chen et al., 2024a; Li et al., 2026), and response policy optimization (Wang et al., 2025c; 2026). We introduce a model-agnostic harness tuning strategy to elicit proactivity in VLMs. Our selected harness combines a sliding visual window, interaction memory, and structured decision states. The same interface then supports efficient teacher-forced training with our datasets, further improving proactive responsiveness in VLMs, and resulting in two models, PACE-Gemma and PACE-Qwen. Our PACE-Qwen model achieves state-of-the-art performance across four proactive streaming benchmarks, outperforming multiple closed source models as well.

We summarize our key contributions as follows:

1. **PACE-780K Dataset:** A high-quality proactive streaming dataset for training built through automated generation of proactive question–response pairs, followed by comprehensive human review and filtering to ensure accurate response content and timing annotations.
2. **PAUC-F Metric:** Evaluation metric designed to measure proactive understanding in VLMs that captures both response timeliness *and* firing precision.
3. **Inference Harness:** Model-agnostic harness tuning and calibration that adapts offline VLMs for proactive responding under streaming style inference.
4. **Training Framework:** Efficient training pipeline resulting in PACE-Gemma and PACE-Qwen models that yield relative improvements up to 44% over prior baselines across benchmarks.

2 Related Work

Proactive Video Understanding. Proactive responsiveness is tackled under two key directions: inference harness and training pipeline. Harness design involves response gates, event detection, and long-context memory management (Wang et al., 2025a; Qian et al., 2025; Ding et al., 2025; Azad et al., 2026; Zhang et al., 2026a; Tian et al., 2026; Yao et al., 2025; Zheng et al., 2026; Yao et al., 2026), but its roles is mainly streaming setting adaptation, overlooking direct optimization for proactivity. Proactive behavior is handled mostly in model through response decision states (Chen et al., 2024a; Xia et al., 2026; Li et al., 2026), implicit signals in interleaved video-text data (Wang et al., 2025c; Yang et al., 2025), and reinforcement learning (Wang et al., 2026). In contrast, our proposed harness is model agnostic and tuned specifically for proactivity, while the same interface extends to supervision when training data is available and transfers across model families rather than committing to fixed architectures.

Proactive Streaming Datasets. Prior works convert offline annotations from existing datasets to streaming style, often with LLM assistance. Chen et al. (2024a) reformats COIN and Ego4D narrations directly into streaming dialogues, while Wang et al. (2025c;a) use dense captioning task data to determine response timing using annotated temporal segments. Wang et al. (2026); Yao et al. (2025) generate dialogue from scene captions with text-only LLMs. Xia et al. (2026); Li et al. (2026) de-

Table 1: **Proposed Streaming Proactive Dataset:** We construct a training dataset for proactive streaming task with human-annotated content and intervals.

Dataset	Videos	Questions	Responses	Human-verified	
				Content	Intervals
MMDuet2 (Wang et al., 2026)	52,771	105,800	350,800	No	No
VideoChat3-OL (Li et al., 2026)	191,000	617,183	617,183	No	No
ESTP-IT (Yu et al., 2025)	1,166	80,000	80,000	No	No
PARSE-Ego4D (Abreu et al., 2025)	5,515	18,360	18,360	Yes	No
Ours	**177,789**	**177,789**	**784,944**	**Yes**	**Yes**

rive labels from a VLM at scale. However, existing datasets suffer from poor response timing (Wang et al., 2026), affecting quality of proactive responsiveness. In contrast, PACE-780K ensures accurate response content and timing through comprehensive human verification (see Table 1).

Evaluation Metrics for Proactive Understanding. Early benchmarks evaluate proactivity through targeted tasks. OVO-Bench (Li et al., 2025) introduces CRR (Clue Reveal Responding) task where the same query requires different responses at different timesteps. StreamingBench proposes Proactive Output (PO) task that evaluates the first response occurring within ± 2 seconds of an annotated timestamp (Lin et al., 2024). In contrast, later ESTP-F1 (Yu et al., 2025) and PAUC (Wang et al., 2025b) metrics evaluate continuous interaction under true streaming settings, but both overlook firing-precision to certain extents (Figure 2), motivating our proposed PAUC-F metric.

Automated Harness Design. Evolving inference harnesses is explored for general agents and offline video understanding (Zhang et al., 2026c; Xu & Chen, 2026; Zhang et al., 2026b), but ours is a first to demonstrate successful integration to proactive or streaming video tasks.

3 THE PACE-780K DATASET

We construct a dataset for learning proactive response behavior in streaming video tasks, containing 784,944 proactive responses across 177,789 videos. Each video includes a user query followed by a sequence of temporally grounded natural-language responses. Our dataset uses the same source videos as LLaVA-Video-178K (Zhang et al., 2024). We employ a three-stage annotation pipeline:

Stage 1: Automated Generation. Creates candidate proactive query and response sequences.
Stage 2: Human Verification and Correction. Human annotators inspect generated samples and correct errors related to both response timing and content.
Stage 3: Expert Quality Review. Independent PhD-level reviewers evaluate corrected annotations in batches. Batches below acceptance threshold are returned to Stage 2 for further revision.

Note that each query can require a single or multiple responses across the video, with each response corresponding to a single frame or frame interval. Our annotations capture both response and silent timings to encourage firing precision. Table 1 shows how compared to earlier work, our dataset is larger in scale and provides human-verified annotations for both response content and intervals.

Stage 1: Automated Proactive Query-Response Generation. We generate proactive interactions for a given video using a VLM in two successive stages. First, for query generation, we provide the VLM with a single video and obtain a single video-specific user query relevant to measuring proactive responsiveness. Next, for response generation, we provide the VLM both the video and query to obtain an ordered sequence of response or silence states. This is generated at 1 FPS, i.e. one state per second. Response states also include the response content. We maintain causal behavior in generated responses (resembling online streaming setting) by restricting the VLM to process the video in a streaming fashion with no access to future frames.

Stage 2: Human Verification and Correction. Human annotators first filter videos based on query validity, and then evaluate each response or silent state. Incorrect states are fixed, including correction of response content when necessary. Query validity is defined as requiring only visually verifiable conditions and unambiguous answers. Response and silent states are evaluated based on alignment with evidence present in video until the relevant timestep, and corrected accordingly. Note

that response states capture not only instantaneous events but also continuous events which involve consecutive response states over multiple timesteps.

Stage 3: Quality Control. All annotators are recruited and trained by a professional video data annotation team. We provide detailed instructions and visual examples describing our annotation requirement. The annotated samples undergo quality control at batch level, using PhD-level expert reviewers (who designed the annotation process). After each batch, annotation proceeds to next only if quality exceeds our required threshold. Failing batches are returned for further review and correction. While initial batches required one repeat cycle, later batches passed required quality directly, ascertaining that annotation process was well aligned with our specified requirement.

4 PAUC-F: Proactive Streaming Evaluation Metric

We propose PAUC-F that jointly captures answer quality, timeliness, and firing precision. PAUC-F addresses limitations of earlier metrics and shows over 90% better alignment with human preference.

4.1 Limitations of Prior Metrics

PAUC (Wang et al., 2025b) and ESTP-F1 (Yu et al., 2025) are two key metrics in recent literature relevant to proactivity in streaming video agents. PAUC measures quality weighted timeliness in responses. However, PAUC only measures recall within the gold-window. This results in responses outside of the gold-window window being discarded, and repeated responses within the window not affecting the score (see Figure 2). Similar observations are noted in MMDuet2 (Wang et al., 2026), where removing the repetition reward increases PAUC scores although duplicate replies increase by 4x. ESTP-F1 addresses out-of-window firing with the addition of a precision term, but ignores redundant in-window firing and requires task-dependent timing term for recall calculation. These two key metrics provide complementary but incomplete views, which we demonstrate in Figure 2.

4.2 PAUC-F Definition

We propose PAUC-F, which maintains PAUC as its recall and introduces a precision term:

$$R = \text{PAUC}, \qquad P = \frac{N_{\text{value}}}{N_{\text{fires}}}, \qquad \text{PAUC-F} = \frac{2PR}{P+R}, \tag{1}$$

where N_{fires} counts every emitted response, including responses outside gold windows, and N_{value} counts value-adding responses. For each gold turn, at most one response contributes to N_{value}, the earliest in-window fire with a positive judge score. All additional responses remain in N_{fires} but cannot increase N_{value}. It inherits ω from PAUC allowing quality-timeliness weight adjustment.

This formulation balances the two desired behaviors directly. PAUC-F reflects the PAUC recall score for models with similar firing precision. For models with unnecessary firing, the precision term lowers PAUC-F compared to PAUC. Thus, PAUC-F simply wraps the quality-weighted timeliness value introduced in PAUC with a *firing restraint aware* precision term (details in Appendix D).

4.3 Metric Properties

We investigate properties of our proposed PAUC-F metric in Figure 2 and Table 2. First, we construct a toy experiment simulating undesired excessive firing behavior. We use ground-truth response content and intervals to mimic models exhibiting within golden window repetitions and excessive out-of-window firing. As observed in Figure 2, both PAUC and ESTP-F1 fail to capture at least one of these undesired repetitive response behaviors, while PAUC-F correctly penalizes both. This excessive firing simulation was motivated by findings in prior work where models optimized for PAUC alone exhibit such repetitive behavior in their responses (Wang et al., 2026).

We also observe similar findings in our experiments, which we summarize in Table 2. We optimize two inference harness variants (details in Section 5.1) to independently maximize PAUC (`v1`) and PAUC-F (`v2`) on a held-out validation set of our PACE-780K dataset. PAUC-F optimized `v2` leads to clearly more desired behavior on downstream tasks compared to the PAUC optimized `v1` agent. We also conduct a human preference study involving 10 participants rating 20 samples from WEB

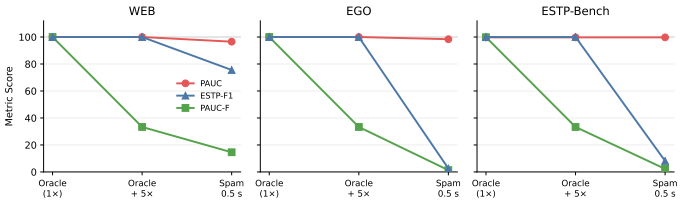

Figure 2: **PAUC-F vs Prior Metrics:** We simulate excessive firing behaviour (undesired) across WEB & EGO splits of ProactiveVideoQA (Wang et al., 2025b) and Explicit Proactive subset of ESTP-Bench (Yu et al., 2025) benchmarks, reporting our proposed PAUC-F metric alongside PAUC & ESTP-F1 metrics. Along the x-axes, we repeat ground-truth responses (Oracle) 5 additional times within golden windows (Oracle + 5x) and every 0.5s throughout the video (Spam 0.5). PAUC-F correctly identified these undesired response patterns while PAUC and ESTP-F1 incorrectly rewards all or some of these undesired test cases.

Table 2: **PAUC-F vs PAUC for model selection.** We build two harness variants optimizing PAUC (v1) vs PAUC-F (v2). PAUC-F optimized v2 shows better firing precision, closer firing counts as ground-truth (Fires:Gold), less consecutive repeated responses, and higher human preference.

	WEB		EGO	
Metric	v1	v2	v1	v2
PAUC ($\omega = 0.5$) ↑	64.82	57.30	66.57	37.57
PAUC-F ($\omega = 0.5$) ↑	31.92	43.39	3.58	35.94
Firing Precision ↑	21.17	34.91	1.84	34.45
Fires : Gold ↓	4.50×	2.29×	48.18×	0.81×
Repeat Responses ↓	14.30	2.20	185.90	1.32
Human Preference ↑	6%	94%	3%	97%

and EGO splits of (Wang et al., 2025b). Each sample contains response predictions from `v1` and `v2` agents that are rated for better proactive response quality. Human raters show clear preference for the PAUC-F optimized `v2` agent (more details in Appendix D).

5 METHOD

In this section, we introduce our inference harness and training pipeline for streaming video tasks.

Streaming Interaction Protocol. We first establish our definition of streaming inference, which is aligned with prior work (Chen et al., 2024a; Wang et al., 2026; Li et al., 2026). At time t, a model receives a window of past video frames and a textual state capturing prior queries and responses. In turn, the model must produce either silence or a response at the current timestamp. The model cannot access future frames, revise an earlier response, or assign a new answer to an older timestamp. This protocol aligns with the same prior work and allows models to understand streaming videos extending beyond model visual context and to proactively respond at any time after a user query.

Streaming Video Model. We also define a *streaming video model* as one trained specifically to consume inputs and generate outputs in this format, as opposed to offline VLMs trained for offline video question-answering that can be adapted to this task through suitable harness design.

5.1 INFERENCE HARNESS

Prior works often measure proactive behaviour of offline VLMs using non-streaming inference settings (Wang et al., 2025b; Chen et al., 2024a; Yu et al., 2025), making direct comparison to streaming models difficult. For example, Wang et al. (2025b) evaluates offline VLMs on 5s frame chunks with responses of frame t conditioned on frames up to $t + 5$, deviating from the streaming task setting. More recent works integrate offline VLMs with an inference harness (Li et al., 2026; Wang et al., 2025a; Yao et al., 2026) for evaluation under true streaming settings. While these works hand-craft harness designs, we focus on tuning a harness, motivated by recent work on automated harness evolution (Zhang et al., 2026c), to elicit stronger proactive responsiveness from an offline model.

Design Overview. As illustrated in Figure 3, our harness contains four key components: system prompt defining task and output schema, past frame window, memory holding prior interactions, and structured output parsing. The underlying model is called at the desired frame rate, with dynamic inputs at each timestep conditioned on the past model responses.

Harness Tuning. We next optimize the design of its four key components. Instead of hand-crafting these design choices, we utilize agent-assisted search to modify the harness, keeping underlying VLM model weights frozen (i.e. no model training). The agent first proposes hypotheses for modifying one of these four components, implements relevant code changes, and evaluates on PACE-780K validation set using our PAUC-F metric. These results guide new hypothesis creation over several rounds and the final design selection. Each round contains ten candidate variants and we promote variants that improve our objective (PAUC-F on validation set). Combination of complementary

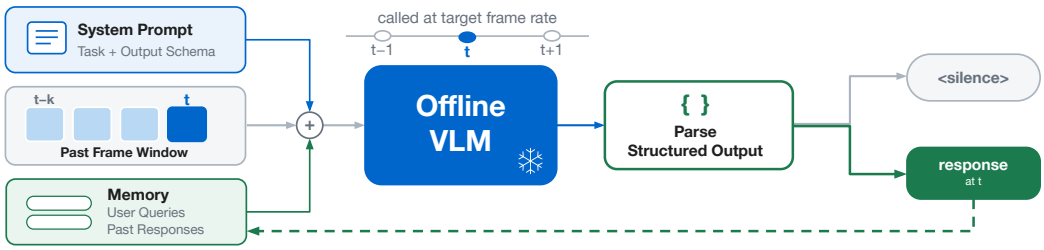

Figure 3: **Inference Harness.** At each streaming timestep t, the harness constructs an input from a system prompt (task and output schema), a rolling past-frame window, and memory of user queries and prior responses. A frozen offline VLM is queried at the target frame rate; structured output is parsed into either silence or a response at t, and accepted responses are updated into memory.

promoted changes is treated as a novel hypothesis for next round. We run 10 rounds of propose-evaluate-promote cycles with a total of 100 candidate evaluations each for two different models: `Gemma4-26B-A4B` (Gemma Team, 2026) and `Qwen3.5-35B-A3B` (Qwen Team, 2026). Both variants converge to a common design with minimal variations in system prompt and output schema.

Harness Calibration. Given our fixed harness, we perform further model-specific calibration of system prompt and output schema with a more modest compute budget, using the same validation set and PAUC-F metric. We evaluate five candidates on a fixed 20 sample subset, with each candidate modifying only the system prompt and output schema (joint changes allowed). This corresponds to 5 rounds (one candidate per round) and a total budget of evaluating 100 video samples. We perform this limited calibration on all other offline models prior to evaluation.

Interactivity: The user query is included in our memory which can be updated midway during a video. This design choice allows user interaction with the framework mid-video: a human user can pose new queries as the framework continues processing an existing video, further aligning with true streaming setting. We investigate and evaluate this behavior further in Table 6.

See Appendix E for further details of our harness design including all prompts.

5.2 Streaming Model Training

We next investigate how supervised finetuning (SFT) can further improve proactivity. We use streaming data derived from PACE-780K and construct a training pipeline that retains the efficiency of standard teacher-forced SFT. An overview of our training formulation is illustrated in Figure 4.

5.2.1 Streaming Episodes and Targets

Causal episode construction. We construct dense sequences of frame-level decisions from our annotated videos. First we build full video timelines followed by division into bounded chunks. A video timeline is a sequence of timestamped frame inputs and model turns, where model turns contain one of three state tokens, `</Silence>`, `</Response>`, or `</Hold>`. The response token is followed by the annotated natural-language response. These timelines are split into chunks that contain the memory, absolute timestamps, and its corresponding source-frame range. Within chunks, all response turns are fully supervised, while long runs of non-response states are subsampled to avoid silence dominating the token budget. A chunk corresponds to k frames, which is the past frame window that is provided to our model.

State targets. We ground every state in the human annotations. Frames with no relevant event are labeled `Silence`, an event onset that provides sufficient information is labeled `Response`, and subsequent annotated frames for the same ongoing event are labeled `Hold`. The hold target aims to avoid repetition of same response by model when no state change occurs. It captures the role for when relevant information is present, but another response is not yet warranted. Note that not all response states are followed by hold states; in our training corpus only 18% of response states are followed by a hold state. We also include post-response negative chunks that only contain silence states to allow learning sustained silence after a response. Further details on state targets and distribution in our training data in Appendix F.

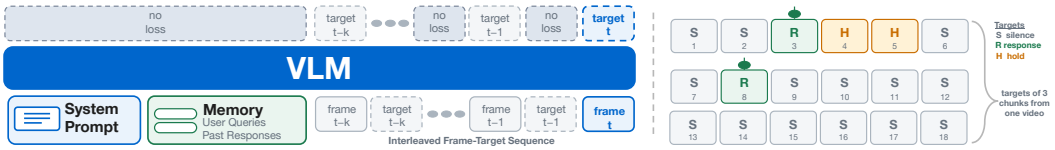

Figure 4: **Streaming Model Training.** Each training sequence concatenates a system prompt, memory of user queries / past responses, and an interleaved frame-target stream. Prompt, memory, and visual tokens receive no loss; the VLM is supervised on state targets (silence, response, and hold) and the accompanying response text. One video is split into multiple chunks so that long streams remain within the training context.

5.2.2 Streaming Supervision

We perform teacher-forcing SFT by using individual chunks as samples. Each chunk contains a system prompt, memory, and an interleaved sequence of frame-target pairs. The targets are the states `</Silence>`, `</Response>`, or `</Hold>`.

Memory Module. Each training chunk is paired with a memory component containing past user queries and last k annotated responses. This is a dynamic memory that changes with each timestep, since new queries and responses update its content. The ability to include new queries mid-video retains the interactive nature of our harness design. The past responses allow the model to keep record of responses outside the current frame window and avoid repetitions in long periods of silence. During inference, we fill the past response fields with the model's own generated responses.

Loss Masking. While our training employs a standard next token prediction loss, within each chunk we mask the cross-entropy loss term at token level. This is motivated by the large portion of silence timestamps inherent to the task and similar strategies in prior work (Li et al., 2026). Our token-level loss masking retains all `Response` targets while subsampling repetitive `Silence` and `Hold` continuations. Chunks containing only `Silent` responses also undergo masking. The train-time sequence also appends the natural language response after each response token, where the language tokens undergo no loss masking.

Overall, our inference harness and SFT pipeline share a single streaming interface, and demonstrate how offline VLMs can be adapted to streaming inference eliciting their proactive responsiveness.

6 Experiments

Experimental Setup. We train on the PACE-780K train split and select harnesses and checkpoints by validation PAUC-F; the four test benchmarks are held out. We report PAUC-F (GPT-4.1 judge) on ProactiveVideoQA and ESTP-Bench Explicit Proactive. Configurations are in Appendices E and F. Both training and inference use $8\times$ H200 GPU nodes.

6.1 Main Results

ProactiveVideoQA. We present PAUC-F evaluations across the four splits from Wang et al. (2025b) in Table 3. All offline proprietary and open-weight models are evaluated using our harness with per-model harness calibration. The Gemini-Live model is evaluated with its default harness as well as an integration of our harness (`ours-v`) without calibration. For streaming video model baselines, we evaluate the official model checkpoints with their default streaming inference settings. Our two trained models, PACE-Gemma and PACE-Qwen, use our harness for inference. Our trained PACE-Qwen model achieves state-of-the-art performance overall, demonstrating the usefulness of our PACE framework. PACE-Qwen (49.08) also outperforms prior best MMDuet2 (34.01) by 44%. We also report the raw PAUC numbers (official metric of benchmark) in Appendix A for reference.

ESTP-Bench. We next report PAUC-F on the *Explicit Proactive* subset of tasks from Yu et al. (2025), which measures proactive responsiveness. Our results in Table 4 indicate state-of-the-art performance of proposed PACE-Gemma and PACE-Qwen. All baselines use the same official model checkpoints with our harness, with streaming native models using their default inference setting.

Official ESTP-Bench, StreamingBench (PO), and OVOBench (CRR). We next report results on existing benchmarks using their default metrics, allowing direct comparison to baseline results

Table 3: **Results on ProactiveVideoQA Benchmark.** We report PAUC-F with varying ω using GPT-4.1 judge. Harness is ours for offline models and default from streaming native models. Our harness is modified to integrate with Gemini-Live (ours-v) due to closed API nature. **Best** and second-best marked in each column.

Model	Harness	[WEB]			[EGO]			[TV]			[VAD]			[Mean]		
ω =		0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
Proprietary Models																
GPT-4.1	ours	**57.92**	60.19	62.31	33.15	34.33	35.36	31.30	32.56	33.58	28.00	28.80	29.49	37.59	38.97	40.19
GPT-4.1-mini	ours	54.36	57.19	59.81	21.63	22.07	22.44	24.07	24.82	25.42	19.92	20.36	20.74	30.00	31.11	32.10
GPT-5.6-Sol	ours	57.62	**60.35**	**62.87**	31.01	32.03	32.92	38.75	40.94	42.74	33.59	34.81	35.90	40.24	42.03	43.61
Gemini-2.5-Pro	ours	55.13	57.67	60.01	32.70	33.97	35.06	36.14	37.98	39.48	23.84	24.53	25.11	36.95	38.54	39.92
Gemini-2.5-Flash	ours	40.02	41.31	42.45	9.27	9.33	9.38	17.92	18.33	18.65	4.31	4.33	4.35	17.88	18.33	18.71
Gemini-3.1-Pro	ours	50.21	52.76	55.00	19.83	20.13	20.38	37.62	39.49	41.02	0.66	0.66	0.66	27.08	28.26	29.27
Gemini-3.5-Flash-Lite	ours	48.78	51.20	53.33	11.14	11.23	11.31	20.44	20.96	21.38	0.44	0.44	0.44	20.20	20.96	21.62
Gemini-3.8-Flash	ours	51.22	53.84	56.15	19.12	19.44	19.71	30.56	31.78	32.76	0.33	0.33	0.33	25.31	26.35	27.24
Gemini-Live-2.5	default	27.56	27.56	27.56	0.00	0.00	0.00	15.48	15.48	15.48	0.00	0.00	0.00	10.76	10.76	10.76
Gemini-Live-2.5	ours-v	42.04	46.59	50.29	9.89	10.00	10.09	6.85	6.94	7.03	7.60	7.65	7.69	16.60	17.80	18.78
Open-Weight Models (No Streaming Video Training)																
Molmo2 8B	ours	36.17	38.22	40.07	10.70	10.81	10.90	9.79	9.88	9.95	6.61	6.64	6.66	15.82	16.39	16.90
MiniCPM-V 4.5	ours	37.16	39.43	41.40	11.88	12.04	12.18	4.38	4.40	4.41	4.38	4.39	4.40	14.45	15.07	15.60
Inkling-Small	ours	48.32	50.42	52.39	24.62	25.39	26.06	15.11	15.31	15.49	9.62	9.66	9.69	24.42	25.20	25.91
Gemma4-A4B	ours	41.92	43.39	44.71	34.63	35.94	37.15	26.17	26.60	27.01	27.82	28.50	29.13	32.64	33.61	34.50
Qwen3.5-A3B	ours	51.85	54.24	56.54	37.62	39.29	40.82	33.42	34.47	35.46	41.90	43.75	45.51	41.20	42.94	44.58
Streaming Video Models (Trained for Streaming Responses)																
VideoLLM-Online	default	32.19	32.20	32.20	6.64	6.64	6.64	2.45	2.45	2.46	0.00	0.00	0.00	10.32	10.32	10.33
EyeWO	default	38.97	40.42	41.81	4.76	4.77	4.78	3.24	3.23	3.23	0.00	0.00	0.00	11.74	12.11	12.46
MMDuet	default	48.43	50.76	52.92	7.48	7.52	7.55	2.70	2.71	2.72	2.41	2.42	2.42	15.26	15.85	16.40
MMDuet2	default	50.69	56.51	61.46	31.64	32.28	32.89	30.34	32.29	33.91	14.70	14.97	15.22	31.84	34.01	35.87
VideoChat3	default	33.68	34.70	35.68	28.86	29.71	30.51	36.90	39.53	41.89	22.84	23.26	23.66	30.57	31.80	32.94
PACE-Gemma (ours)	ours	52.19	56.46	60.38	41.81	43.80	45.61	31.65	33.27	34.70	30.85	31.98	33.01	39.13	41.38	43.43
PACE-Qwen (ours)	ours	53.13	57.48	61.41	**43.77**	**46.05**	**48.13**	**40.71**	**43.11**	**45.25**	**46.56**	**49.67**	**52.55**	**46.04**	**49.08**	**51.84**

Table 4: **Results on ESTP-Bench.** We report PAUC-F (ω = 0.5, GPT-4.1 judge) on Explicit Proactive subset of ESTP-Bench. Our harness used for offline models and default harness for streaming models. The task categories are Object Recognition (OR), Attribute Perception (AP), Text-Rich Understanding (TRU), Object Localization (OL), Object State Change Recognition (OSC), Ego Object Localization (EOL), Ego Object State Change Recognition (EOSC), and Action Recognition (AR). **Best** and second-best marked in each column.

Model	Harness	OR	AP	TRU	OL	OSC	EOL	EOSC	AR	All
Proprietary Models										
GPT-4.1	ours	20.12	26.88	23.57	23.60	17.89	19.71	19.51	24.01	21.91
GPT-4.1-mini	ours	21.80	27.41	27.04	**31.39**	19.48	26.51	18.17	21.81	24.20
Gemini-2.5-Pro	ours	**25.36**	27.14	**34.53**	30.14	21.01	19.36	18.94	22.89	24.92
Gemini-2.5-Flash	ours	9.44	15.67	19.02	9.40	9.22	8.04	7.62	12.98	11.42
Open-Weight Models (No Streaming Video Training)										
Molmo2 8B	ours	8.84	11.18	13.01	4.93	3.98	5.57	3.25	8.04	7.35
MiniCPM-V 4.5	ours	6.75	9.35	3.20	8.10	4.11	6.32	3.98	6.38	6.02
Inkling-Small	ours	10.28	14.11	10.45	13.27	9.08	10.91	8.12	12.25	11.06
Gemma4-A4B	ours	11.05	16.90	15.59	14.07	7.52	11.16	6.41	10.33	11.63
Qwen3.5-A3B	ours	23.88	27.17	23.07	29.78	29.24	23.10	29.87	23.47	26.20
Streaming Video Models (Trained for Streaming Responses)										
VideoLLM-Online	default	0.73	0.08	0.05	0.33	1.19	0.45	0.45	2.53	0.73
EyeWO	default	10.86	7.08	3.43	10.72	7.16	10.20	4.56	11.23	8.16
MMDuet	default	1.91	2.96	0.72	3.24	3.83	3.06	2.43	6.22	3.05
MMDuet2	default	15.47	20.89	14.15	14.74	11.32	11.50	10.43	10.17	13.58
VideoChat3	default	12.63	16.71	13.07	19.83	12.08	17.19	12.76	11.65	14.49
PACE-Gemma (ours)	ours	20.23	25.97	30.06	29.83	20.89	24.41	17.73	21.28	23.80
PACE-Qwen (ours)	ours	24.62	**31.77**	30.47	29.08	**30.82**	**27.84**	**31.64**	**24.24**	**28.81**

originally reported in Yu et al. (2025); Lin et al. (2024); Li et al. (2025). We report results for Gemma4 and Qwen3.5 offline models converted to online streaming setting with our harness, as well as PACE-Gemma and PACE-Qwen which are our two trained models.

In Table 5, we report ESTP-F1 metric on full ESTP-Bench benchmark. Baseline results are identical to Yu et al. (2025). Our PACE-Qwen model achieves best overall performance and on Explicit (Exp.) split. Implicit (Imp.) and Contextual (Ctx.) splits contain longer gold intervals and because ESTP-F1 overlooks in-window repetition, models with repetitive behavior can score higher on these (more in Table A.5).

In Table 6, we report time accuracy for the Proactive (PO) split of StreamingBench with baseline results obtained from official leaderboard. Our PACE-Qwen model similarly achieves overall best

Table 5: ESTP-F1 on ESTP-Bench with baseline results identical to Yu et al. (2025). Our methods (both harness and trained) are highlighted in gray. **Best** and second-best marked in each column.

Model	Exp.	Imp.	Ctx.	All
Offline MLLMs				
LLaVA-OV	13.6	31.8	8.2	18.0
Qwen2-VL	15.4	39.3	10.4	21.3
MiniCPM-V	15.2	36.5	26.6	22.9
LLaVA-NeXT	16.5	34.6	13.8	21.3
InternVL-V2	5.6	6.7	5.7	5.9
Online MLLMs				
$\text{LIVE}_{0.8}$	7.0	9.3	11.8	9.1
$\text{LIVE}_{0.9}$	9.5	25.6	18.9	15.5
MMDuet	9.1	34.2	20.3	17.8
EyeWO	23.6	**52.5**	**43.6**	34.7
Gemma4-A4B	18.2	31.2	27.1	23.2
Qwen3.5-A3B	16.6	22.6	14.1	18.0
PACE-Gemma	31.6	30.4	40.5	32.5
PACE-Qwen	**35.2**	32.1	40.4	**35.1**

Table 6: StreamingBench Proactive Split (PO) time accuracy (follow official leaderboard) & our interactive protocol on same split (PO-I). **Best** and second-best marked in each column.

Model	PO	PO-I
Gemini 1.5 Pro	45.10	–
GPT-4o	39.22	–
Claude 3.5 Sonnet	35.29	–
VideoLLM-Online	1.96	–
VITA-1.5	25.49	–
Dispider	25.34	–
InternVL2	20.45	–
MiniCPM-o 2.6	29.41	30.27
MMDuet2	34.69	5.95
VideoChat3	9.80	18.38
Gemma4-A4B	36.80	38.9
Qwen3.5-A3B	25.20	15.1
PACE-Gemma	44.00	48.6
PACE-Qwen	**47.60**	**54.6**

Table 7: OVO-Bench Forward CRR Task. Baseline results follow Li et al. (2025). Our methods highlighted gray. **Best** and second-best marked.

Model	CRR
Gemini 1.5 Pro	61.67
GPT-4o	59.40
LLaVA-Video-7B	60.42
LLaVA-OV-7B	58.75
Qwen2-VL-72B	45.00
Qwen2-VL-7B	48.75
Gemma4-A4B	59.58
Qwen3.5-A3B	56.67
PACE-Gemma	60.83
PACE-Qwen	**62.50**

Table 8: **Annotation & Data Scale Ablation.** PAUC-F ($\omega = 0.5$) on EGO & WEB for PACE-Gemma. Each stage of our dataset construction (human review and filtering) improves performance. Results also scale when training with increasing data.

Version	Frac.	EGO ↑	WEB ↑
Raw Gemini-generated	100%	54.8	42.4
Intermediate, pre-cleaning	100%	55.4	42.9
Final human-verified	100%	56.5	43.8
Final human-verified	10%	53.8	42.3

Table 9: **Inference Harness Ablation.** PAUC-F ($\omega = 0.5$) on WEB and EGO for Gemma4-A4B. Baseline from VideoChat3 for comparison purposes.

Method	WEB	EGO
Ours (Default)	43.4	36.0
Ours w/o sliding window	33.6	7.0
Ours w/o online memory	36.8	14.1
Ours w/o structured output	42.2	34.6
VideoChat3 Harness	26.3	2.4

Table 10: **Ablation on Training Pipeline.** PAUC-F ($\omega = 0.5$) on WEB & EGO for PACE-Gemma. `Memory` and `Hold-State` help avoid repetitive firing. `Loss-Masking` is crucial to avoid collapsing to silence.

Memory	Hold State	Loss Masking	WEB ↑	EGO ↑
✓	✓	✓	56.5	43.8
✗	✓	✓	54.8	40.5
✓	✗	✓	32.9	24.4
✓	✓	✗	0.0	0.0

performance. We additionally introduce an interactive split (PO-I) where new queries are introduced mid-video in a true streaming fashion (more details in Appendix B). Baselines are evaluated using official checkpoints under identical settings as our models.

In Table 7, we report accuracy for the CRR (Clues Reveal Responding) task of OVOBench. In this task, a query is provided before the video contains sufficient information to respond, and models are expected to wait until evidence appears and then proactively respond. This is the key task in OVOBench that measures proactive responsiveness; hence we use it in our evaluations. In this task too, our PACE-Qwen model achieves overall best performance.

6.2 Ablative Studies

Dataset. We establish the contribution of our human review and filtering during dataset construction in Table 8. Results also highlight how performance scales going from 10% to 100% of data in PACE-780K for model training.

Inference Harness. In Table 9, we first ablate the key components of our proposed inference harness to verify their usefulness. Next we compare against an existing generic harness design from Li et al. (2026) to demonstrate our gains in proactive responsiveness.

Training Pipeline. We ablate our training pipeline in Table 10. Both our memory (past queries and responses) and `hold` state play a role in avoiding repetitive firing (more details in Appendix A.1). Loss masking is crucial to prevent models from collapsing to always remaining silent, as query-relevant events are sparse and the decision-token distribution is heavily skewed toward silence.

7 Conclusion

Proactive streaming video agents must decide between responding and remaining silent repeatedly for each timestep. We formulate this task as a series of causal response decisions at each timestep. Our PACE framework unifies data, metric, harness, and training around this formulation and demonstrates state-of-the-art performance across four proactive streaming benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools for following tasks: generate synthetic data sets, implement methods, clean and reformat dataset, propose or refine hypotheses, design or provide feedback on research methodology or experiments. To reiterate, as explained in the paper, generative AI tools were used for automated preliminary dataset creation (part of our contribution) and automated inference harness tuning (part of our contribution). We have not used generative AI tools for the following tasks: help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, support qualitative and thematic data analysis, interpret results. The rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for the following: formulate questions for surveys or interviews, create or modify scientific figures or images, create or edit software code, summarize or analyse existing literature, sourcing/searching for information, identify relevant literature, format references, propose a title. We have reviewed all AI-assisted work. Particularly, all AI generated code and AI assisted literature survey have been verified by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our data, code, and trained model checkpoints will be released publicly. For closed source models, all generation outputs will be released. For open models, we use publicly available checkpoints in all our evaluations.

REFERENCES

Steven Abreu, Tiffany D. Do, Karan Ahuja, Eric J. Gonzalez, Lee Payne, Daniel McDuff, and Mar Gonzalez-Franco. PARSE-Ego4D: Toward bidirectionally aligned action recommendations for egocentric videos. In *ICLR Workshop on Bidirectional Human-AI Alignment*, 2025.

Shehreen Azad, Vibhav Vineet, and Yogesh S. Rawat. StreamReady: Learning what to answer and when in long streaming videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40494–40504, 2026.

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

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Donglin Bai, Zhibo Chen, and Ting Cao. StreamMind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13448–13459, 2025.

Wirth F. Ferger. The nature and use of the harmonic mean. *Journal of the American Statistical Association*, 1931.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, et al. VITA-1.5: Towards GPT-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025.

Gemma Team. Gemma 4 Technical Report, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3D in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, Xinyu Chen, Haoran Chen, Zizheng Huang, Jun Zhang, Kun Ouyang, Lin Sui, Ziang Yan, Yicheng Xu, Chenting Wang, Yinan He, Hongjie Zhang, Yi Wang, Yu Qiao, Yali Wang, Ziwei Liu, Kai Chen, and Limin Wang. Videochat3: Fully open video mllm for efficient and generalist video understanding, 2026.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. In *Advances in Neural Information Processing Systems*, 2022.

OpenBMB. MiniCPM-o 2.6: A GPT-4o level MLLM for vision, speech, and multimodal live streaming on your phone. Technical report, 2025. URL `https://openbmb.notion.site/185ede1b7a558042b5d5e45e6b237da9`.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video LLMs with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24045–24055, 2025.

Qwen Team. Qwen3.5-35B-A3B. Hugging Face model card, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Thinking Machines Lab. Inkling-small. `https://thinkingmachines.ai/news/inkling-small/`, 2026.

Xueyun Tian, Wei Li, Bingbing Xu, Heng Dong, Yuanzhuo Wang, and Huawei Shen. ROMA: Real-time omni-multimodal assistant with interactive streaming understanding, 2026.

C.J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. StreamBridge: Turning your offline video large language model into a proactive streaming assistant. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-4406.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models, 2025b.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6338–6359. Association for Computational Linguistics, 2025c. doi: 10.18653/v1/2025.findings-emnlp.336.

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. In *International Conference on Learning Representations*, 2026.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31219–31229, 2026.

Guoyang Xu and Hao Chen. VideoHarness-RSI: Recursive harness self-improvement for long-video understanding with frozen vision-language models, 2026.

Zhenyu Yang, Kairui Zhang, Yuhang Hu, Bing Wang, Shengsheng Qian, Bin Wen, Fan Yang, Tingting Gao, Weiming Dong, and Changsheng Xu. LiveStar: Live streaming assistant for real-world online video understanding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1050.

Dingyu Yao, Shuhuan Gu, Qingyi Si, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Naibin Gu, Zheng Lin, Weiping Wang, Nan Duan, and Jiaqi Wang. Harnessing streaming video in the wild, 2026.

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 *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 2025.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM-V: A GPT-4V level MLLM on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, Wenhao Hu, Bingxiang He, Jie Zhou, Jie Cai, Ji Qi, Zonghao Guo, Chi Chen, Guoyang Zeng, Yuxuan Li, Ganqu Cui, Ning Ding, Xu Han, Yuan Yao, Zhiyuan Liu, and Maosong Sun. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Xueyang Yu, Cheng Shi, Yang Wang, and Sibei Yang. Eyes wide open: Ego proactive Video-LLM for streaming video. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-0453.

G.U. Yule and M.G. Kendall. *An Introduction to the Theory of Statistics*. C. Griffin, limited, 1947.

Kecheng Zhang, Zongxin Yang, Mingfei Han, Haihong Hao, Yunzhi Zhuge, Changlin Li, Junhan Zhao, Zhihui Li, and Xiaojun Chang. Progressive online video understanding with evidence-aligned timing and transparent decisions. In *International Conference on Learning Representations*, 2026a.

Luan Zhang, Ruochen Zhou, Dandan Song, Zhengyu Chen, Yuhang Tian, Jun Yang, Huipeng Ma, Chenhao Li, Guangyuan Feng, Xudong Li, Yizhou Jin, and Yan Xu. HarnessCompass: Guiding automatic harness evolution toward generalizable and effective agent harnesses, 2026b.

Yifan Zhang, Yutong Dai, Juntao Tan, Luyu Yang, Rishi Mullur, Thai Hoang, Zhiyuan Hu, James Zhu, Phil Mui, Silvio Savarese, Ran Xu, and Zeyuan Chen. DarwinX: Evolving agent harnesses through natural selection, 2026c.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6586–6597, 2023.

Yikai Zheng, Xin Ding, Yifan Yang, Shiqi Jiang, Hao Wu, Qianxi Zhang, Weijun Wang, Ting Cao, and Yunxin Liu. Em-garde: A propose-match framework for proactive streaming video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

APPENDIX CONTENTS

A Additional Experiments **14**

B Interactive PO Task **16**

C Additional Dataset Details **17**

D PAUC-F Metric Details **19**

E Inference Harness Details **22**

F Training Pipeline Details **23**

A ADDITIONAL EXPERIMENTS

We report additional results proactive streaming video benchmarks, ProactiveVideoQA and ESTP-Bench as well as more splits of OVO-Bench. We also provide additional ablations comparing our method to existing harness designs and experiments involving our harness tuning.

Baselines. We use official checkpoints of Molmo2 8B (Clark et al., 2026), MiniCPM-V 4.5 (Yu et al., 2026), Inkling-Small (Thinking Machines Lab, 2026), MiniCPM-o 2.6 (OpenBMB, 2025), VideoLLM-Online (Chen et al., 2024a), MMDuet (Wang et al., 2025c), MMDuet2 (Wang et al., 2026), EyeWO (Yu et al., 2025), and VideoChat3 (Li et al., 2026) for the baselines we report in our evaluations. We also borrow numbers directly from prior evaluations in published papers for LLaVA-OneVision (Li et al., 2024a), LLaVA-NeXT-Video (Li et al., 2024b), LLaVA-Video (Zhang et al., 2024), Qwen2-VL (Wang et al., 2024), MiniCPM-V (Yao et al., 2024), InternVL2 (Chen et al., 2024b), CLIP (Radford et al., 2021), LaViLa (Zhao et al., 2023), EgoVLP (Lin et al., 2022), VITA-1.5 (Fu et al., 2025), and Dispider (Qian et al., 2025).

ProactiveVideoQA PAUC. In Table A.1, we report the original PAUC metric Wang et al. (2025b) based evaluations for the same methods in Table 3. We highlight that the zero-point of PAUC is 25.0 (i.e. a model with no responses) and models scoring below 25 are worse than fully silent models (according to PAUC). Also note how models like Gemini-2.5-Flash (64.51) outperform the Gemini-2.5-Pro model (55.07) at $\omega = 0.5$ in the Mean PAUC column, misaligning with our priors about these models' strengths. We reiterate our observations (see Figure 2) how models can game this metric with excessive firing, sacrificing precision for increased recall.

Nevertheless, it is interesting to note that our models (PACE-Gemma and PACE-Qwen) optimized for PAUC-F also achieve strong PAUC numbers while reducing excessive firing (see Table 2), with PACE-Qwen achieving state-of-the-art performance among non-proprietary models.

ESTP-Bench. We also report the full per-task breakdown of ESTP-Bench ESTP-F1 (official metric) scores in Table A.2. These numbers correspond to the results summary in Table 5 of main paper. Our PACE-Qwen showcases state-of-the-art overall performance on this benchmark as well.

We note how on *Implicit* and *Contextual* splits several models outperform both our models by large margins. We argue this is due to weaknesses of the ESTP-F1 evaluation metric. Across the three splits, source videos are of similar length, but *Implicit* and *Contextual* gold windows are much longer (Table A.5). Models that repeat fires get penalized much less with the longer golden windows under ESTP-F1 metric. We highlight how our models outperform prior best EyeWO model across all 3 splits when evaluating with the more balanced PAUC-F metric in Table A.3. Table A.4 shows why: EyeWO baseline emits an order of magnitude more fires per item than the ground-truth, especially on *Implicit* and *Contextual* splits, while PACE-Gemma and PACE-Qwen stay near the gold rate. Measuring repeats in this way to affirm prediction quality has been used in the prior work MMDuet2 (Wang et al., 2026) as well.

A.1 HARNESS ABLATIONS

We next provide additional ablations on our harness design. We first compare against existing harness designs from Wang et al. (2025a); Yao et al. (2026) in Table A.6. Then we compare against harness tuning with a general strategy built following Zhang et al. (2026c) without our explicit separation of harness components used in our tuning pipeline. This comparison is also included in

Table A.1: **PAUC on ProactiveVideoQA Benchmark.** We report PAUC with varying ω using GPT-4.1 judge, for the same runs, predictions, and judge configuration as Table 3. PAUC is the recall component of PAUC-F and does not penalize the number of responses emitted, so these values should be read together with Table 3. Models with excessive responsiveness (over-firing) can score higher on this metric than conservative models with balance precision and recall. Harness is ours for offline models and default for streaming native models. Our harness is modified to integrate with Gemini-Live (ours-v) due to closed API nature. **Best** and second-best are marked separately within proprietary models and within the remaining models.

Model	Harness	[WEB]			[EGO]			[TV]			[VAD]			[Mean]		
ω =		0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
Proprietary Models																
GPT-4.1	ours	56.01	60.42	64.83	48.62	54.06	59.51	56.44	65.57	74.70	**50.00**	**55.49**	60.98	52.77	58.89	65.01
GPT-4.1-mini	ours	47.22	51.67	56.12	54.44	60.49	66.54	55.31	64.25	73.20	45.27	50.20	55.14	50.56	56.65	62.75
GPT-5.6-Sol	ours	52.54	57.26	61.97	49.44	55.04	60.63	52.15	60.90	69.65	43.28	47.58	51.87	49.35	55.19	61.03
Gemini-2.5-Pro	ours	51.93	56.63	61.33	46.32	51.80	57.29	54.05	63.20	72.35	43.79	48.65	53.50	49.02	55.07	61.12
Gemini-2.5-Flash	ours	**60.62**	**67.00**	**73.38**	62.40	68.78	75.16	57.61	67.25	76.90	48.31	55.00	**61.68**	**57.23**	**64.51**	**71.78**
Gemini-3.1-Pro	ours	56.25	63.07	69.90	61.44	67.72	74.00	56.82	66.31	75.80	25.52	26.11	26.70	50.00	55.80	61.60
Gemini-3.5-Flash-Lite	ours	55.83	62.59	69.35	**64.67**	**71.57**	**78.48**	53.29	61.15	69.00	23.42	23.28	23.13	49.30	54.65	59.99
Gemini-3.8-Flash	ours	56.80	63.66	70.52	60.37	67.32	74.27	**58.29**	**68.32**	**78.35**	25.13	25.35	25.57	50.15	56.16	62.18
Gemini-Live-2.5	default	24.10	24.10	24.10	25.00	25.00	25.00	20.31	20.31	20.31	25.00	25.00	25.00	23.60	23.60	23.60
Gemini-Live-2.5	ours-v	41.06	50.74	60.42	46.66	52.15	57.64	16.42	17.58	18.75	47.91	52.08	56.25	38.01	43.14	48.27
Open-Weight Models (No Streaming Video Training)																
Molmo2 8B	ours	34.80	38.80	42.81	35.39	37.81	40.24	28.70	30.23	31.75	26.99	27.86	28.74	31.47	33.68	35.88
MiniCPM-V 4.5	ours	40.43	46.22	52.01	39.07	42.87	46.67	23.11	23.83	24.55	19.40	19.75	20.09	30.50	33.17	35.83
Inkling-Small	ours	40.87	43.96	47.04	**43.31**	**48.54**	**53.76**	32.84	34.77	36.70	30.59	31.30	32.01	36.90	39.64	42.38
Gemma4-A4B	ours	**52.45**	**57.30**	**62.14**	34.81	37.57	40.32	28.70	29.75	30.80	34.26	36.41	38.55	37.56	40.26	42.95
Qwen3.5-A3B	ours	37.04	39.53	42.02	38.43	42.10	45.76	33.32	35.48	37.65	33.28	35.68	38.08	35.52	38.20	40.88
Streaming Video Models (Trained for Streaming Responses)																
VideoLLM-Online	default	26.20	26.21	26.22	25.08	25.10	25.13	16.92	17.46	18.00	25.00	25.00	25.00	23.30	23.44	23.59
EyeWO	default	30.25	32.03	33.81	39.84	41.32	42.79	20.91	20.48	20.05	22.49	21.41	20.33	28.37	28.81	29.25
MMDuet	default	43.13	46.97	50.81	41.51	43.62	45.73	22.13	23.14	24.15	24.18	24.36	24.53	32.74	34.52	36.31
MMDuet2	default	43.26	52.48	61.71	27.98	28.99	30.00	37.50	44.10	**50.70**	28.41	30.56	32.71	34.29	39.03	43.78
VideoChat3	default	28.74	30.26	31.78	29.00	30.77	32.54	32.98	37.44	41.90	27.03	28.23	29.44	29.44	31.68	33.91
PACE-Gemma (ours)	ours	40.00	45.26	50.51	41.42	45.52	49.62	35.36	39.68	44.01	34.16	37.06	39.95	37.74	41.88	46.02
PACE-Qwen (ours)	ours	42.57	48.43	54.29	41.63	45.96	50.29	**40.43**	**45.47**	50.50	**36.47**	**40.43**	**44.39**	**40.28**	**45.07**	**49.87**

Table A.2: **ESTP-F1 score on Full ESTP-Bench.** We report complete per-task ESTP-Bench results corresponding to the category-level summary in Table 5 of our main paper. Baselines are from Yu et al. (2025) and our 4 models evaluated following identical settings. **Best** and second-best are marked in each column.

Model	Explicit Proactive Task									Implicit Proactive Task					Contextual Q			Overall
	OR	AP	TRU	OL	OSC	EOL	EOSC	AR	All	OFR	IFR	NAR	TU	All	ORC	TRC	All	
Offline MLLMs Response-in-Last																		
LLaVA-OneVision	7.2	11.5	4.9	10.0	4.9	6.9	5.6	3.2	6.8	3.8	6.3	11.6	29.8	12.9	10.8	5.7	8.2	8.7
Qwen2-VL	11.7	8.1	14.9	10.5	1.7	8.9	10.6	6.0	9.0	10.2	4.4	26.5	49.5	22.6	13.3	9.4	11.3	13.3
MiniCPM-V	12.3	12.6	10.7	13.7	8.6	7.5	11.9	5.5	10.4	11.8	9.2	36.0	55.3	28.1	32.6	25.4	29.0	18.1
LLaVA-NeXT-Video	8.3	9.4	7.4	10.2	7.8	7.4	10.3	5.6	8.3	6.4	6.7	21.1	45.9	20.0	10.1	9.8	9.9	11.9
InternVL-V2	9.3	14.6	9.5	10.6	1.7	6.3	3.0	3.6	7.3	3.3	9.2	15.5	28.2	14.0	16.9	15.6	16.2	10.5
VLMs for Streaming Detection																		
CLIP	7.3	9.5	7.4	8.5	1.8	4.7	2.2	2.7	5.5	2.8	5.2	51.3	29.3	22.2	4.6	3.8	4.2	10.1
LaViLa	8.4	10.7	9.0	9.1	3.1	5.4	3.6	4.3	6.7	7.8	10.0	56.2	34.4	27.1	9.4	28.9	19.2	14.3
EgoVLP	10.5	11.0	8.7	8.5	5.5	5.6	5.3	4.4	7.4	6.2	10.7	58.4	48.3	30.9	8.0	25.3	16.6	15.5
Offline MLLMs Polling Strategy																		
LLaVA-OneVision	8.3	8.8	22.8	25.4	13.5	9.8	9.6	10.3	13.6	20.3	20.9	35.9	49.9	31.8	14.6	1.9	8.2	18.0
Qwen2-VL	13.7	13.5	15.4	29.5	8.0	15.4	16.6	10.9	15.4	17.8	19.8	56.4	63.1	39.3	13.0	7.7	10.4	21.3
MiniCPM-V	14.9	16.8	17.1	26.8	7.7	12.9	12.5	13.1	15.2	15.9	21.0	46.8	62.2	36.5	24.3	28.9	26.6	22.9
LLaVA-NeXT-Video	15.6	14.6	21.9	26.8	12.8	14.2	13.5	12.3	16.5	18.6	23.2	44.9	51.6	34.6	19.9	7.7	13.8	21.3
InternVL-V2	11.3	5.9	7.0	10.1	0.7	2.7	5.2	2.2	5.6	8.3	2.9	4.3	11.2	6.7	6.1	5.3	5.7	5.9
Online MLLMs																		
LIVE(threshold=0.8)	9.7	11.0	7.4	10.8	1.9	6.0	3.6	5.6	7.0	4.2	7.4	12.9	12.8	9.3	19.6	13.8	11.8	9.1
LIVE(threshold=0.9)	11.2	13.9	7.9	13.2	5.6	9.4	6.0	8.9	9.5	5.8	8.9	41.0	46.7	25.6	11.3	26.5	18.9	15.5
MMDuet	7.2	10.3	17.6	10.2	4.2	6.1	8.8	8.5	9.1	10.0	7.7	50.1	69.1	34.2	17.4	23.1	20.3	17.8
EyeWO	26.6	26.6	25.1	26.8	19.8	22.3	20.8	20.7	23.6	24.8	**31.0**	**75.3**	**78.7**	**52.5**	39.5	**47.8**	**43.6**	34.7
Ours																		
Gemma4-A4B	19.4	23.5	24.0	20.9	10.6	16.8	13.1	17.4	18.2	12.6	16.1	44.1	52.0	31.2	21.0	33.3	27.1	23.2
Qwen3.5-A3B	14.4	15.5	22.9	19.7	16.9	12.2	20.0	11.1	16.6	19.9	14.4	14.5	41.8	22.6	16.1	12.2	14.1	18.0
PACE-Gemma	30.3	24.7	**38.9**	33.5	28.3	31.5	32.8	32.9	31.6	26.4	30.5	28.4	36.3	30.4	41.4	39.6	40.5	32.5
PACE-Qwen	**32.8**	**31.8**	36.9	**35.6**	**32.2**	**34.5**	**38.6**	**39.1**	**35.2**	**27.6**	30.4	25.3	45.3	32.1	**45.8**	34.9	40.4	**35.1**

Table A.6. These results demonstrate how our gains through automated harness tuning are non-trivial and result in a harness outperforming prior work on proactive streaming tasks.

We also report the full numbers for harness tuning on the PACE-780K validation set for our Gemma4-A4B model in Figure A.1.

Table A.3: **PAUC-F on Full ESTP-Bench.** We report per-task PAUC-F ($\omega = 0.5$, GPT-4.1 judge) on the full ESTP-Bench. **Best** and second-best are marked in each column. EyeWO (Yu et al., 2025) official checkpoint evaluated under identical settings.

Model	Explicit Proactive Task									Implicit Proactive Task					Contextual Q			Overall
	OR	AP	TRU	OL	OSC	EOL	EOSC	AR	All	OFR	IFR	NAR	TU	All	ORC	TRC	All	
EyeWO	10.86	7.08	3.43	10.72	7.16	10.20	4.56	11.23	8.16	17.48	18.35	5.76	6.80	12.10	16.22	6.64	11.43	9.75
Gemma4-A4B	11.05	16.90	15.59	14.07	7.52	11.16	6.41	10.33	11.63	9.66	10.70	9.92	13.43	10.93	11.51	14.02	12.77	11.59
Qwen3.5-A3B	23.88	27.17	23.07	29.78	29.24	23.10	29.87	23.47	26.20	**29.49**	23.28	**31.65**	35.15	29.90	35.67	28.46	32.07	28.09
PACE-Gemma	20.23	25.97	30.06	**29.83**	20.89	24.41	17.73	21.28	23.80	23.43	30.27	26.51	35.95	29.04	32.36	25.61	28.99	26.04
PACE-Qwen	**24.62**	**31.77**	**30.47**	29.08	**30.82**	**27.84**	**31.64**	**24.24**	**28.81**	28.56	**36.56**	27.07	**36.42**	**32.15**	**38.87**	**29.50**	**34.19**	**30.53**

Table A.4: **Golden Fires per query on Full ESTP-Bench.** We report mean assistant fires per query, so repeated and rephrased responses both count. A query denotes one question on *Explicit* and *Implicit* splits, and one multi-question session on *Contextual* split. The first row calculates Gold Rate by assigning a single response to each gold window, which is how an ideal assistant is expected to respond.

Model	Explicit Proactive Task									Implicit Proactive Task					Contextual Q			Overall
	OR	AP	TRU	OL	OSC	EOL	EOSC	AR	All	OFR	IFR	NAR	TU	All	ORC	TRC	All	
Gold Rate	3.98	4.63	2.22	2.67	4.77	3.45	2.10	6.29	4.07	2.73	2.73	5.34	6.34	4.45	2.86	15.10	8.87	5.41
EyeWO	15.43	17.08	8.83	11.23	25.45	17.36	11.82	23.73	17.12	13.81	10.41	96.02	109.93	62.61	19.01	234.29	124.65	55.28
Gemma4-A4B	19.72	12.96	14.25	17.79	33.29	25.57	33.45	39.48	25.65	26.15	19.60	49.41	32.92	33.33	23.03	48.90	35.75	29.98
Qwen3.5-A3B	1.02	0.92	0.96	1.02	0.86	1.24	1.10	1.06	1.04	1.34	1.21	0.84	3.81	1.86	0.58	2.19	1.37	1.31
PACE-Gemma	4.97	5.01	2.36	4.51	7.00	6.67	5.07	7.05	5.53	3.68	3.19	4.21	4.25	3.90	4.29	9.72	6.96	5.55
PACE-Qwen	4.57	3.58	1.89	4.14	3.09	3.58	2.87	5.73	3.94	2.29	1.75	3.83	5.20	3.43	3.17	8.29	5.69	4.29

Table A.5: **Gold-window lengths across ESTP-Bench splits.** Explicit, Implicit, and Contextual items use source videos of similar length, but their gold windows differ sharply. Compared to explicit, implicit gold windows are $3.85\times$ longer and contextual gold windows are $3.14\times$ longer.

Split	Queries	Average Video Length	Average Gold Window	Total Gold-Window Time / Query
Explicit	851	29:09	4.81 s	19.99 s
Implicit	361	30:29	18.53 s	84.07 s
Contextual	430	33:34	15.08 s	133.70 s

Table A.6: **Additional Ablation on Harness Design.** PAUC-F ($\omega = 0.5$) on WEB and EGO for Gemma4-A4B. We compare our harness against a harness tuned by general harness evolution (Zhang et al., 2026c), and harnesses from StreamBridge (Wang et al., 2025a) and Yao et al. (2026). All baselines are implemented by us and evaluated under identical settings.

Method	WEB	EGO
Ours (Default)	43.4	36.0
StreamBridge Harness	32.5	6.2
Streaming Harness	28.5	12.7
General Harness Evolution	40.2	31.4

B INTERACTIVE PO TASK

Existing proactive evaluation tasks contain a single query asked at start of the video and their established inference strategy does not involve injecting new queries mid video. We introduce our PO-I task using the same videos in the PO split of StreamingBench Lin et al. (2024), where each video has up to 5 queries injected during the video. This requires agentic models to handle multiple queries at different times of the videos, introducing several new challenges.

We build each PO-I episode from one video in the StreamingBench PO split. The video is streamed once, and its five annotated queries are injected at their original timestamps. We score every query with the same time-accuracy rule as the official PO split. The key difference lies in the inference setting, which injects queries interleaved with video frames, making the task more difficult. In these videos, each next query arrives only after the preceding gold response, so PO-I evaluates



Figure A.1: **Automated Harness Tuning.** We plot all 100 evaluations on our PACE-780K validation with various harness design candidates for Gemma4-A4B model. The selected models for each round are highlights (blue). The full tuning process achieves a gain of 12.85 PAUC-F points on the validation data.

completed-request updates. The model must switch to the newly injected query while earlier frames and responses remain in context, and must keep later responses specific to that query.

We note that our training videos do not contain this style of queries mid video. However, our memory design allows model to condition its behavior on the current memory state. This is enforced in the harness and training as well. This allows the model to generalize zero-shot to multi-query interactions during inference, which aligns with positive results we obtain for this task.

C Additional Dataset Details

Dataset Splits. We report statistics on the data splits of PACE-780K dataset used for our training and validation in Table A.7.

Table A.7: **PACE-780K Additional Details.** Training and validation split details of our PACE-780K dataset.

Split	Videos	Questions	Responses
Train	176,021	176,021	777,278
Validation	1,768	1,768	7,666
Total	177,789	177,789	784,944

C.1 Dataset Generation Details

Automated Proactive Query-Response Generation: For a given video V, we automatically generate a proactive interaction in two successive model calls.

First, an instruction-generation model observes the complete video and synthesizes a single video-specific user query q. The query requests notifications about a concrete and visually verifiable event, action, or state change (e.g., an object being picked up or a person entering the scene). We require each query to define a specific, visually observable trigger, rather than requesting general video narration or relying on subjective criteria such as "interesting" events.

Next, a response-generation model receives V and q and produces a temporally ordered sequence $\mathcal{R} = \{(t_i, r_i)\}_{i=1}^{N}$, where t_i is the trigger time and r_i is a concise natural-language response. To

ensure that the generated interactions are representative of real-time deployment, we constrain the model to behave as an online observer for which future frames are unavailable. Although the complete video is provided to the model, we prompt the model and require it to emulate causal streaming observation: each response must rely only on evidence visible up to t_i, must not anticipate future events, and must be emitted at the first frame at which the query condition is satisfied. This constraint reduces hindsight bias and encourages responses to be triggered as soon as sufficient visual evidence becomes available. The model remains silent between triggers and returns an empty sequence when no valid trigger is detected throughout the video.

To balance computational cost with the need for fine-grained temporal reasoning, we use different models for the two generation steps. Specifically, we use `gemini-3-flash-preview` for query generation and `gemini-3-pro-preview` for response generation. We remove the audio track and sample each video at 1 FPS. Consequently, trigger times are annotated at one-second resolution. The generated queries, responses, timestamps, and associated metadata are stored in structured JSON format and subsequently verified and refined during the human-review stage.

Human Annotation Repair: Human annotators first assess the validity of each generated query. A valid query must specify a concrete and visually recognizable condition that can be detected causally from a streaming video and can therefore support a well-defined response decision. We reject queries whose triggers are ambiguous, subjective, not visually observable, or dependent on future context. Only samples with valid queries proceed to response-level annotation repair.

For each valid query, annotators inspect the timing and content of every generated response and repair the annotations when necessary. A valid response must be relevant to the query, visually grounded at its assigned timestamp, free of hallucinated content, and consistent with causal observation. In particular, the response should be unambiguously supported by the frames observed up to that time, while it may use preceding context but must not reveal information from future frames. Annotators correct inaccurate text, shift temporally misaligned responses, remove spurious responses, and add missed responses as needed.

For instantaneous events, the response is generated at the moment when the event becomes visually identifiable. For continuous events that persist over a period of time, annotators additionally enforce semantic continuity across consecutive responses. For example, later responses use expressions such as "continues" or "still" to indicate that the previously observed event remains ongoing.

Quality Control: A professional video data-annotation team recruits and trains the annotators, who receive detailed instructions and examples before performing annotation repair. We then conduct quality control at the batch level using PhD-level expert reviewers who are independent of the annotation repair process.

At the beginning of the human annotation process, we partition the automatically generated samples into ten disjoint batches and process them sequentially for human annotation repair. After the annotators complete a batch, independent expert reviewers audit a randomly selected subset of videos and their annotations using a purpose-built web interface. The reviewers examine the complete video timeline. They evaluate the annotation at each one-second time step, treating both the presence and absence of a response as a decision to assess. A response is labeled a true positive if the reviewer confirms that it is warranted and its content is correct, whereas a no-response decision is labeled a true negative if the reviewer agrees that no response is warranted. An unwarranted or incorrect response is labeled a false positive, and a missing response is labeled a false negative.

We quantify annotation quality using the F1 metric. A batch is accepted, and annotation proceeds to the next batch, only if $F1 > 0.95$. Batches that do not meet this threshold undergo further correction and review. During the initial batches, we have repeated the cycle of annotator training, annotation, and independent evaluation multiple times to calibrate the annotation criteria and establish consistent quality before processing the remaining batches.

Dataset Review Tool We use the purpose-built interface shown in Figure A.2 for frame-level review of the generated annotations. Reviewers step through the video at one-second intervals and flag problematic frames as missing, hallucinated, boundary cases, or incorrect deletion decisions. Unmarked frames are recorded as correct when the reviewer completes the video, enabling efficient and consistent auditing of both responses and silence decisions.

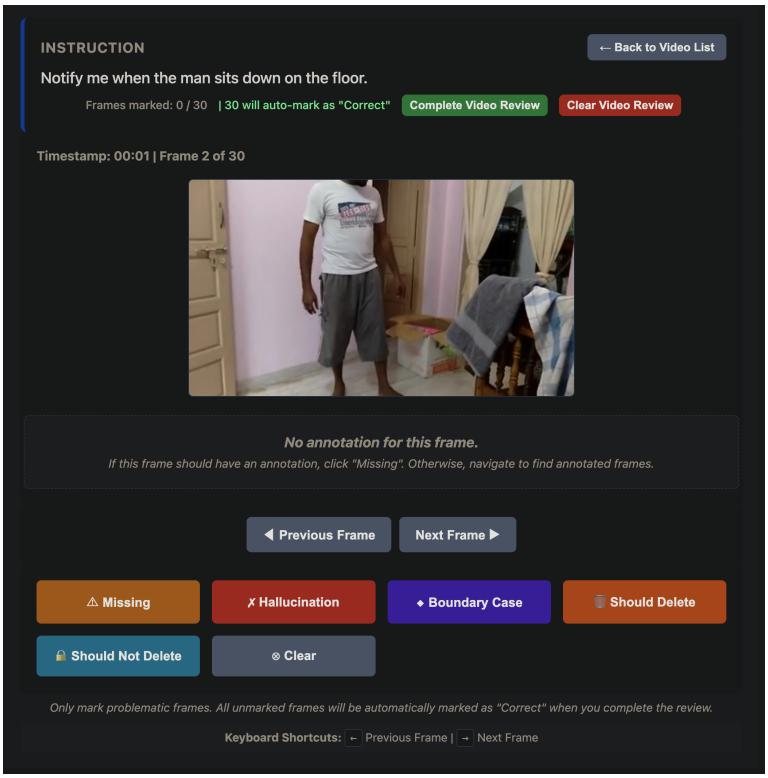

Figure A.2: **Dataset review interface.** Reviewers inspect each sampled video frame against the instruction and existing annotation, navigate temporally, and mark only frames requiring correction. The interface records missing responses, hallucinations, temporal boundary cases, and deletion decisions while automatically treating unmarked frames as correct.

D PAUC-F Metric Details

We discuss the behavior of proposed PAUC-F compared to the parent PAUC metric and then present details on our human study.

Behavior Compared to PAUC. PAUC (Wang et al., 2025b) scores each gold turn on its reply window $(t^{\text{start}}, t^{\text{end}})$. In-window replies at $\tau_1 < \cdots < \tau_P$ get judge scores s_p, and PAUC is the area under that time–score polyline divided by the maximum area $(t^{\text{end}} - t^{\text{start}})S$, with $S = 2$:

$$\text{PAUC} = \frac{(\tau_1 - t^{\text{start}}) \cdot 0.5 + \sum_{p=1}^{P-1} (\tau_{p+1} - \tau_p)\, s_p + (t^{\text{end}} - \tau_P)\, s_P}{(t^{\text{end}} - t^{\text{start}})\, S}. \tag{2}$$

With no in-window reply this ratio is fixed at $0.5/S = 0.25$. Before the area is taken, timestamps are shifted by $\omega \in [0, 1]$,

$$\tau'_p = t^{\text{start}} + (1 - \omega)(\tau_p - t^{\text{start}}), \tag{3}$$

and τ'_p replaces τ_p in Equation (2). $\omega = 0$ leaves time unchanged, so a late reply adds little area. $\omega = 1$ moves every τ_p to t^{start}, so PAUC reduces to s_P/S and timing drops out. Our PAUC-F metric inherits the ω term from PAUC, which allows adjusting the weight between quality and timeliness.

Repeating the definition from Section 4,

$$R = \text{PAUC}, \qquad P = \frac{N_{\text{value}}}{N_{\text{fires}}}, \qquad \text{PAUC-F} = \frac{2PR}{P + R}. \tag{4}$$

A model that never fires has $P = 0$. PAUC still returns its silence floor 0.25, but the harmonic mean is zero, so PAUC-F $= 0$.

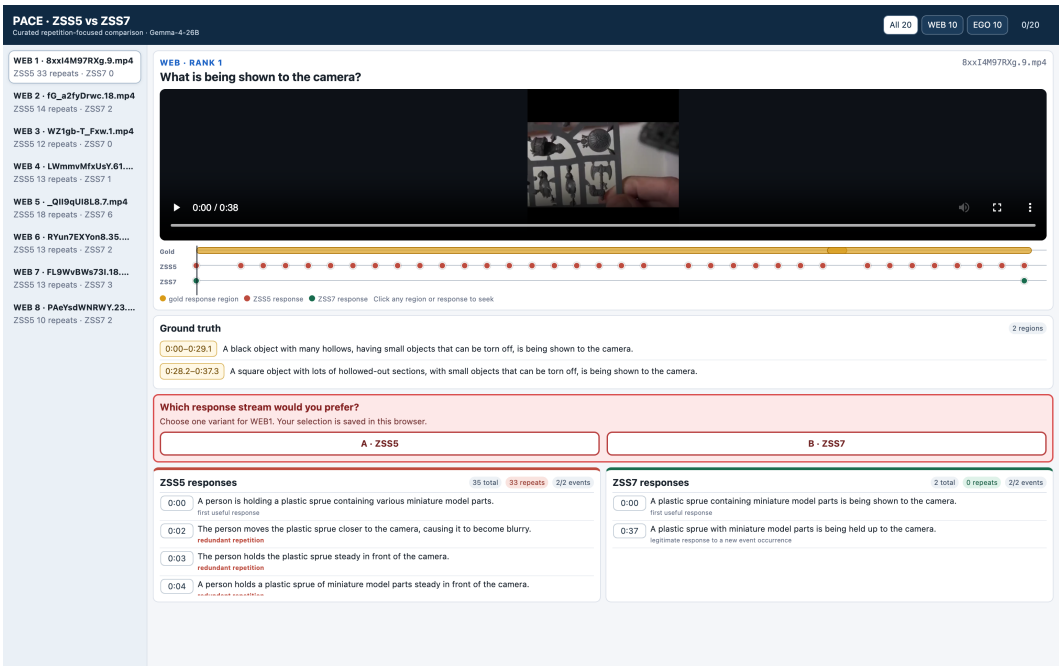

Figure A.3: **Human rating interface for PAUC-F.** Raters inspect the video, ground-truth response regions, and timestamped responses from two anonymized system variants, then select the preferred response stream. The aligned timeline and side-by-side response lists expose differences in event coverage, response timing, and repetitive firing behavior.

Table A.8: **Human Preference Study.** We present the exact results from each of the 10 human participants involved in our study. A clear preference for the PAUC-F optimized `v2` model was observed.

		Rater											
Split	Choice	1	2	3	4	5	6	7	8	9	10	Total	%
WEB	v1	0	0	1	0	0	4	3	0	2	2	12	6.0
	v2	20	20	19	20	16	17	18	20	18	18	188	94.0
EGO	v1	0	1	0	0	2	2	0	0	1	0	6	3.0
	v2	20	19	20	20	18	18	20	20	19	20	194	97.0

Human Study. We conduct a human preference study with ten raters using the interface shown in Figure A.3. Humans rate response streams from two distinct models across 20 videos from WEB and EGO splits of ProactiveVideoQA dataset. Raters are instructed to select the response that seems more helpful from a real-time video assistant and are shown the video, query, and ground-truth responses (content and intervals).

Human raters overwhelmingly select responses generated from agentic model `v2`. This model uses a harness tuned to maximize PAUC-F metric. The other `v1` model whose responses are less preferred by humans uses a harness tuned to maximize PAUC metric, often resulting in excessive repetition within windows and firing outside windows. The exact ratings from each human participant are provided in Table A.8.

Theoretical Motivation. Our PAUC-F metric is simply the harmonic mean between two established components, the PAUC metric from Wang et al. (2025b) and firing precision. We justify our use of the harmonic mean with two key reasons.

First, the harmonic mean is the strictly correct operator for calculating the true average of rates or ratios (Yule & Kendall, 1947; Ferger, 1931). In our case, both PAUC and Firing Precision represent

Table A.9: Final harness shared by Gemma4-A4B and Qwen3.5-A3B. Model weights stay frozen.

Component	Configuration
Decision rate	1 Hz
Frames	Three most recent, $[k-2, k]$, truncated at the video start
Timestamps	Absolute `MM:SS` immediately before each frame
Question	Standing query; a later query enters only at its reveal time
Memory	Last accepted response and its emission time. Acceptance replaces it; silence leaves it unchanged
Decode	Greedy, at most 192 new tokens
Output	`[]` for silence, or one {`time, content`} object
Parser	Strip optional Markdown fences. Decode a JSON array, otherwise the first array-shaped span. Keep objects with a numeric `time` and non-empty `content`. Drop times outside ± 0.5 s, snap the rest to the current step, and keep at most one. Anything else is silence
Gemma input	Native system, query, and observation turns. Timestamps sit beside image embeddings. Generation starts after an empty model turn
Qwen input	Official chat template with thinking disabled. Query, memory, and timestamp–image pairs share one user message

Table A.10: Search and calibration budgets. A model-example evaluation is one candidate on one video.

Component	Full harness search	Limited calibration
Models	Gemma and Qwen	Other offline VLMs
Objective	Maximize validation PAUC-F	Maximize validation PAUC-F
Editable surface	Window, memory, prompt, and output protocol	Prompt and output schema only
Proposal schedule	10 candidates per round	1 candidate per round
Budget	10 rounds; 100 candidate evaluations/model	5 candidates $\times$ 20 examples/model
Samples per round	500	20
Feedback	PAUC-F Metric	PAUC-F Metric

Table A.11: Prompt Details.

Task	Prompt Description
Harness tuning	Tune a frozen VLM harness so validation PAUC-F rises. Each round provides the current harness, one hypothesis, the editable surface, and prior scores and diagnostics. Propose the budgeted edits on that surface, and state the expected effect of each. Edits may only change the window, memory, prompt, or output protocol. Keep an edit only when validation PAUC-F improves.
Harness calibration	Calibrate the prompt and output schema of a frozen harness so validation PAUC-F rises. Each round provides the fixed harness, the previous candidate, and its score and diagnostics. Propose one edit, and state its expected effect. Only the prompt and output schema may change. Keep an edit only when validation PAUC-F improves.
Inference (Final)	Streaming video assistant. Each second, the input is the last accepted answer and its time (empty if none) plus the latest frames, each marked with absolute `MM:SS`. The newest stamp is now. Speak only when those frames show a useful change; otherwise return an empty list and leave the last answer as it stands. A spoken turn is one short factual sentence tagged with the current time in seconds.
Finetuning	Streaming assistant over frames labeled with absolute `MM:SS` times. Emit `</Silence>` when the frame is irrelevant, `</Hold>` when the event continues but no new answer is due, and `</Response>` plus one complete answer when the frames suffice. A prior answer may be attached; repeat it only when the new frames add evidence. Stay within what the frames show.

rates of success. PAUC is the rate of accumulated quality-score over a bounded temporal window, while Firing Precision is the rate of valid fires over total fires.

Second, from an optimization perspective, the harmonic mean severely penalizes high variance between its inputs (Van Rijsbergen, 1979). For streaming video agents, timeliness (PAUC) and firing

restraint (Precision) are competing objectives; a model can easily maximize PAUC by spamming responses, driving Precision near zero. An arithmetic or geometric mean allows linear or proportional substitutability, allowing the model to "hide" terrible precision behind a perfect PAUC. The harmonic mean enforces that the marginal utility of maximizing PAUC drops to zero if Precision is sacrificed, mathematically guaranteeing that a high PAUC-F strictly requires Pareto-optimal balance between the two. This also mirrors the theoretical foundation of the F1 score in information retrieval.

E Inference Harness Details

We provide additional details on our inference harness in this section. Table A.9 is the harness used for every reported offline model. Table A.10 is the validation-only search that selected it, and Table A.11 presents the instruction prompts.

During harness tuning, each round tests ten edits of one hypothesis. An edit is promoted when validation PAUC-F improves on its parent. Combined promotions are kept only after that combination itself improves, and those confirmation runs count toward the budget in Table A.10.

Note that Table A.11 contains prompts relevant to finetuning the model as well (system prompt).

Table A.12: Episode construction and token supervision. A non-response run is a maximal stretch of `Silence` and `Hold`.

Step	Rule
Timeline	1 FPS frames with absolute `MM:SS` timestamps. Consecutive frames that share a caption are one event
States	First frame that supports the caption is `Response`; later frames of that event are `Hold`; every other frame is `Silence`
Chunks	At most 32 frames. A cut that would split a response from its hold is moved so they stay together. Original indices and timestamps are kept
Context	System instruction, standing query, and responses that precede the visual window
Extra chunks	Response-bearing chunks are kept, together with trailing and post-response chunks so sustained silence is seen
Response tokens	Every `Response` position, including its answer text
Non-response run	Its first position, plus up to four further `Silence` positions and two further `Hold` positions
No-response chunk	Two `Silence` positions

Table A.13: Shared optimization configuration. Adapters sit in the language tower; the visual encoder stays frozen.

Setting	Value
Optimizer	AdamW
Learning rate	2×10^{-4}
Schedule	Cosine decay; 5% warmup
Training duration	One epoch
Per-device batch size	1
Gradient accumulation	1
Effective global batch size	8
Precision	bfloat16 with TF32 enabled
Attention	SDPA
Memory optimization	Non-reentrant gradient checkpointing; ZeRO-2
Hardware	8× NVIDIA H200, one data-parallel process per GPU
Random and data seeds	42
Frames	384×384, bicubic, then the native image processor
LoRA	Rank 64, scale 128, dropout 0.05
Gemma modules	Attention, dense MLP, and router
Qwen modules	Attention, shared-expert MLP, and linear-recurrence projections

F TRAINING PIPELINE DETAILS

Finetuning uses the instruction sketched in Table A.11. Table A.12 describes how we build the target episodes for supervision. The loss is next-token prediction on the sampled state tokens and full answer text for response text tokens, all at equal weight. System, query, memory, visual, and unsampled target tokens stay in context only.

Examples of similar length are grouped into batches. Intermediate exposure checkpoints are saved, and the reported checkpoint is the one selected by validation PAUC-F (Section 6).